Semantic Information and Derivation Rules for Robust Dialogue Act Detection in a Spoken Dialogue System

Wei-Bin Liang[1] **Chung-Hsien Wu**[2]
Department of Computer Science and Information Engineering
National Cheng Kung University
Tainan, Taiwan
[1]liangnet@gmail.com
[2]chunghsienwu@gmail.com

Chia-Ping Chen
Department of Computer Science and Engineering
National Sun Yat-sen University
Kaohsiung, Taiwan
cpchen@mail.cse.nsysu.edu.tw

Abstract

In this study, a novel approach to robust dialogue act detection for error-prone speech recognition in a spoken dialogue system is proposed. First, partial sentence trees are proposed to represent a speech recognition output sentence. Semantic information and the derivation rules of the partial sentence trees are extracted and used to model the relationship between the dialogue acts and the derivation rules. The constructed model is then used to generate a semantic score for dialogue act detection given an input speech utterance. The proposed approach is implemented and evaluated in a Mandarin spoken dialogue system for tour-guiding service. Combined with scores derived from the ASR recognition probability and the dialogue history, the proposed approach achieves 84.3% detection accuracy, an absolute improvement of 34.7% over the baseline of the semantic slot-based method with 49.6% detection accuracy.

1 Introduction

An intuitive framework for spoken dialogue system (SDS) can be regarded as a chain process. Specifically, the automatic speech recognition (ASR) module accepts the user's utterance U_t and returns a string of words $\mathbf{W}_t$ The spoken language understanding (SLU) module converts $\mathbf{W}_t$ to an abstract representation of the user's dialogue act (DA). The dialogue management (DM) module determines the user's dialogue act A_t^* and accordingly decides the current act of the system. The system DA is converted to a surface representation by natural language generation in the textual form, which is passed to a text-to-speech synthesizer for speech waveform generation. The cycle repeats when the user responds with a new utterance. Clearly, one can see that the inference of the user's overall intention via DA detection is an important task in SDS.

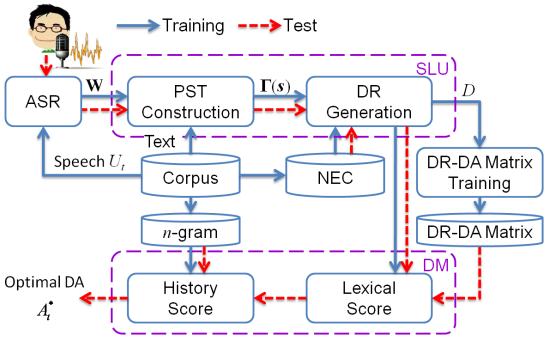

Figure 1: Details of the SLU and DM modules.

Figure 1 depicts the training and test phases of the SLU module and the DM module in our system. The dataflow for training and testing are indicated by blue arrows and red arrows, respectively. The input word sequences are converted to partial sentence trees (PST) (Wu and Chen, 2004) in the PST Construction block. The derivation rule (DR) Generation block extracts derivation rules from the training text. The DR-DA matrix is created after clustering the sentences into different dialogue acts (DAs), counting the occurrences the DRs in DA, and introducing an entropy-based weighting scheme (Bellegarda, 2000). This matrix is pivotal in the computation of the lexical score. Finally, the lexical, the history, and the ASR scores are combined to decide the

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 603–608,
Portland, Oregon, June 19-24, 2011.

optimal dialogue act, and a proper action by the system is taken. In our system, not only the clean text data but also the noisy ASR output data are used in order to take the error-proneness of ASR output into account. Furthermore, a predefined keyword list is used and the keyword tokens are replaced by the corresponding named entity classes (NEC) in order to obtain a compact feature set.

2 Models for Dialogue Act Detection

Referring to the SDS depicted in Figure 1, the DA detection can be formulated as follows. At turn t, the most likely DA is determined by

$$A_t^* = \arg\max_{A \in \Omega} Pr(A|U_t, H_t), \quad (1)$$

where U_t is the user's utterance, H_t is the dialogue historical information, and $\Omega = \{A_1, \ldots, A_q\}$ is the set of DAs. Using the maximum approximation for summation, (1) can be written as

$$\begin{aligned} A_t^* &= \arg\max_{A \in \Omega} \sum_{\mathbf{W}} Pr(A, \mathbf{W}|U_t, H_t) \\ &\approx \arg\max_{A \in \Omega} \max_{\mathbf{W}} Pr(A, \mathbf{W}|U_t, H_t) \\ &= \arg\max_{A \in \Omega, \mathbf{W}} Pr(\mathbf{W}|U_t, H_t) Pr(A|\mathbf{W}, U_t, H_t), \end{aligned} \quad (2)$$

where $\mathbf{W}$ is the ASR output. Since the ASR output is independent of H_t given U_t, the ASR-related first term in (2) can be re-written as

$$Pr(\mathbf{W}|U_t, H_t) = Pr(\mathbf{W}|U_t) \propto f(\mathbf{W}, U_t), \quad (3)$$

where the function $f(\mathbf{W}, U_t)$ is introduced as the ASR score function. In addition, assuming that the information provided by U_t is completely conveyed in $\mathbf{W}$, we can approximate the second term in (3) by the product of two functions

$$\begin{aligned} Pr(A|\mathbf{W}, U_t, H_t) &= Pr(A|\mathbf{W}, H_t) \\ &\propto g(A, \mathbf{W})\, h(A, H_t), \end{aligned} \quad (4)$$

where $g(A, \mathbf{W})$ is introduced as the lexical score function, and $h(A, H_t)$ is introduced as the history score function. Thus, (3) can be re-written as

$$A_t^* \approx \arg\max_{A \in \Omega, \mathbf{W}} f(\mathbf{W}, U_t)\, g(A, \mathbf{W})\, h(A, H_t). \quad (5)$$

In Sections 3 and 4, we specify and explain how the scores in (5) are computed.

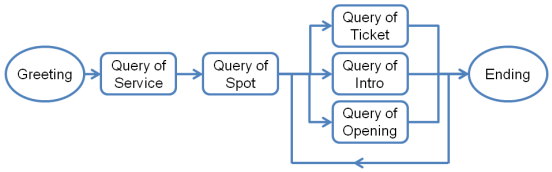

Figure 2: An example of a dialogue management module using n-gram model for dialogue act sequence in the domain of historic spot.

3 ASR Score and History Score

For the ASR score, we use the conventional recognition probability of the ASR recognition model. For the history score, similar to the schemes used in (Hori et al., 2009c; Hori et al., 2009b; Hori et al., 2009a), a back-off bi-gram model for DA sequence is estimated from the data collected by the SDS. The estimated bi-gram model is used to calculate the history score. That is,

$$h(A, H_t) = Pr(A_t = A \mid A_{t-1}). \quad (6)$$

Essentially, (6) is based on a Markov model assumption for the chain of the dialogue acts. Figure 2 shows an example of dialogue controlling model of an SDS. In this example, each state represents a DA. A dialogue begins with the greeting state and ends with the ending state. During a session, a user can inquire the system about the provided services and then choose one service to continue (e.g., the loop-back connection in Figure 2).

4 The Lexical Score Function

The main challenge of this system is the computation of the lexical score $g(A, \mathbf{W})$. In this paper, we propose a novel data-driven scheme incorporating many techniques.

4.1 Construction of Partial Sentence Tree

In an SDS, it is often beneficial to define a set of keywords $\mathcal{K}$, and a set of non-keywords $\mathcal{N}$. Each word $w \in \mathcal{K}$ should be indicative of the DA of the sentence. The set of sentences $\mathcal{S}$ containing at least one keyword in $\mathcal{K}$, can be represented as $\mathcal{S} = \mathcal{N}^* (\mathcal{K}\, \mathcal{N}^*)^+$, where $\mathcal{K}^+$ means a string of one or more words in $\mathcal{K}$. Given a sentence $s \in \mathcal{S}$, a partial sentence is formed by keeping all the keywords in s and some of the non-keywords in s. These

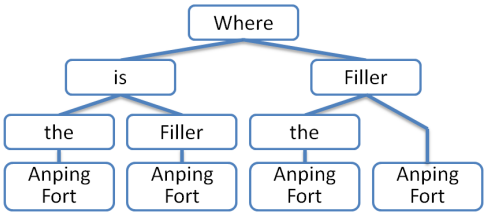

Figure 3: Construction of the partial sentence tree for the sentence *Where is the Anping-Fort*.

partial sentences can be compiled in a tree, called the partial sentence tree (PST) and denoted as $\mathcal{T}(s)$. The motivation for using PST is to achieve robust DA detection as the ASR module could be error-prone in adverse environments. In addition, words that are not confidently recognized are replaced by a special non-keyword token called *Filler*. Specifically, we compute the z-score (Larsen and Marx, 2000) of each word w in the ASR output. Figure 3 illustrates the PST for the sentence$s$: *Where is the Anping-Fort*. There are two keywords *Where* and *Anping-Fort* and two non-keywords *is* and *the*. Note that with 2 non-keywords in the original sentence s, we have $2^2 = 4$ partial sentences in the PST $\mathcal{T}(s)$.

4.2 Extraction of the Derivation Rules

After text processing, a sentence s is parsed by the statistical Stanford parser (S-parser) (Levy and Manning, 2003). Let the grammar of the S-parser be denoted as a 5-tuple $G = (\mathcal{V}, \Sigma, \mathcal{P}, S, D)$ where $\mathcal{V}$ is the variable (non-terminal) set,Σ is the terminal symbol set,$\mathcal{P}$ is the production rule set, S is the sentence symbol, and D is a function defined on $\mathcal{P}$ for rule probability (Jurafsky and Martin, 2009). A derivation rule is defined to be a derivation of the form $A \rightarrow B \rightarrow w$ where $A, B \in \mathcal{V}$ and $w \in \Sigma$. The parsing result of the exemplar sentence s represented in the parenthesized expression is shown in Figure 4. From the parsing result, four DRs are extracted. Essentially, we have one DR for each lexical word in the sentence. Totally, given a corpus, l rules are extracted and defined as $\mathcal{D} = \{R_1, R_2, \ldots, R_l\}$.

Based on PST $\mathcal{T}(s)$ and DR set $\mathcal{D}$, a vector representation $v(s)$ for sentence s can be constructed according to the DRs used in $\mathcal{T}(s)$. That is

$$v_i(s) = \begin{cases} 1, & \text{if } R_i \in \mathcal{T}(s) \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

Parse Result

```
(Root
 (SINV
  (FRAG
   (WHADVP (WRB Where)))
  (VP (VBZ is))
  (NP (DT the) (NNP Anping-Fort))))
```

Derivation Rule

DR1: WHADVP (WRB Where)
DR2: VP (VBZ is)
DR3: NP (DT the)
DR4: NP (NNP Anping-Fort)

Figure 4: The parse result (left) and the extracted derivation rules (right) for the exemplar sentence s.

For example, $v(s) = [1\,0\,1\,0]^T$ means that there are four derivation rules, of which R_1 and R_3 are used in $\mathcal{T}(s)$. The motivation for using DRs instead of the lexical words is to incorporate the part-of-speech (POS) tags information. POS tags are helpful in the disambiguation of noun-and-verb homonyms in Chinese. Moreover, the probabilistic nature of the S-parser renders the DRs extracted from the parsing results quite robust and consistent, even for the error-prone ASR output sentences.

4.3 Generation of Dialogue Acts

The basic idea of data-driven DA is to cluster sentences in the set and identify the clusters as formed by the sentences of the same DA. In this work, the *spectral clustering algorithm* (von Luxburg, 2007) is employed for sentence clustering. Specifically, suppose we have n vectors represented as $\mathcal{C} = \{v_k \triangleq v(s_k),\ k = 1, \ldots, n\}$ converted from sentences according to (7). From $\mathcal{C}$, we construct an $n \times n$ similarity matrix M, in which each element $M_{kk'}$ is a symmetric nonnegative distance measure between v_k and $v_{k'}$. In this work, we use the cosine measure. The matrix M can be regarded as the adjacency matrix of a graph G with node set $\mathcal{N}$ and edge set $\mathcal{E}$, where $\mathcal{N}$ is 1-to-1 correspondent to the set $\mathcal{C}$, and $\mathcal{E}$ corresponds to the non-zero entries in M. The normalized Laplacian matrix of M is

$$L \triangleq I - D^{-\frac{1}{2}} M D^{-\frac{1}{2}}, \quad (8)$$

where D is a diagonal matrix with entries

$$D_{kk'} = \delta_{kk'} \sum_{j=1}^{n} M_{kj}. \quad (9)$$

It has been shown (von Luxburg, 2007) that the multiplicity of the eigenvalue 0 for L equals the number of disjoint connected components in G. In our implementation, we find the q eigenvectors of the normalized Laplacian matrix of M of the smallest

eigenvalues. We put these eigenvectors in an $n \times q$ orthogonal matrix Q, and cluster the row vectors to q clusters. Each cluster correspond to a data-driven DA A_j , and the n sentences are classified according to the cluster they belong to.

In order to use the DRs in a PST as a knowledge source for DA detection, we essentially need to model the relationship between the random DA and the random DR. Denote the random DA by X and the random DR by Y. Given a text corpus, let n_{ij} be the accumulated count that R_i occurs in a sentence labeled as A_j. From n_{ij}, the conditional probability of $Y = A_j$ given $X = R_i$ can be defined as

$$\gamma_{ij} = \hat{p}(Y = A_j | X = R_i) \triangleq \frac{n_{ij}}{\sum_{j'=1}^{q} n_{ij'}}, \quad (10)$$

where $j = 1, \ldots, q$. The normalized entropy for the conditional probability function (10) is

$$\epsilon_i = -\frac{1}{\log q} \sum_{j=1}^{q} \gamma_{ij} \log \gamma_{ij}. \quad (11)$$

From (10) and (11), a matrix Φ can be constructed by $\Phi_{ij} = (1 - \epsilon_i)\gamma_{ij}$. We call Φ the derivation-rule dialogue-act (DR-DA) matrix, in which each row corresponds to a derivation rule and each column corresponds to a dialogue act.

4.4 Distance Measure

In our system, the lexical score $g(A, \mathbf{W})$ in (5) is further broken into two terms

$$g(A, \mathbf{W}) \approx g_R(A, s) g_N(A, \mathbf{W}) \quad (12)$$

where $g_R(A, s)$ is called the DR score and $g_N(A, \mathbf{W})$ is called the named entity score. Note that s denotes the sentence after text processing. The cosine distance measure is employed for the derivation rule score,

$$g_R(A = A_j, s) = \max_{\sigma \in \mathcal{T}(s)} \frac{\mathbf{b}_\sigma^T \mathbf{a}_j}{|\mathbf{b}_\sigma||\mathbf{a}_j|} \quad (13)$$

where $\mathbf{b}_\sigma^T$ is the vector representation (using the coordinates of the DRs) of a partial sentence σ in $\mathcal{T}(s)$, and $\mathbf{a}_j$ is the j^{th} column vector in the DR-DA matrix Φ. For the named entity score, we use the approximation

$$g_N(A, \mathbf{W}) = \prod_k \nu(A, \alpha_k) \quad (14)$$

NEC/SC	Name entities/Words
City	Tainan, Taipei, Kaohsiung
Spot	Anping-Fort, Sun-Moon Lake
Greeting	Welcome, Hello
Ending	Thanks, Bye

Table 1: Examples of named entity classes (NEC) and semantic classes (SC)

where α_k is the k^{th} named entity in $\mathbf{W}$. Note that $\nu(A, \alpha)$ is estimated from a training corpus by relative frequencies.

5 Experiments and Discussion

To evaluate the proposed method of dialogue act detection for robust spoken dialogue system, we adopt the commonly-used Wizard-of-Oz approach (Fraser and Gilbert, 1991) to harvest the Tainan-city tour-guiding dialogue corpus in a lab environment and experiment with simulated noisy ASR results. The details are given in this section. Two types of data from different sources are collected for this work. The first type of data, called A-data, is a travel information data set harvested from the databases available on the web, e.g., Wikipedia and Google Map. A-data consists of $1,603$ sentences with 317 word types. The second type of data, called Q-data, is the edited transcription of a speech data set simulating human-computer dialogues in a lab environment. Q-data is intended for the system to learn to handle the various situations, e.g., misunderstanding the user's intention. It consists of 144 dialogues with $1,586$ utterances. From the Q-data, 28 named entity classes and 796 derivation rules were obtained from the S-parser. Table 1 gives some examples of the selected NECs and semantic classes.

5.1 Experimental Conditions

A Mandarin speech recognition engine was realized using the HTK (Young et al., 2006), which is commonly used in research and development. For speech features, 39 dimensions were used, including 12 dimensions of mel-frequency cepstral coefficients (MFCCs), one dimension of log energy, and their delta and acceleration features. In total, the acoustic models are composed of 153 subsyllable and 37 particle models (e.g., EN, MA, OU) based

number of DA types	37	38	39
detection accuracy	82.7	84.3	77.2

Table 2: Detection accuracies with varying numbers of DA types.

on Hidden Markov Model (HMM) with 32 Gaussian mixture components per state. For the language model, SRILM toolkit (Stolcke, 2002) was employed to estimate a bi-gram model with the Q-data. The average word accuracy of the ASR module is 86.1% with a lexicon of 297 words. Note that the vocabulary size is small due to a limited domain. 5-fold cross validation method was utilized for system evaluation.

As shown in Table 2, one can see that 38 DA types achieve the best performance for the proposed detection model. Therefore, we use 38 DA types ($q = 38$) in our system. Note that some exemplar DAs are shown in Figure 2.

5.2 Incremental Evaluation

We incrementally add techniques in our SDS until the complete proposed overall system is implemented, to observe the effect of these techniques. The detection accuracies are shown in Table 3. In this table, the third column (ASR) represents the results of the experiment using the ASR transcripts directly. The fourth column (REF) uses the reference transcripts, so it represents the case with perfect ASR. The first (40%-sim) and second (60%-sim) column represents the simulation where 40% and 60% of the words in the reference transcripts are retained, respectively. There are five sets of experiments summarized in this table. For the baseline, each keyword corresponds to a coordinate in the vector representation for a sentence. The results are shown in the first row (baseline). In the second set of experiments (NEC), the keywords are replaced by their NEC. In the third set of experiments (PST), the PST representation for a sentence is used. In the fourth set of experiments (DR), the derivation rule representation of a sentence is used. Finally, the entropy-normalized DR-DA matrix is used to represent sentences, and the results are shown in the last row (DR-DA). There are strong improvements when NEC (from 49.6% to 56.8%) and PST (from 56.8% to 76.2%) representations are introduced. Moreover,

	40%-sim	60%-sim	ASR	REF
baseline	17.2	32.6	49.6	60.9
NEC	22.4	36.8	56.8	76.9
PST	29.8	49.2	76.2	91.1
DR	26.3	48.0	81.6	92.1
DR-DA	26.3	47.4	82.9	93.3

Table 3: Detection accuracies of cascading components for the lexical score.

value of λ_L	0.5	0.6	0.7	0.8
Accuracy (%)	84.3	84.6	85.1	84.9

Table 4: Evaluation on different weighted product fusion

the DR and DR-DA representations also lead to significant improvements, achieving 81.6% to 82.9%, respectively. For the other conditions of 40%-sim, 60%-sim, and REF, similar improvements of using NEC and PST are observed. Using DR-DA, however, suffers from performance degradation when the keywords are randomly discarded.

5.3 Evaluation on the Weighting Scheme

We examine the effect of different weighted product fusion and rewrite the formulation in (5) as

$$A_t^* \approx \arg\max_{A \in \Omega, \mathbf{W}} [f(\mathbf{W}, U_t) g(A, \mathbf{W})]^{\lambda_A} [h(A, H_t)]^{\lambda_L} \quad (15)$$

where λ_A is the weight for the ASR score and the lexical score, λ_L is the weight of the history score, and $\lambda_A + \lambda_L = 1$. Table 4 shows the results that history information will effect on the DA detection, because it was estimated by the dialogue turns that captured the user behaviors.

6 Conclusions

In this paper, a noise-robust dialogue act detection using named entity classes, partial sentence trees, derivation rules, and entropy-based dialogue act-derivation rule matrix is investigated. Data-driven dialogue acts are created by the spectral clustering algorithm, which is applied on the vectors of sentences represented by the derivation rules. Our spoken dialogue system benefits when the proposed components are integrated incrementally. For the fully integrated system, we find that the proposed approach achieves 84.3% detection accuracy.

References

J. Bellegarda. 2000. Exploiting latent semantic information in statistical language modeling. *Proceedings of the IEEE*, 88:1279–1296.

N. Fraser and G. N. Gilbert. 1991. Simulating speech systems. *Computer Speech and Language*, 5(1):81–99.

C. Hori, K. Ohtake, T. Misu, H. Kashioka, and S. Nakamura. 2009a. Recent advances in wfst-based dialog system. In *Proc. INTERSPEECH*, pages 268–271.

C. Hori, K. Ohtake, T. Misu, H. Kashioka, and S. Nakamura. 2009b. Statistical dialog management applied to wfst-based dialog systems. In *Proc. IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 4793–4796.

C. Hori, K. Ohtake, T. Misu, H. Kashioka, and S. Nakamura. 2009c. Weighted finite state transducer based statistical dialog management. In *Proc. ASRU*.

D. Jurafsky and J. H. Martin. 2009. *Speech and Language Processing, 2nd Edition*. Pearson Education.

R. J. Larsen and M. L. Marx. 2000. *An Introduction to Mathematical Statistics and Its Applications, 3rd Edition*. ISBN: 0139223037.

R. Levy and C. Manning. 2003. Is it harder to parse chinese, or the chinese treebank? In *Proc. Annual Meeting of ACL*, pages 439–446.

A. Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Proc. International Conference on Spoken Language Processing*, pages 901–904.

U. von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and Computing*, 17(4).

C.-H. Wu and Y.-J. Chen. 2004. Recovery from false rejection using statistical partial pattern trees for sentence verification. *Speech Communication*, 43(1-2):71–88.

Steve J. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. 2006. *The HTK Book Version 3.4*. Cambridge University Press.