A Noisy Channel Approach to Error Correction in Spoken Referring Expressions

Su Nam Kim, Ingrid Zukerman, Thomas Kleinbauer and **Farshid Zavareh**
Faculty of Information Technology, Monash University
Clayton, Victoria 3800, Australia

Abstract

We offer a noisy channel approach for recognizing and correcting erroneous words in referring expressions. Our mechanism handles three types of errors: it removes noisy input, inserts missing prepositions, and replaces mis-heard words (at present, they are replaced by generic words). Our mechanism was evaluated on a corpus of 295 spoken referring expressions, improving interpretation performance.

1 Introduction

One of the main stumbling blocks for Spoken Dialogue Systems (SDSs) is the lack of reliability of Automatic Speech Recognizers (ASRs) (Pellegrini and Trancoso, 2010). Recent research prototypes of ASRs yield Word Error Rates (WERs) between 15.6% (Pellegrini and Trancoso, 2010) and 18.7% (Sainath et al., 2011) for broadcast news. However, the commercial ASR employed in this research had a WER of 30% and a Sentence Error Rate (SER) (proportion of sentences for which no correct textual transcription was produced) of 65.3% for descriptions of household objects.

In addition to mis-recognized entities or actions, ASR errors often yield ungrammatical sentences that cannot be processed by subsequent interpretation modules of an SDS, e.g., "the blue plate" being mis-heard as "*to build played*", and hesitations (e.g., "ah"s) being mis-heard as "and" or "on" — all of which happened in our trials.

In this paper, we offer a general framework for error detection and correction in spoken utterances that is based on the noisy channel model, and present a first-stage implementation of this framework that performs simple corrections of referring expressions. Our model is implemented as a preprocessing step for the *Scusi?* spoken language interpretation system (Zukerman et al., 2008; Zukerman et al., 2009).

Table 1: Spoken, heard and labeled descriptions.

Spoken:	the stool	*to*	the left of	the table
Heard:	the *storm*		the left of	the table
Labels:	**O**bject	**P**rep	**S**pecifier	**L**andmark
Spoken:	the plate		in	the microwave
Heard:	*to play*	*it*	in	the microwave
Labels:	**O**bject	**N**oise	**P**rep	**L**andmark

The idea of the noisy channel model is that a message is sent through a channel that introduces errors, and the receiver endeavours to reconstruct the original message by taking into account the characteristics of the noisy channel and of the transmitted information (Ringger and Allen, 1996; Brill and Moore, 2000; Zwarts et al., 2010). The system described in this paper handles three types of errors: noise (which is removed), missing prepositions (which are inserted), and mis-heard words (which are replaced). Table 1 shows two descriptions that illustrate these errors. The first row for each description displays what was spoken, the second row displays what was heard by the ASR, and the third row shows the semantic labels assigned to each segment in the description by a shallow semantic parser (Section 3.2). Specifically, in the first example, the preposition "*to*" is missing, and the object "stool" is mis-heard as "*storm*"; and in the second example "the plate" is mis-heard as "*to play*", and the noisy "*it*" has been inserted by the ASR.

Ideally, we would like to replace mis-heard words with phonetically similar words, e.g., use "plate" instead of "*play*". However, at present, as a first step, we replace mis-heard words with generic options, e.g., "thing" for an object or landmark. Further, we insert the generic preposition "at" for a missing preposition. Thus, we deviate from the noisy channel approach in that we do not quite reconstruct the original message. Instead, we construct a grammatically correct version of this message that enables the generation of reasonable interpretations (rather than no interpretation or non-

International Joint Conference on Natural Language Processing, pages 234–242,
Nagoya, Japan, 14-18 October 2013.

sensical ones). For example, the mis-heard description "*to play it* in the microwave" in Table 1 is modified to "the thing in the microwave". Clearly, this is not what the speaker said, but hopefully, this modified text, which describes an object, rather than an action, enables the identification of the intended object, e.g., a plate, or at least a small set of candidates, in light of the rest of the description.

Our mechanism was evaluated on a corpus of 295 spoken referring expressions, significantly improving the interpretation performance of the original *Scusi?* system (Section 6.3).

The rest of this paper is organized as follows. In the next section, we discuss related work. In Section 3, we outline the design of our system. Our probabilistic model is described in Section 4, followed by the noisy channel error correction procedure. In Section 6, we discuss our evaluation, and then present concluding remarks.

2 Related Research

This research combines three main elements: correction of ASR output, noisy channel models and shallow semantic parsing.

López-Cózar and Griol (2010) used lexical approaches to replace, insert or delete words in a textual ASR output, and syntactic approaches to modify tenses of verbs and grammatical numbers to better match grammatical expectations. However, these actions make *ad hoc* changes.

The noisy channel model has been employed for various NLP tasks, such as ASR output correction (Ringger and Allen, 1996), spelling correction (Brill and Moore, 2000), and disfluency correction (Johnson and Charniak, 2004; Zwarts et al., 2010). Our approach differs from the traditional noisy channel approach in that it uses a word-error classifier to model the noisy channel, and semantic information to model the input characteristics.

Shallow semantic parsers for SDSs have been used in (Coppola et al., 2009; Geertzen, 2009). Coppola *et al.* (2009) used FrameNet (Baker et al., 1998) to detect and filter the frames for target words, and employed a Support Vector Machine (SVM) classifier to perform semantic labeling. Geertzen (2009) used a shallow parser to detect semantic units only when a dependency parser failed to produce a parse tree. In contrast, our shallow semantic parser is part of a noisy channel model that post-processes the output of an ASR.

3 System Design

Our error correction procedure (Section 5) receives as input alternatives produced by an ASR, and generates modified versions of these alternatives. It employs the following modules: (1) a classifier that determines whether a word in a text produced by the ASR is correct; (2) a shallow semantic parser (SSP) that assigns semantic labels to segments in the text; and (3) a noisy channel error correction mechanism that decides which alterations should be made to the ASR output on the basis of the information provided by the other two modules. The resultant texts are given as input to the *Scusi?* spoken language interpretation system.

In this section, we describe the word error classifier and SSP together with our semantic labels, and report on their performance. We also provide a brief outline of the *Scusi?* system.

The performance of the classifier and SSP was evaluated in terms of accuracy over the corpus constructed to evaluate the *Scusi?* system (Kleinbauer et al., 2013). This corpus comprises 400 spoken descriptions generated by 26 speakers. We performed 13-fold cross-validation, where each fold contains two speakers (Section 6.1).

3.1 Word error classifier

We investigated three classifiers to determine whether a word in the ASR textual output is correct: the Weka implementation of Decision Trees (Quinlan, 1993) and Naïve Bayes classifiers (Domingos and Pazzani, 1997) (`cs.waikato.ac.nz/ml/weka/`), and the Mallet implementation of the linear chain Conditional Random Fields (CRF) algorithm (Lafferty et al., 2001) (`mallet.cs.umass.edu`).

The best performance was obtained by the Decision Tree, which yielded an average accuracy of 80.9% over the 13 folds. The most influential features were $rr(w, d)$ and Part-of-Speech (PoS) tag of the current word w in levels 1 and 2 of the Decision Tree respectively, where rr is the *repetition ratio* of the current word w in the textual ASR outputs for description d:

$$rr(w,d) = \frac{\text{\# of ASR outputs for } d \text{ that contain } w}{\text{\# of alternative ASR outputs for } d}.$$

3.2 Shallow Semantic Parser (SSP)

We found the following semantic labels useful for referring expressions:

- **O**bject – a lexical item designating an object, optionally preceded by a determiner and one

or more gerunds, adjectives or nouns, e.g., "*the blue ceramic drinking mug*".

- **P**reposition – a preposition or prepositional expression, e.g., "*on*" or "*further away from*".
- **L**andmark – same pattern as Object, but a description may contain more than one landmark, e.g., "the mug on *the table* in *the corner*".
- **N**oise – sighs or hesitations that are often misheard by the ASR as "and", "on" or "in".
- **S**pecifier – a further specification that normally precedes a Landmark, e.g., "the *center* of", "*front* of" or "the *left* of". The preposition "of" at the end of a Specifier that precedes a Landmark is always required.
- **A**dditional – words that are often superfluous, e.g., "the mug *that is* on the table".

We employed the Mallet implementation of the linear chain Conditional Random Fields (CRF) algorithm (Lafferty et al., 2001) to learn sequences of semantic labels (`mallet.cs.umass.edu`).

Accuracy over texts and segments was respectively measured as follows:

$$\frac{\text{\# of texts with perfectly matched label sequences}}{\text{total \# of texts}}$$

$$\frac{\text{\# of segments with perfectly matched labels}}{\text{total \# of segments}}.$$

The CRF was trained separately for textual transcriptions of spoken descriptions and for ASR outputs. Two annotators labeled the 400 transcribed texts, and 800 samples from the ASR output: 400 from the best output and 400 from the worst. The first annotator segmented and labeled the descriptions, and the second annotator verified the annotations; disagreements were resolved by consensus.

We considered the features found useful in the CoNLL2001 shared task (`http://www.cnts.ua.ac.be/conll2000/chunking/`). The features that yielded the best performance were *current word*, *current PoS* and *previous word*, achieving an accuracy of 92% over the 400 textual transcriptions, and 76.13% over the 800 ASR outputs. Accuracy over segments was higher, at 96.26% for texts, and 87.28% for ASR outputs. However, SSP's performance for the identification of Noise was rather poor, with an average accuracy of 54.75%.

3.3 *Scusi?*

Scusi? is a system that implements an anytime, probabilistic mechanism for the interpretation of spoken utterances, focusing on a household context. It has four processing stages, where each stage produces multiple outputs for a given input, early processing stages may be probabilistically revisited, and only the most promising options at each stage are explored further.

The system takes as input a speech signal, and uses an ASR (Microsoft Speech SDK 6.1) to produce candidate texts. Each text is assigned a probability given the speech wave. The second stage applies Charniak's probabilistic parser (`http://bllip.cs.brown.edu/resources.shtml#software`) to syntactically analyze the texts in order of their probability, yielding at most 50 different parse trees per text. The third stage applies mapping rules to the parse trees to generate concept graphs (Sowa, 1984) that represent the semantics of the utterance. The final stage instantiates the concept graphs within the current context. For example, given a parse tree for "the blue mug on the table", the third stage returns the uninstantiated concept graph *mug*(COLOR: blue) – *on* – *table*. The final stage then returns candidate instantiated concept graphs, e.g., `mug1`–*`location_on`*–`table2`, `mug2`–*`location_on`*–`table1`. The probability of each instantiated concept graph depends on (1) how well the objects and relations in this graph match the corresponding objects and relations in the uninstantiated concept graph (e.g., whether `mug1` is a mug, and whether it is blue); and (2) how well the relations in this graph match the relations in the context (e.g., whether `mug1` is indeed on `table2`).

4 Probability Estimation

We use a distance measure inspired by the *Minimum Message Length* (*MML*) principle (Wallace, 2005) to estimate the goodness of a message and its semantic model. This principle is normally used for model selection, based on the following formulation:

$$\Pr(data\&model) = \Pr(data|model) \times \Pr(model),$$

which strikes a balance between model complexity and data fit, i.e., the highest-probability model that best explains the data is the best model overall. That is, the best model is not necessarily the model that fits the data best, as such a model may over-fit the data; the model itself must also have a high prior probability. In our case, the data is a text, either heard by the ASR or modified, and the model is a sequence of semantic labels. At present,

our model is restricted to semantic labels for segments in referring expressions, but in the future we will use this formalism to compare models representing different dialogue acts, e.g., commands.

Our use of the MML principle differs from its normal usage in that we employ it to compare a text and its semantic model with a modified version of this text and its own semantic model (rather than comparing two models that try to account for the same text). Modifications attract a penalty that depends on the probability that they are required (the higher the probability, the lower the penalty). This penalty is applied to prevent arbitrary modifications where a system hears what it expects.

Below we describe the estimation of the probability of a text and its semantic model. The next section describes the combination of the noisy channel model with the word-error classifier, SSP, and the modifications made to texts.

The joint probability of a Text and its Semantic Model is estimated as follows:

$$\Pr(\textit{Text}\&\textit{SemModel}) = \Pr(\textit{Text}|\textit{SemModel}) \times \Pr(\textit{SemModel}) ,$$

where

- $\Pr(\textit{SemModel}) = \Pr(\textit{SSP}) \times \prod_{i=0}^{N+2} \Pr(L_i|L_0, \ldots, L_{i-1}) ,$

where $\Pr(\textit{SSP})$ reflects SSP's confidence in the sequence of semantic labels it produced for *Text*, N is the number of segments in the sequence, L_i is the label for segment i, L_{-1} and L_0 are the special labels **B**eginning, and L_{N+1} and L_{N+2} are the special labels **E**nd. To make this calculation tractable, we employ trigrams, i.e., $\Pr(L_i|L_0, \ldots, L_{i-1}) \cong \Pr(L_i|L_{i-2}, L_{i-1})$.

- $\Pr(\textit{Text}|\textit{SemModel}) = \prod_{i=1}^{N} \Pr(\textit{text}_i|L_i) ,$

where $\textit{text}_i$ is the sequence of words in segment i, and $\Pr(\textit{text}_i|L_i)$ is estimated as follows:

$$\Pr(\textit{text}_i|L_i) = \prod_{j=1}^{M_i} \Pr(\textit{HWord}_{ji}|L_i) ,$$

where M_i is the number of words in $\textit{text}_i$, and $\textit{HWord}_{ji}$ is the jth heard word in $\textit{text}_i$.

Owing to the relatively small size of our corpus, $\Pr(\textit{HWord}_{ji}|L_i)$ is roughly estimated as follows:

$$\Pr(\textit{HWord}_{ji}|L_i) = \sum_{k=1}^{T_{ji}} \Pr(\textit{HWord}_{ji}|\textit{XpctPoS}_{kji}) \Pr(\textit{XpctPoS}_{kji}|L_i),$$

where $\textit{XpctPoS}_{kji}$ is a PoS expected at position j in $\textit{segment}_i$, and T_{ji} is the number of PoS expected at position j in $\textit{segment}_i$. $\Pr(\textit{HWord}_{ji}|\textit{XpctPoS}_{kji})$ is obtained from a corpus, and $\Pr(\textit{XpctPoS}_{kji}|L_i)$ is estimated from our textual transcriptions of spoken descriptions, except for the PoS associated with Noise, which are estimated from our spoken corpus (there is no Noise in texts). We obtain a rough estimate of $\Pr(\textit{XpctPoS}_{kji}|L_i)$ by considering three positions in a segment: first, middle (intermediate positions) and last. For instance, the possible PoS for the first position of an Object or Landmark are determiner, adjective, gerund, verb(past) or noun.

To illustrate this calculation, consider the second description in Table 1, which is heard as "to play it in the microwave". The probability of the Semantic Model for this description is

$$\Pr(\textit{SemModel}) = \Pr(O|B,B) \Pr(N|O,B) \Pr(P|N,O) \Pr(L|P,N) \Pr(E|L,P) \Pr(E|E,L) .$$

All the probabilities involving Noise are set to an arbitrarily low ϵ, which yields

$$\Pr(O|B,B) \Pr(E|L,P) \Pr(E|E,L) \epsilon^3 .$$

The probability of the Text given the Semantic Model is

$$\Pr(\textit{Text}|\textit{SemModel}) = \Pr(\text{"to play"}|O) \Pr(\text{"it"}|N) \Pr(\text{"in"}|P) \Pr(\text{"the microwave"}|L) ,$$

which is quite high for "it"|N, "in"|P and "the microwave"|L, but is reduced due to the mismatch between the PoS of "to play" (TO VB) and the PoS expected by an Object, which are: DT/JJ/VBG/VBD/NN for the first position, and NN for the last position (Section 5.1.3).

Our system modifies this heard description by replacing "to play" with "the thing" and removing the noisy "it", which yields "the thing in the microwave" (Section 5). The probability of the Semantic Model for this modified sentence is

$$\Pr(\textit{SemModel}') = \Pr(O|B,B) \Pr(P|O,B) \Pr(L|P,O) \Pr(E|L,P) \Pr(E|E,L) ,$$

which is higher than that of the original Semantic Model, as is the probability of the new Text given the new Semantic Model:

$$\Pr(\textit{Text}'|\textit{SemModel}') = \Pr(\text{"the thing"}|O) \Pr(\text{"in"}|P) \Pr(\text{"the microwave"}|L) .$$

However, this gain is offset by the penalties incurred by the modifications. The estimation of these penalties is described in the next section.

5 Noisy Channel Error Correction

Given a textual output produced by an ASR, we apply Algorithm 1 to remove noise, insert prepo-

sitions and replace wrong words. The probability of the resultant text and its semantic model is recalculated after each change as described in Section 4, and is moderated by the probability of the penalty for the change. Since a modification may yield a text where SSP identifies Noise, the Noise removal step is repeated after every change.

After each modification, the probability of the original text and semantic model is compared with the probability of the new text, its semantic model and any incurred penalties. The winning text and semantic model (without penalties) are then taken as the originals for the next modification. Upon completion of this process, all the incurred penalties are re-incorporated into the final probability of a modified text, in order to enable a fair comparison with other texts that were not altered.

The application of this process to all the texts produced by an ASR for a particular utterance may yield identical texts (e.g., when words with unexpected PoS are converted to "thing"). These texts are merged, and their probabilities are recalculated. The resultant texts are ranked in descending order of probability and ascending order of the number of replaced words (i.e., texts with fewer replacements are ranked ahead of texts with more replacements, irrespective of their probability). The final probabilities are adjusted to reflect the ranking of a text.

5.1 Estimating penalties from modifications

The modifications performed by our system attract a penalty that depends on the probability that the relevant portion of a heard utterance is wrong. The higher this probability, the lower the penalty, which is implemented as a multiplier of Pr(*Text*&*SemModel*).

5.1.1 Removing noise

The penalty for removing a heard word j in $segment_i$ that is labeled as Noise by SSP is estimated on the basis of its probability of being Wrong (obtained from the word-error classifier, Section 3.1), as follows:

$$\Pr(\text{remove } HWord_{ji}) = \quad (1)$$
$$\begin{cases} \Pr(\text{IsW}(HWord_{ji}))\Pr(\text{Class}) & \text{if label = W} \\ (1-\Pr(\text{IsC}(HWord_{ji})))\Pr(\text{Class}) & \text{if label = C} \end{cases}$$

where Pr(Class) is the accuracy of the classifier (on training data), Pr(IsW($HWord_{ji}$)) is the probability assigned by the classifier to heard word j in $segment_i$ being Wrong, and Pr(IsC($HWord_{ji}$)) is the probability of this word being Correct (the last two probabilities add up to 1).

Algorithm 1 Noisy channel ASR error correction

Require: *Text*
SemModel← Run SSP on *Text*
Calculate Pr(*Text*&*SemModel*) **(Section 4)**
{ REMOVE NOISE }
while there is Noise **do**
  Text' ← Remove Noise from *Text*
  SemModel' ← Run SSP on *Text*'
  Calculate Pr(*Text*′&*SemModel*′)
  Text&*SemModel*← $\arg\max\{\Pr(\textit{Text}\&\textit{SemModel}),$
    $\Pr(\textit{Text}'\&\textit{SemModel}')\Pr(\textit{Removal})\}$
end while
{ INSERT PREPOSITIONS }
while a preposition is missing **do**
  Text' ← Insert missing preposition into *Text*
  SemModel' ← Run SSP on *Text*'
  Text' ← Remove Noise from *Text*' **(Steps 3-9)**
  Calculate Pr(*Text*′&*SemModel*′)
  Text&*SemModel*← $\arg\max\{\Pr(\textit{Text}\&\textit{SemModel}),$
    $\Pr(\textit{Text}'\&\textit{SemModel}')\Pr(\textit{Insertion})\}$
end while
{ REPLACE WRONG WORDS }
for i=1 **to** N **do**
  Text' ← Replace wrong words in $segment_i$
  SemModel' ← Run SSP on *Text*'
  Text' ← Remove Noise from *Text*' **(Steps 3-9)**
  Calculate Pr(*Text*′&*SemModel*′)
  Text&*SemModel*← $\arg\max\{\Pr(\textit{Text}\&\textit{SemModel}),$
    $\Pr(\textit{Text}'\&\textit{SemModel}')\Pr(\textit{Replacement})\}$
end for
Pr(*Text*&*SemModel*) ← Pr(*Text*&*SemModel*)
  $\Pr(\textit{Removal})\Pr(\textit{Insertion})\Pr(\textit{Replacement})$

The rationale for this formula is that if SSP deems a heard word to be Noise, and the classifier labels it Wrong with high probability, then its removal should cause only a small reduction in Pr(*Text*&*SemModel*). Conversely, if a heard word deemed to be Noise by SSP is labeled Correct by the classifier with high probability, then its removal should cause a large reduction in Pr(*Text*&*SemModel*). In both cases, the probabilities assigned to the labels by the classifier are moderated by the classifier's accuracy.

To illustrate this process, let's return to the example "to play it in the microwave", where "it" is labeled Noise by SSP, and Wrong by the classifier with probability Pr(IsW("it")). A new text *Text*' is obtained as a result of the removal of "it", and the penalty Pr(IsW("it")) Pr(Class) is multiplied by the new Pr(*Text*′&*SemModel*′).

5.1.2 Inserting a preposition

If a preposition is not found in a position where one is expected, e.g., between an Object and Landmark or between an Object and a Specifier, we insert a generic preposition "at". The penalty

for the insertion of a preposition depends on the probability that the ASR failed to hear an uttered preposition, which is estimated as follows:

$$\Pr(\text{insert } P_i) = \Pr(P_i \text{ appears in } \textit{Text} \text{ and doesn't appear in the ASR output for } \textit{Text}),$$

where P_i is a preposition in position i in *Text*.

To determine the frequency of this event, we employ an edit distance algorithm that aligns the texts produced by the ASR with their corresponding textual transcriptions. This was done for 800 alternatives produced by the ASR (400 best and 400 worst), yielding a probability of 0.02 of the ASR dropping a preposition. The corresponding penalty for inserting a preposition (0.02) is hopefully offset by the increase in $\Pr(\textit{SemModel}')$ as a result of this insertion. For instance, the probability of the Semantic Model for the heard description (without a preposition) in the first example in Table 1 is

$$\Pr(\textit{SemModel}) = \Pr(O|B,B)\Pr(S|O,B)\Pr(L|S,O)\Pr(E|L,S)\Pr(E|E,L),$$

where $\Pr(S|O,B)$ and $\Pr(L|S,O)$ are low, as they are ungrammatical. After adding the preposition,

$$\Pr(\textit{SemModel}') = \Pr(O|B,B)\Pr(P|O,B)\Pr(S|P,O)\Pr(L|S,P)\Pr(E|L,S)\Pr(E|E,L).$$

Although the new expression has an extra factor, the probabilities of the new factors are higher than those of their original counterparts.

5.1.3 Replacing a word

The decision to replace a word is based on the match between expected PoS and the PoS of a heard word. If they match, no replacement is performed. Otherwise, replacements are performed by applying the following rules, which are based on the PoS expected by the different types of segments at each position (first, middle, last).

- **Objects and Landmarks –** The expected PoS for Objects and Landmarks are: DT/JJ/VBG/VBD/NN for the first word, JJ/VBG/VBD/NN for the middle words, and NN for the last word. Thus, if there is a PoS mismatch, we perform the following replacements (if there is only one word in an Object or Landmark, we replace it with "thing" (NN)):
  - $\textit{HWord}_1 \Rightarrow$ "the" (DT)
  - $\textit{HWord}_{\text{mid}} \Rightarrow$ "unknown" (JJ) (multiple times)
  - $\textit{HWord}_{\text{last}} \Rightarrow$ "thing" (NN)

  To illustrate this process, consider the heard Object "to:TO battle:NN played:VB", which is replaced with "the:DT battle:NN thing:NN". Even though "battle" is incorrect, it is not modified, as its PoS is expected. However, *Scusi?* can cope with such unknown object attributes.

- **Prepositions and Prepositional Phrases –** This segment is more restricted than Objects and Landmarks, as it is largely composed of closed class words. We therefore use edit distance to find the prepositional phrase in the corpus of textual transcriptions that best matches the words in a heard prepositional phrase. The phrase from the corpus then replaces the heard segment. If there is no best-matching prepositional phrase, the generic "at" is used as a replacement. For example, "for the wave from" is replaced with "further away from" (with "from" being the next-best match), while "a all" is replaced with "at".

- **Specifiers –** This segment is similar to Objects and Landmarks plus a final "of" when it precedes a Landmark (about 5% of the descriptions had Specifiers without Landmarks). In addition, the head noun, which is normally the penultimate word in a Specifier, must be a positional noun, such as "center", "edge" or "corner". Thus, a word is replaced if a PoS mismatch occurs or the penultimate word is not an expected positional noun, as follows:
  - $\textit{HWord}_1 \Rightarrow$ "the" (DT)
  - $\textit{HWord}_{\text{mid}} \Rightarrow$ "unknown" (JJ) (multiple times)
  - $\textit{HWord}_{\text{last}-1} \Rightarrow$ "position" (NN)
  - $\textit{HWord}_{\text{last}} \Rightarrow$ "of" (IN, preposition)

  For instance, given the Specifier "the:DT ride:NN into:IN" followed by a Landmark, "of:IN" is appended, and "into:IN" is replaced with "position:NN", yielding "the:DT ride:NN position:NN of:IN". Clearly, other replacement options are possible, which will be investigated in the future.

In principle, the penalty for replacing a word should depend on both the probability that it is wrong (as for noise removal) and on the similarity between the wrong word and the proposed replacement. That is, the higher the probability that a word is wrong, and the higher the similarity between the original word and the replacement, the lower the penalty for the replacement. However, at present, we replace words that do not match an

expected PoS only with generic options, e.g., "unknown" for expected adjectives, "thing" for expected nouns in Objects and Landmarks, and "position" for expected positional nouns in Specifiers. Thus, our penalty consists only of the first of the above factors moderated by a generic similarity factor $\delta(= 0.5)$, as follows:

$$\Pr(\text{replace } \mathit{HWord}_{ji}) = \quad (2)$$
$$\begin{cases} \delta \Pr(\text{IsW}(\mathit{HWord}_{ji}))\Pr(\text{Class}) & \text{if label = W} \\ \delta\,(1-\Pr(\text{IsC}(\mathit{HWord}_{ji})))\Pr(\text{Class}) & \text{if label = C} \end{cases}$$

In the future, the generic δ will be replaced by a function of the similarity between an original word and its candidate replacements.

6 Evaluation

In this section, we describe our corpus and evaluation metrics, and compare the results obtained with *Scusi?* plus error correction with those obtained by the original *Scusi?* system.

6.1 Corpus

Our model's performance was evaluated using part of the corpus constructed to evaluate the *Scusi?* system (Kleinbauer et al., 2013). The original corpus comprises 432 free-form descriptions spoken by 26 trial subjects to refer to 12 designated objects in four scenarios (three objects per scenario, where a scenario contains between 8 and 16 objects; participants repeated or rephrased some descriptions). 32 descriptions could not be processed by the ASR, and 105 contained constructs that could not be represented by *Scusi?*. The remaining 295 descriptions were used in our evaluation.

The descriptions, which varied in length and complexity, had an average description length of 10 words. Sample descriptions are: "the green plate next to the screwdriver at the top of the table", "the large pink ball in the middle of the room", "the plate on the corner of the table" and "the computer under the table".

6.2 Evaluation metrics

We use the evaluation metrics discussed in (Kleinbauer et al., 2013), viz *%NotFound@*N, the percentage of descriptions that have no correct interpretation within the top N ranks; *Fractional Recall@N* (*FRecall@*N), which represents the fact that the ranked order of equiprobable interpretations is arbitrary; and *Normalized Discounted Cumulative Gain@N* (*NDCG@*N), which discounts interpretations with higher (worse) ranks (Järvelin and Kekäläinen, 2002). The last two metrics are defined as follows:

$$\mathit{FRecall@}N(d) = \frac{\sum_{j=1}^{N} \mathit{fc}(I_j)}{|C(d)|}\,,$$

where d is a description, $C(d)$ is the set of correct interpretations for d, I_j is an interpretation generated by *Scusi?* at rank j, and *fc* is the fraction of correct interpretations among those with the same probability as I_j (this is a proxy for the probability that I_j is correct):

$$\mathit{fc}(I_j) = \frac{c_j}{h_j - l_j + 1}\,,$$

where l_j is the lowest rank of all the interpretations with the same probability as I_j, h_j the highest rank, and c_j the number of correct interpretations between rank l_j and h_j inclusively.

DCG@N allows the definition of a relevance measure for a result, and divides this measure by a logarithmic penalty that reflects the rank of the result. Using $\mathit{fc}(I_j)$ as a measure of the relevance of interpretation I_j, we obtain

$$\mathit{DCG@}N(d) = \mathit{fc}(I_1) + \sum_{j=2}^{N} \frac{\mathit{fc}(I_j)}{\log_2 j}\,.$$

This score is normalized to the $[0, 1]$ range by dividing it by the score of an ideal answer where $|C(d)|$ correct interpretations are ranked in the top $|C(d)|$ places, yielding

$$\mathit{NDCG@}N(d) = \frac{\mathit{DCG@}N(d)}{1 + \sum_{j=2}^{\min\{|C(d)|,N\}} \frac{1}{\log_2 j}}$$

6.3 Results

Table 2 compares the performance of the original *Scusi?* system with that of *Scusi?* plus error correction, and displays the performance obtained for three types of modifications: N+P, P+R and N+P+R, where N stands for noise removal, P for preposition insertion, and R for word replacement (preposition insertion was folded into all the options, as it happens in only 2% of the cases). The table shows the average of *%NotFound*, *FRecall* and *NDCG* for the 295 descriptions in our corpus. The best performance is boldfaced.

As seen in Table 2, *Scusi?* plus error correction *with word replacement* generally outperforms the original *Scusi?* system (the Object/Landmark replacement has the greatest impact on performance among the three types of word replacements). *Scusi?*+N+P+R yields the best overall performance for *FRecall@*∞ and

Table 2: Performance comparison: original *Scusi?* versus *Scusi?* + Noisy Channel Error Correction.

Average of	*Scusi?*	Noisy Channel Error Correction		
		N+P	P+R	N+P+R
%NotFound@∞	22.37%	22.03%	14.24%	**13.90%**
%NotFound@20	28.14%	28.47%	**23.39%**	24.41%
%NotFound@10	31.86%	31.19%	**24.75%**	26.78%
%NotFound@3	37.97%	40.00%	**32.88%**	36.27%
%NotFound@1	44.75%	47.80%	**40.00%**	44.41%
FRecall@∞	0.776	0.778	0.858	**0.859**
FRecall@20	0.709	0.699	**0.753**	0.741
FRecall@10	0.667	0.662	**0.731**	0.712
FRecall@3	0.598	0.567	**0.636**	0.600
FRecall@1	0.488	0.462	**0.508**	0.481
NDCG@∞	0.641	0.626	**0.688**	0.666
NDCG@20	0.628	0.610	**0.669**	0.644
NDCG@10	0.617	0.601	**0.663**	0.636
NDCG@3	0.589	0.562	**0.624**	0.591
NDCG@1	0.516	0.490	**0.538**	0.511

%NotFound@∞ (statistically significantly better than *Scusi?* with $p<<0.01$ for the Wilcoxon signed rank test), while *Scusi?*+P+R yields the best performance for the remaining measures (statistically significantly better than *Scusi?* for *FRecall*@∞,20,10,3, *NDCG*@∞,20,10,3 and all values of *%NotFound*; and statistically significantly better than *Scusi?*+N+P+R for *FRecall*@3,1, all values of *NDCG* and *%NotFound*@3,1, $p\leq 0.05$). The fact that *Scusi?*+N+P+R outperforms *Scusi?*+P+R only for *%NotFound*@∞ and *FRecall*@∞ indicates that while the combination of noise removal with the other corrections enables *Scusi?* to find additional correct interpretations, these interpretations tend to appear in high (bad) ranks. The performance of *Scusi?*+N+P is generally worse than that of the original *Scusi?* system — a disappointing outcome that may be attributed to the low accuracy of SSP in the identification of Noise (54.75%, Section 3.2).

Further examination of our results reveals the following types of errors: (1) ASR errors that rendered a description unprocessable by other stages, e.g., "the green plate next to the hammer" heard as "*degree in applied* next to *him are*", and "the picture above the table" heard as "the picture *of the that*"; (2) ASR errors that were not corrected, as the PoS was expected, e.g., "the center *off/IN* the table"; (3) wrong expression replacements, e.g., "the plate:O | next to *scholar of*:P" corrected as "the plate:O | next to:P"; and (4) out of vocabulary terms, e.g., "motherboard" and "frame".

An interesting pattern emerges when considering ASR errors. Both *Scusi?*+N+P+R and *Scusi?*+P+R outperform the original version of *Scusi?* for the 193 descriptions with no correct textual interpretation (SER = 65.3%, Section 1), while the original version of *Scusi?* performs at least as well as the best option, *Scusi?*+P+R, for the 102 descriptions where a correct textual interpretation was found (Table 3). This indicates that SSP is over-zealous in finding errors, and its performance must be further investigated, or another mode of operation considered (e.g., retaining both the original and the modified ASR output).

Table 3: Performance broken down by SER.

ASR output	Average of	*Scusi?*	P+R	N+P+R
all wrong	*%NotFound*@1	61.66%	**52.85%**	54.40%
(193 desc.)	*%NotFound*@10	44.56%	**33.68%**	35.75%
one correct	*%NotFound*@1	**12.75%**	15.69%	25.50%
(102 desc.)	*%NotFound*@10	**7.84%**	**7.84%**	9.80%

7 Discussion and Future Work

We have offered a noisy channel approach for error correction in spoken utterances, with a first-stage implementation that corrects errors by removing noise, inserting prepositions, and replacing wrong words with generic terms. Our approach yields significant improvements in interpretation performance, and shows promise for achieving further improvements with more sophisticated interventions.

The structure of referring expressions is rather rigid in terms of the order of the semantic segments. To test the general applicability of our noisy channel model, we propose to consider other types of dialogue acts, and take into account the expectations from the dialogue, e.g., "to play a CD" is modified when it is considered a mis-heard description, but if it were a response to the question "what would you like me to do?", no changes would be required. In addition, we will extend our approach to propose specific, rather than generic, word replacements, and to handle superfluous information (i.e., information that is meaningless to the language interpretation module) or missing information (e.g., missing landmarks). Another avenue of research involves versions of *Scusi?* that employ SSP as an alternative to or in combination with a syntactic parser.

Acknowledgments

This research was supported in part by grants DP110100500 and DP120100103 from the Australian Research Council. The authors thank Masud Moshtaghi for his help with statistical issues.

References

C.F. Baker, C.J. Fillmore, and J.B. Lowe. 1998. The Berkeley FrameNet project. In *COLING'98 – Proceedings of the 17th International Conference on Computational Linguistics*, pages 86–90, Montreal, Canada.

E. Brill and R.C. Moore. 2000. An improved error model for noisy channel spelling correction. In *ACL'2000 – Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 286–293, Hong Kong.

B. Coppola, A. Moschitti, and G. Riccardi. 2009. Shallow semantic parsing for spoken language understanding. In *NAACL-HLT 2009 – Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 85–88, Boulder, Colorado.

P. Domingos and M. Pazzani. 1997. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130.

J. Geertzen. 2009. Semantic interpretation of dutch spoken dialogue. In *IWCS-8 – Proceedings of the 8th International Conference on Computational Semantics*, pages 286–290, Tilburg, The Netherlands.

K. Järvelin and J. Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

M. Johnson and E. Charniak. 2004. A TAG-based noisy channel model of speech repairs. In *ACL'04 – Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 33–39, Barcelona, Spain.

Th. Kleinbauer, I. Zukerman, and S.N. Kim. 2013. Evaluation of the *Scusi?* spoken language interpretation system – A case study. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, Nagoya, Japan.

J.D. Lafferty, A. McCallum, and F.C.N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML'2001 – Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, Williamstown, Massachusetts.

R. López-Cózar and D. Griol. 2010. New technique to enhance the performance of spoken dialogue systems based on dialogue states-dependent language models and grammatical rules. In *Proceedings of Interspeech 2010*, pages 2998–3001, Makuhari, Japan.

T. Pellegrini and I. Trancoso. 2010. Improving ASR error detection with non-decoder based features. In *Proceedings of Interspeech 2010*, pages 1950–1953, Makuhari, Japan.

J. R. Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, California.

E. Ringger and J.F. Allen. 1996. Error correction via a postprocessor for continuous speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 427–430, Atlanta, Georgia.

T.N. Sainath, B. Ramabhadran, M. Picheny, D. Nahamoo, and D. Kanevsky. 2011. Exemplar-based sparse representation features: From TIMIT to LVCSR. *IEEE Transactions on Audio, Speech and Language Processing*, 19(8):2598–2613.

J.F. Sowa. 1984. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, MA.

C.S. Wallace. 2005. *Statistical and Inductive Inference by Minimum Message Length*. Springer, Berlin, Germany.

I. Zukerman, E. Makalic, M. Niemann, and S. George. 2008. A probabilistic approach to the interpretation of spoken utterances. In *PRICAI 2008 – Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence*, pages 581–592, Hanoi, Vietnam.

I. Zukerman, P. Ye, K.K. Gupta, and E. Makalic. 2009. Towards the interpretation of utterance sequences in a dialogue system. In *Proceedings of the 10th SIGdial Conference on Discourse and Dialogue*, pages 46–53, London, United Kingdom.

S. Zwarts, M. Johnson, and R. Dale. 2010. Detecting speech repairs incrementally using a noisy channel approach. In *COLING'2010 – Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1371–1378, Beijing, China.