

Sentence Verification in Spoken Dialogue System

Huei-Ming Wang and Yi-Chung Lin

Advanced Technology Center, Computer and Communications Research Laboratories,
Industrial Technology Research Institute, Taiwan
{hmw, lyc}@itri.org.tw

Abstract

In spoken dialogue systems, sentence verification technique is very useful to avoid misunderstanding user's intention by rejecting out-of-domain or bad quality utterances. However, compared with word verification and concept verification, sentence verification has been seldom touched in the past. In this paper, we propose a sentence verification approach which uses discriminative features extracted from the edit operation sequence. Since the edit operation sequence indicates what kinds of errors (i.e., insertion, deletion and substitution errors) may occur in the hypothetical concept sequence, it conveys sentence-level information for evaluating the quality of system's interpretation for the user's utterance. In addition, a sentence verification criterion concerning precision and recall rates of hypothetical concepts is also proposed to pursue efficient and correct spoken dialogue interactions. Compared with the verification method using acoustic confidence measure, the proposed approach reduces 17.3% of errors.

1. Introduction

Although speech recognizer provides more and more accurate recognition, it's still far from perfect. The spontaneous utterances are usually disfluent, noisy, or out-of-domain. Errors in recognition are inevitable and, consequently, degrade the performance of spoken dialogue systems. Therefore, to detect the errors and then stop the errors propagating, the mechanism of utterance verification has become increasingly popular in real-world spoken dialogue systems recently.

Many researches have been done for verifying utterances of different grain. Most of them focused on word- and concept-level verification and have achieved applauded accuracy in detecting speech recognition errors [1][2][3]. On the other hand, however, sentence verification is still seldom touched. One may infer that the sentence verification can be done by precisely verifying every word/concept in the utterance. The reality is that in most cases of out-of-domain or bad quality utterances, even the state-of-the-art word/concept verification is adopted, unexpected words/concepts are likely to be accepted. The undesirable action and response due to the unexpected input would confuse the user and result in diverse dialogue. Besides, it is clear to decide to accept or reject the whole utterance if all words/concepts are verified to be correct or incorrect, however, it is arguable how to make a good decision if only some words/concepts in the utterance are incorrect. To reject every utterance with one or few incorrect words/concepts and ask the user to repeat again and again results in too lengthy dialogue. On the contrary, to accept the utterance with too many errors tends to make the system misunderstand user's need and, consequently, give an inappropriate response that would confuse the user. To sum

up, a sentence verification method considering both interaction efficiency and response correctness is highly demanded for spoken dialogue systems.

In prior study [4], sentence verification is using classification model to classify utterances into the accepted and the rejected. The sentence-level features used for classification are the recognition scores for the entire utterances of top-choice hypothesis, and the scores extracted from N-best hypotheses. This sentence verification is willing to reject utterances that are extremely difficult in recognition. The verification criterion is accepting utterances that the correct answer in one of top four hypotheses or at least two correctly recognized out of every three words in the top-choice, and rejecting the others. Another sentence verification [5] is using the concept verification to verify each phrase in the utterance first, then the verified hypothesis that doesn't contain complete semantic representation and most of its concepts are rejected will be rejected.

In this paper, we propose a sentence verification mechanism using novice sentence-level features from the edit operation sequence obtained from the error-tolerant language understanding approach [6]. The edit operation sequence indicates what kinds of errors (i.e., insertion, deletion and substitution errors) may occur in the hypothetical concept sequence. The selection of most probable concept sequence and its corresponding edit operation sequence is an integrated procedure of concept parsing and exemplary sentences matching. Therefore, the edit operation sequence is an important sentence-level evidence for evaluating the quality of system's interpretation.

How to decide which utterance should be accepted is a critical issue. We also propose a decision criterion for sentence verification by considering both efficiency and correctness of spoken dialogue systems. The idea is that losing information is more tolerable than misunderstanding for spoken dialogue systems. But losing too much information also indicates less reliability. Since the precision rate of the hypothetical concepts reflects the correctness of the interpretation of user utterance, it must be 100% to make sure the system does not misunderstand the user. On the other hand, the threshold of recall rate can be lower than 100% to pursue efficient interaction.

The proposed sentence verification approach is tested on the utterances collected from a Chinese spoken dialogue system providing off-line delegation service. The experimental results show that the edit operation sequence is helpful in sentence verification. Compared to the verification method using acoustic confidence measure, the accuracy is improved from 63.8% to 70%, which corresponds to 17.3% error reduction rate.

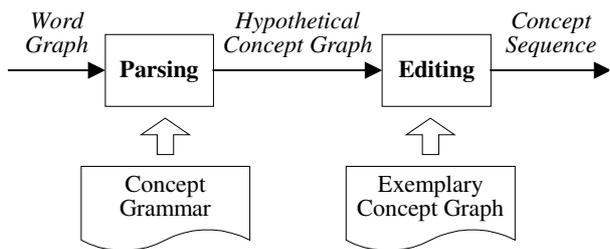


Figure 1: Block diagram of the error-tolerant language understanding model.

2. Error-tolerant language understanding

The proposed sentence verification uses the edit operation sequence generated from the error-tolerant language understanding model [6]. The central idea of this approach is using a set of exemplary concept sequences to provide the clues for detecting and recovering errors. As shown in Figure 1, the error-tolerant language understanding method takes two steps. First step is to parse the word graph into a concept graph according to a predefined concept grammar. Then, checking and editing the concepts by matching the hypothetical concept graph with the exemplary concept graph. The exemplary concept graph is generated from the exemplary concept sequences that are the outcome of parsing the exemplary sentences collected in the application domain.

In our design, there are four categories of edit operations, including *Insertion*, *Deletion*, *Substitution* and *Acceptance*. An edit operation is denoted as $\langle x, y \rangle$, where x is a concept in the hypothetical concept sequence and y is a concept in the exemplary concept sequence. The detailed descriptions of edit operations are shown in Table 1.

A simplified example describes how the error-tolerant language understanding works. Assume that the user said “tell me forecast in Taipei tonight” but the speech recognizer mis-recognize it as “Miami forecast in Taipei tonight”, where “Miami” is a mis-recognized word. The hypothetical word sequence can be parsed to the hypothetical concept sequence “*Location Topic Location Date*”. The duplicated crucial concept *Location* will confuse the system. In the error-tolerant model, the error can be detected immediately because no path in the exemplary concept graph is the same as the hypothetical concept sequence. Then, the most probable exemplary concept sequence, “*Query Topic Location Date*”, is selected as the user’s real intention, and the corresponding edit operation sequence, substituting the first concept and accepting the others, is performed to recover the error. Consequently, the corrected concept sequence makes dialogue system immune to this recognition error.

In general, the task of understanding user’s utterance can be formulated as follows.

$$(\tilde{W}, \tilde{F}, \tilde{C}, \tilde{K}, \tilde{E}) = \arg \max_{(W, F, C, K, E)} P(W, F, C, K, E | U) \quad (1)$$

where U is the user utterance, W is one possible word sequence, F is one possible concept parse forest of W , C is the concept sequence of F , K is one exemplary concept sequence, and E denotes one edit operation sequence which edits C to K . $(\tilde{W}, \tilde{F}, \tilde{C}, \tilde{K}, \tilde{E})$ is the most probable word sequence, concept parse forest, hypothetical concept sequence, exemplary

Table 1: Description of edit operation $\langle x, y \rangle$.

Condition	Category	Description
$x = \epsilon, y \neq \epsilon$	Insertion	Insert y
$x \neq \epsilon, y = \epsilon$	Deletion	Delete x
$x \neq \epsilon, y \neq \epsilon, x \neq y$	Substitution	Substitute x with y
$x \neq \epsilon, x = y$	Acceptance	Take x

concept sequence and edit operation sequence respectively. If $\tilde{C} = \tilde{K}$, the hypothetical concept sequence is determined to be legal. Otherwise the hypothesis \tilde{C} is determined to be incorrect and the edit operations in \tilde{E} will be used to fix the errors in the hypothesis.

The following scoring function derived in [6] was used to select the most probable configuration $(\tilde{W}, \tilde{F}, \tilde{C}, \tilde{K}, \tilde{E})$ as follows:

$$S(W, F, C, K, E) = w_1 \times S_A + w_2 \times S_G + w_3 \times S_K + w_4 \times S_E \quad (2)$$

where w_1, w_2, w_3, w_4 are positive weighting factors and S_A, S_G, S_K, S_E are four kinds of scores obtained from different analysis phases. The score S_A is the acoustic score provided by the speech recognizer. The score S_G is the grammar score estimated by the stochastic context-free grammar model as:

$$S_G = \sum_{T \in F, A \rightarrow \alpha \in T} \log P(\alpha | A) \quad (3)$$

where T is a concept parse in F and $A \rightarrow \alpha$ is one of the context-free rules that assemble T . The score S_K is the example score estimated by a bigram model as follows:

$$S_K = \sum_{i=1}^m \log P(k_i | k_{i-1}) \quad (4)$$

where m is the number of concept in K and k_i is the i -th concept. The score S_E is the edit score defined as follows:

$$S_E = \sum_{\substack{e = \langle x, y \rangle \in E \\ x = \epsilon}} \log P(\Psi(e)) + \sum_{\substack{e = \langle x, y \rangle \in E \\ x \neq \epsilon}} \log P(\Psi(e) | \delta_x) \quad (5)$$

where $\Psi(\cdot)$ is a mapping function to map an edit operation to its category, δ_x denotes the confidence measure of the hypothetical concept x . The first factor on the right hand side indicates that the penalty of a *Insertion* operation is assigned according to its prior probability because there is no hypothetical concept associated to *Insertion* operations. For the other categories of operation, as indicated by the second factor, the penalty is assigned according to the confidence measure of the associated hypothetical concept.

A dynamic programming procedure is used to find the most matched pair of the hypothetical concept sequence and the exemplary concept sequence according to the scoring function (2). Tested on cellular phone calls, the error-tolerant model improves the precision and recall for 32.2% and 13.7% respectively in terms of error reduction rate in comparison with the concept bigram model.

3. Sentence verification and criterion

Even error-tolerant language understanding model achieves better understanding performance, sentence verification is

indispensable for spoken dialogue systems. Considering an out-of-domain utterance "will Los Angeles Lakers win Houston Rockets tomorrow" for weather forecast system, its recognition word graph contains the best hypothesis of "will Los Angeles late wind Houston rain needs tomorrow". For the best hypothesis, the error-tolerant model matches the exemplary concept sequence "Query Location Topic Date" with the edit operation sequence of accepting the concepts of "will", "Los Angeles", "rain", "tomorrow" and deleting the other concepts. Thus, the user's intention is extracted as "will Los Angeles rain tomorrow" and the probability of rain in Los Angeles will be replied. A reply that system cannot handle the request will be better if the sentence can be verified.

3.1. Sentence verification with edit operation sequence

Take the above out-of-domain example. The user's real intention is distorted by applying many *Deletion* operations. The other example mentioned in section 2, the erroneous recognition output is corrected by applying many *Acceptance* operations with one *Substitution* operation. Edit operation sequences should convey useful information to verify sentences based on these observations.

In the error-tolerant approach, the input utterance is interpreted by finding the most probable word sequence, concept parse forest, hypothetical concept sequence, exemplary concept sequence and edit operation sequence as formulated in (1). Thus, in addition to the total score of formula (2), the edit operation sequence can be taken as an indicator of overall interpretation quality for the input utterance. Moreover, as shown in equation (5), the selection of edit operation is dependent on the confidence measure of edited concept. If the confidence measure is high, the probability of selecting *Acceptance* operation is greater than the probability of selecting the other operations and it makes the corresponding concept tend to be retained. Otherwise, the probability of *Deletion* operation is greater and it makes the corresponding concept tend to be deleted [6]. Thus, the entire interpretation is unreliable if the edit operation sequence contains many edit operations except *Acceptance*.

In our design, the interpretation of user's utterance is verified by feeding a classification model with the features extracted from edit operation sequence. The classification model is responsible for cast the given feature vector into one of the two categories: acceptance and rejection. In this paper, the Support Vector Machine (SVM) is adopted as our classifier for its greater ability to generalize comparing to other statistical classification models [7].

The input of SVM is a scalar feature vector and the output is a scalar indicator for two classes such as 1 and -1. Thus, the edit operation sequence has to be transformed to a scalar feature vector as the input of SVM. It is infeasible to directly use the whole edit operation sequence as a feature. Therefore, we decompose the whole sequence into n-gram fragments, including bigrams, trigrams, and four-grams. Then, the following three scores are computed.

$$s_{A,1} = \frac{1}{n} \times \sum_{i=1}^n P_A(e_i | e_{i-1}), \quad s_{A,2} = \frac{1}{n} \times \sum_{i=1}^n P_A(e_i | e_{i-1}, e_{i-2}),$$

$$s_{A,3} = \frac{1}{n} \times \sum_{i=1}^n P_A(e_i | e_{i-1}, e_{i-2}, e_{i-3}),$$

where n is the number of edit operations in the edit operation sequence, e_i is the i -th edit operation, $P_A(\cdot)$ indicates the probability trained with the utterances that should be accepted. Likewise, the other three scores $s_{R,1}, s_{R,2}, s_{R,3}$ are also computed by using the utterances that should be rejected.

3.2. Sentence verification criterion

In addition to a classification model that can classify the accepted and the rejected precisely, the sentence verification criterion is critical to the efficiency and correctness of dialogues. A strict criterion that rejects every utterance with one or few incorrect words/concepts results in too lengthy dialogue. On the contrary, a looser criterion that accepts the utterance with too many errors tends to make the system misunderstand user's need and, consequently, give an inappropriate response that would confuse the user.

The design of verification criterion bases on the concept that losing information is more tolerable than misunderstanding for spoken dialogue systems. The former could be recovered by the following turns but the latter could destroy the dialogues. Besides, losing too much information also indicates less reliability. Based on this criterion, two quantities are referred as indicators, the precision rate and the recall rate that represent the correctness and completeness of understanding respective. Higher precision and recall requirement means a stricter criterion with more false alarms. On the other hand, lower precision and recall requirement introduces more false accept. In this paper, the interpretation of an utterance is accepted only if the precision rate of hypothetical concepts is 100% and the recall rate is more than 60%. That means the utterance is accepted if there is no misunderstanding and at least three out of five concepts that are extracted correctly; otherwise, it is rejected.

4. Experiments and Discussions

The proposed sentence verification is tested on a corpus collected by our Chinese spoken dialogue system that provides off-line delegation services [8], such as morning call, reminder, etc. A task assignment utterance will contain a time expression or a condition expression, and the content of the task, such as "call me and tell me the weather in Taipei at seven o'clock in the weekend" or "call me when the price of Microsoft is down below forty dollars". Actually, the query sentences this system needs to understand are much more complicated than those faced by the systems provide weather information or stock information [9]. The experiment data consists of 2,211 utterances made by wired phone calls collected by the delegation dialogue system. These utterances can be divided into two subsets. The first subset is named well-formed set, which consists of 1,630 utterances that can be parsed into one of the exemplary concept sequences. Another subset is called ill-formed set, which consists of 591 utterances that must be edited according to the selected edit operation sequences. Because there is no discriminative

Table 2: The error rates and error reduction rates of different verification models.

Model	Error Rate	Error Reduction Rate
NV	49.2%	–
CM	36.2%	28.7%
ED	33.0%	35.0%
ED+CM	30.0%	41.0%

information of edit operation sequence for all *Acceptance* sequence, the experiment is conducted on the ill-formed set.

In the experiment, four different models are evaluated to show their capabilities to verify sentence. The first model, named NV (no verification), is a dummy model that accepts all utterances without verifying. The second model, named CM model, assesses the interpretation reliability of an utterance by using the sentence confidence measure, which defined as the average acoustic confidence measure of hypothetical concepts. Sub-word verification approach [1] is used to provide the acoustic confidence measure for the word sequence in a concept. The third model, named ED model, is the proposed verification model adopting only feature vector of six n-gram features of the edit operation sequence. The fourth model, named ED+CM model, is the proposed verification model adopting the sentence confidence measure of CM model as a feature in addition to the features of ED model.

The v -fold cross validation method [10] is used to reduce the error of performance assessment. The ill-formed set are randomly cast into 5 subsets (i.e., v is set to 5). Each subset is comprised of 118 or 119 utterances. A total of 5 simulations are conducted. Every simulation holds out one particular subset for testing and uses the other 4 subsets for training.

The classifier is built with a popular and efficient Support Vector Machine package, libsvm [11]. In this package, there are several useful tools to tune the performance of SVM classifier: a re-scaling tool that helps to re-scale the features in a certain range; a model selection tool that helps to select the combination of model parameters. By using these tools, the features of ED model and ED+CM model are re-scaled to the range of (1,-1) first, and then, the model with parameters, the cost of 4 and radial basis kernel function with gamma of 1, is used as our classifier.

The performances of different models are compared in terms of the error rate of classification. Table 2 lists the error rates of the four models. The error reduction rates listed in the last column are obtained with respect to the error rate of NV model. The experimental results show that the sentence verification is useful in the ill-formed utterances. The simplest CM model gets an error reduction rate of 28.7% comparing to the do-nothing NV model. Even the CM model achieves a great gap from NV model, the proposed ED model still outperforms the CM model for an extra error reduction rate of 8.9%. This result shows that the information of edit operation sequence is really helpful in verifying ill-formed utterances. The best performance comes from ED+CM model, the error reduction rates of 41% and 17.3% compared to NV model and CM model respectively.

5. Conclusions

For robust spoken dialogue systems, sentence verification is indispensable to verify out-of-domain or bad quality

utterances. In this paper, novice sentence-level information of edit operation sequence is used in the proposed sentence verification method. The edit operation sequence generated from error-tolerant language model is used to correct the errors in the hypothetical concept sequence. The less reliable concepts tend to be removed in the model. Thus edit operation sequence is taken as sentence-level evidence for evaluating the quality of system's interpretation. Besides, a sentence verification criterion is also proposed in order to improve efficiency and correctness of spoken dialogue systems. The criterion requires 100% precision and 60% recall of concept sequence to prevent accepting misunderstood utterance but allow recovering the lost information in the following turns. Using support vector machine as the classifier, the proposed sentence verification reduces 17.3% errors comparing to the verification using acoustic confidence measure.

6. Acknowledgements

This paper is a partial result of Project A321XS1A20 conducted by ITRI under sponsorship of the Ministry of Economic Affairs, R.O.C.

7. References

- [1] Sukkar, R. A. and Lee, C.-H., "Vocabulary independent discriminative utterance verification for nonkeyword rejection in subword based speech recognition," *IEEE Trans. Speech and Audio Processing*, 4(6):420-429, 1996.
- [2] Rueber, B., "Obtaining confidence measures from sentence probabilities," *Proc. Eurospeech'97*, pp. 739-742, 1997.
- [3] Lin, Y.-C. and Wang, H.-M., "Probabilistic integration of multiple confidence measures and context information for concept verification," *Proc. ICASSP'02*, pp. 229-232, 2002.
- [4] Hazen, T. J., Seneff, S., and Polifroni, J., "Recognition confidence scoring and its use in speech understanding systems," *Journal of Computer Speech and Language*, Vol. 16, pp. 49-67, 2002.
- [5] Kawahara, T., Lee, C.-H., and Juang B.-H., "Flexible speech understanding based on combined key-phrase detection and verification," *IEEE Trans. Speech & Audio Processing*, 6(6):558-568, 1998.
- [6] Wang, H.-M. and Lin, Y.-C., "Error-tolerant spoken language understanding with confidence measuring," *Proc. ICSLP'02*, pp. 1625-1628, 2002.
- [7] Ma, C., Randolph, M., Drish, J., "A support vector machines-based rejection technique for speech recognition," *Proc. of ICASSP'01*, pp.381-384, 2001.
- [8] Seneff, S., Chuu, C., and Cyphers, D. S., "Orion: from on-line interaction to off-line delegation," *Proc. ICSLP'2000*, pp. 142-145, 2000.
- [9] Hsu, W.-T., Wang, H.-M. and Lin, Y.-C., "The design of a multi-domain Chinese dialogue system," *Proc. ICSLP'2000*, pp. 307-310, 2000.
- [10] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J., *Classification and regression trees*. Chapman & Hall, New York, 1984.
- [11] Chang, C.-C. and Lin, C.-J., "LIBSVM: a library for support vector machines," 2002. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>