CHAPTER 5

Early error detection on word level

In the experiment reported in Chapter 4, the high WER caused many non-understandings, but only a few misunderstandings. This means that humans have an impressive capability of early error detection, meaning that they are to a large extent aware of which hypotheses are reasonable, and which are not. An important question is what this awareness is based on. In other words, if we were to build a dialogue system with such capabilities, which knowledge sources would contribute to the detection of errors?

In this chapter, two studies are presented, based on the data collected for Chapter 4. In Study I, machine learning is used with different sets of features. A main issue for machine learning is which factors (knowledge sources) the learning can and should be based on, and how to operationalise these factors into extractable features. Some factors, such as dialogue history, may seem useful for error detection, but are hard to operationalise, especially for longer contexts. Finding whether a factor contributes to the performance of a human subject doing the error detection task may provide some guidance as to its value to the machine learning task. In Study II, humans were given the task of detecting errors with different combinations of knowledge sources.

As described earlier in 3.3.1, previous studies on early error detection have to a large extent 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, many 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 task in the studies presented in this chapter can be described as *binary word-level early error detection*, in other words, to classify each word in the speech recognition result as correct or incorrect. While it would perhaps be more useful to classify concepts in the semantic interpretation of the speech recognition result, the results from this study are not dependent on the semantic model or interpretation technique used.

5.1 Study I: Machine learning

In this study, machine learning was used for the error detection task. Two learners were trained on several different sets of features in order to measure the contribution of different factors to machine learning of early error detection.

5.1.1 Algorithms used

Two machine learning algorithms were tested and compared: *transformation-based learning* and *memory-based learning*. These algorithms were chosen because they represent different machine learning paradigms and they were familiar to the author.

5.1.1.1 Transformation-based learning

In transformation-based learning, the algorithm learns a set of transformation rules that are applied after each other. It was invented by Eric Brill for use in part of speech tagging (Brill, 1995), but has been used for many other tasks as well, such as dialogue act tagging (Lager, 1999). All instances are initially tagged with the most common class. A set of rule templates has to be written specifically for the task. During training, the algorithm finds the instantiation of a template that creates the rule that most efficiently transforms the classes in the material in a positive direction. Rules learned early in the process may include very drastic general transformations that also have negative effects. However, these negative transformations may be recovered later by more specific rules. In the current study, μ-TBL (Lager, 1999) was used for transformation-based learning. μ-TBL supports the definition of clauses written in Prolog, which makes the use of features more flexible, for example when handling numeric features. However, unlike other rule learning algorithms, such as RIPPER (Cohen, 1995), μ-TBL cannot automatically find thresholds for numeric features.

5.1.1.2 Memory-based learning

In memory-based learning (also called *instance-based learning*), the training set is just stored as examples for later evaluation (Mitchell, 1997). The computation is postponed to classification (so-called “lazy” learning), when the instance to be classified is compared to all examples to find the (set of) nearest neighbour(s). The number of nearest neighbours that are compared can be tuned for the task (the algorithm is sometimes called *k-nearest neighbour*, where *k* is the number of nearest neighbours used). To measure the distance between two instances, the vectors of features for the instances are compared. In this study, TiMBL (Daelemans et al., 2003) was used for memory-based learning. TiMBL supports different ways of comparing features.

The most simple is just an overlap measure, where each feature gets one score if the values of the instances are equal. The features are typically weighted using “gain ratio weighting”, a measure that is computed using information theory. This is done at training time by analysing all examples to compute how much each feature contributes to the task. TiMBL also supports other ways of comparing the value of two features. Using “modified value difference”, the examples can be analysed to form a matrix of distances between the values of the feature. If the feature is numeric, it is also possible to use the numerical difference between features as a direct distance metric.

Memory-based learning has the advantage that learning is extremely fast (just storing examples) and that very little preparation has to be done (for example, no templates have to be written). It may also find so-called “islands of exceptions” more easily, without having to discover very specific exception rules. The disadvantage is that classifying new instances may be slow. It is therefore crucial that the algorithm has efficient methods for indexing the examples. Another disadvantage, compared to transformation-based learning, is that it is hard to study what is actually learnt. It is not possible to study any rules that might give insights into systematic properties of the data.

5.1.2 Data and features

The classification task in this experiment was to determine whether a given recognised word was present at the corresponding location in the transcription of the spoken utterance (TRUE) or not (FALSE). For this study, the recognition results from the corpus presented in Chapter 4 were aligned to the transcriptions (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 majority-class baseline to compare the machine learning performance with. Of the 4470 words, 4/5 were used as training data and 1/5 as test data.

In Table 5.1, 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 verbs.

5.1.3 Results

In order to investigate how the performance varied depending on which features that were used, different combinations of feature set groups were used. The results are shown in Table 5.2. 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 µ-TBL cannot automatically find thresholds for numeric values, a set of ten (equally sized) intervals were defined for the confidence score.

Table 5.1: Features used for error detection.

Group	Feature	Explanation
Confidence	CONFIDENCE	ASR 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 (according to Table 4.2)
	MENTIONED	Is it a content word that has been mentioned previously by the operator in the discourse?

Table 5.2: Performance of the machine learning algorithms depending on feature set.

Feature set	μ-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)

As the table shows, each group seems to add (more or less) to the performance. μ-TBL seems to perform a bit better (although the difference has not been tested for significance). With the richest feature set, μ-TBL performs 11.9% better than baseline.

The performance of the two machine learners seems 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 recognised. This gives a baseline of 69.8%. For these words, the best scores for the classifiers were 87.7% (μ-TBL) and 87.0% (TiMBL). Thus, the best classifier μ-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 μ-TBL are shown in Table 5.3. The first rule states that all content words with confidence less than 0.5 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-syllable content nouns with a confidence score high enough that have been mentioned previously by the operator should be tagged as correct.

Table 5.3: The top rules learned by μ-TBL.

Transformation	**Rule**
TRUE → FALSE	Confidence < 0.5 & Content = TRUE
TRUE → FALSE	Confidence < 0.6 & POS = Verb & Length = 2
TRUE → FALSE	Confidence < 0.4 & POS = Adverb & Length = 1
TRUE → FALSE	Confidence < 0.5 & POS = Adverb & Length = 2
TRUE → FALSE	Confidence < 0.4 & POS = Verb & Length = 1
FALSE → TRUE	Confidence > 0.4 & Mentioned = TRUE & POS = Noun & Length = 2

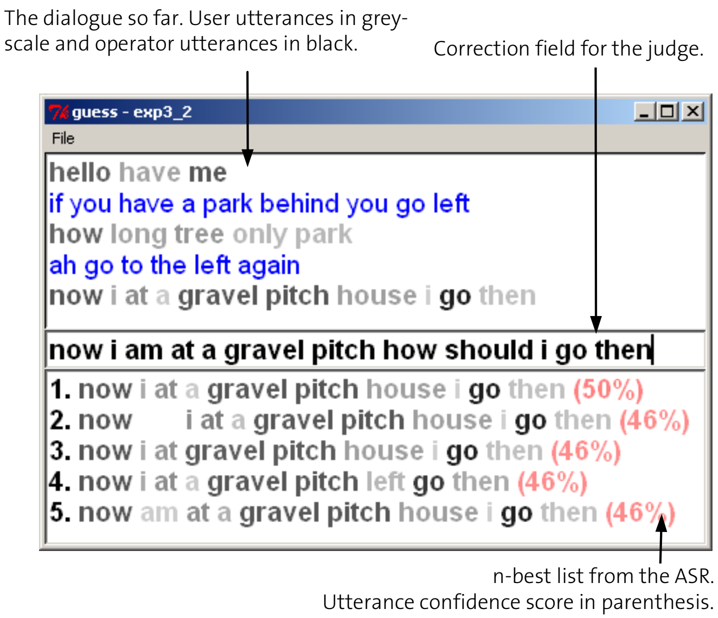
5.2 Study II: Human error detection

The features used in the machine learning study were chosen because they could intuitively contribute to error detection and they were easy to operationalise. However, it should be interesting to examine which factors humans could benefit from in performing the task, especially factors that are hard to operationalise. Finding whether a factor contributes to the performance of a human subject doing the error detection task may provide some guidance as to its value to the machine learning task. In the second study, an experiment was conducted where human subjects (henceforth referred to as judges) were asked to detect 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.

5.2.1 Method

The corpus presented in Chapter 4 was also used for this study. 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 they believed to be wrong, using a custom-made tool. Figure 5.1 shows the tool in English translation.



The dialogue so far. User utterances in grey-scale and operator utterances in black.

Correction field for the judge.

hello have me
if you have a park behind you go left
how long tree only park
ah go to the left again
now i at a gravel pitch house i go then

now i am at a gravel pitch how should i go then

1. now i at a gravel pitch house i go then (50%)
2. now i at a gravel pitch house i go then (46%)
3. now i at gravel pitch house i go then (46%)
4. now i at a gravel pitch left go then (46%)
5. now am at a gravel pitch house i go then (46%)

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

Figure 5.1: The judges' interface with an example translated into English.

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 5.4.

Table 5.4: Context levels.

Label	Description
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.

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 5.5. The order of the dialogues and context levels were systematically varied for each judge.

Table 5.5: ASR information levels.

Label	Description
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 provided.

5.2.2 Data analysis

The data consists of three versions of each recognised utterance: the transcription, the ASR result, and the judge's correction, which were all aligned to measure the judges' performance. An example is shown in Table 5.6. For each word in the recognition result that was misrecognised, the judge received one error detection point if the word was removed or changed. Since this was an error detection task and not an error correction task, the point was received regardless of whether the judge changed the erroneous word to the correct word or not (see the first word in the example). For each word that was correctly recognised, the judge received one point if the word was not removed or changed. The total number of points in each recognition result was then divided by the total number of words in the result to yield an error detection score between 0.0 and 1.0. The example in Table 5.6 yields an error detection score of 0.6. A score of 1.0 indicates that all incorrectly recognised words (insertions and substitutions) were detected and no correctly recognised words were judged as errors. A score of 0.0 indicates the opposite: all correctly recognised words were judged as errors and all errors were judged as correct.

Table 5.6: Made-up example calculation of error detection score (sub=substitution, ins=insertion).

Transcription	*the*			*correct*	*words*
ASR result	*our*	*system*	*thought*	*correct*	*words*
ASR error	sub	ins	ins	-	-
Judge's correction	*users*		*thought*	*correct*	*text*
Detection point	1	1	0	1	0

5.2.3 Results

The left column of Figure 5.2 shows mean error detection scores for the different ASR information and context levels. PreviousContext, FullContext and MapContext turned out to hold no significant differences and are thus combined into Context.

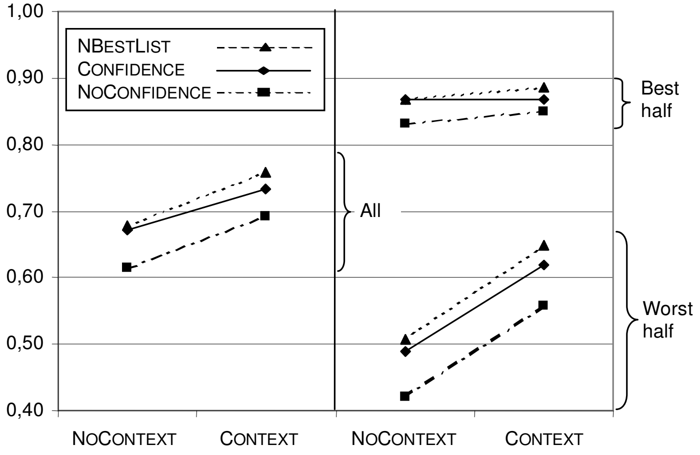


Figure 5.2: Mean error detection scores for the human judges, depending on the availability of the features. The result for all utterances is shown to the left, and the result for the best and worst half are shown to the right.

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 detection score of 0.5.

To investigate what effect average WER had on the judges' results, the figures were recalculated over two subsets of the corpus: one subset containing the 30 utterances with the highest WER, and another subset containing the 30 utterances with the lowest WER. Detection scores for the subsets are shown in the right column of Figure 5.2. 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.

5.3 Discussion

Both studies show that word confidence scores are useful for early error detection, and that other features can be used to improve performance. Utterance context and lexical information improve the machine learning performance. The errors that are found with these features probably reflect constant 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 μ-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 for the ASR language models.

It may be surprising that access to the n-best list improved the judges' performance. When simply detecting errors (and not correcting them), the information contained in the n-best list should be reflected in the word confidence scores; if a word changes in the n-best list, it is a sign that it may be incorrect, but such words usually also get a low confidence score. However, for a human subject, the fact that a word changes in the list may be easier to make use of than the grey scale of the words. Thus, the additional performance that n-best lists give could possibly be achieved by a machine learner by just looking at the confidence scores. If the n-best list would in fact be useful for a machine learner, the question is how it should be operationalised, so that it could be used in the feature set.

The discourse context of the utterance is potentially the most interesting feature, since it is not considered by the ASR. The machine learners 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 operator/system 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 on operationalising context for machine learning 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 the studies presented in this chapter, the task was a binary decision between correct and incorrect. As discussed in 3.3.1, it could sometimes be more useful to derive a continuous probabilistic confidence score as a result of the early error detection. This may be possible to derive from a memory-based learner, either by looking at the entropy of the class distribution or the density of the nearest neighbour set (i.e., the distribution of distances in the different k's; if there are a lot of close competing nearest neighbours, confidence should be low).

Since the classifiers disagree in so many cases, it would also be interesting to test whether it would be possible to use an ensemble method that could pick the right classifier.

5.3.1 Comparison to other findings

As the overview of the research on early error detection in 3.3.1 showed, related studies have also shown that ASR confidence scores are useful for early error detection, but that other features can be used to improve the performance. This study shows that this is equally true for word-level error detection. The other studies in the review did not use features from a larger context and this study confirms that larger context may not contribute much.

As was also shown in the review, other studies of early error detection have benefited from the use of prosody. It would be interesting to see if prosody could also help to detect errors on the word level, either by looking at utterance-level prosodic features or at local features. Local prosodic features may for example help to find world-level errors that arise due to disfluencies.

5.4 Summary

In this chapter, two studies were presented in which the early detection of speech recognition errors on word level was explored. 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, factors humans benefit from when detecting errors were investigated. 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, especially for content words. In the case of poor recognitions, human judges seem to benefit from using the dialogue context. However, larger context than the previous utterance does not seem to improve performance for human judges.