Multimed Tools Appl (2017) 76:11377–11390
DOI 10.1007/s11042-016-3724-4

Two-stage multi-intent detection for spoken language understanding

Byeongchang Kim[1] · Seonghan Ryu[2] ·
Gary Geunbae Lee[2]

Received: 25 November 2015 / Revised: 5 March 2016 / Accepted: 27 June 2016 /
Published online: 6 July 2016

Abstract This paper presents a system to detect multiple intents (MIs) in an input sentence when only single-intent (SI)-labeled training data are available. To solve the problem, this paper categorizes input sentences into three types and uses a two-stage approach in which each stage attempts to detect MIs in different types of sentences. In the first stage, the system generates MI hypotheses based on conjunctions in the input sentence, then evaluates the hypotheses and then selects the best one that satisfies specified conditions. In the second stage, the system applies sequence labeling to mark intents on the input sentence. The sequence labeling model is trained based on SI-labeled training data. In experiments, the proposed two-stage MI detection method reduced errors for written and spoken input by 20.54 and 17.34 % respectively.

Keywords Spoken dialog system · Spoken language understanding · Multi-intent detection

1 Introduction

A spoken dialog system (SDS) provides a dialog interface between human and computer. Especially, in immersive multimedia environments, the spoken dialog interface is indispensable for users without any keyboard or mouse like real world. In general, an SDS consists of

✉ Byeongchang Kim
bckim@cu.ac.kr

Seonghan Ryu
ryush@postech.ac.kr

Gary Geunbae Lee
gblee@postech.ac.kr

[1] Catholic University of Daegu, Gyeongsan, Gyeongbuk, South Korea

[2] Pohang University of Science and Technology, Pohang, Gyeongbuk, South Korea

five sequential processes: automatic speech recognition (ASR), spoken language understanding (SLU), dialog management (DM), natural language generation (NLG), and text-to-speech synthesis (TTS) [1, 5, 6, 10, 11, 18, 21]. SLU converts an input sentence into a meaning representation that can be understood by a machine. Most studies design meaning representation as a combination of an intent and named entities in a given domain [2, 7, 8, 19, 24]. These studies focus on processing simple input sentences that express only one intent. We categorize those sentences as single intent (SI)-type.

However, in the real world, users often express multiple intents (MIs) within one dialog turn. The multiple intents should be processed in the spoken language understanding, so that the subsequent process can process the intents to interact with human. We categorize the types of those sentences as MI conjunctive (MI.C) and MI non-conjunctive (MI.N) types. An MI.C sentence has multiple clauses which are concatenated with conjunctions; ie this sentence is either a compound sentence or a complex sentence. An MI.N sentence has multiple clauses that are concatenated without any conjunction; this sentence occurs when ASR fails at disambiguating the boundary between individual sentences. In summary, we categorized input sentences into three types: SI, MI.C, and MI.N (Table 1). SDS should successfully process all three types, so SLU should be able to detect one or more intent in an input sentence. We named this task MI detection (MID).

In this paper, we propose a two-stage approach to MID. The first stage is conjunction-based MID (ConjMID) which attempts to detect MIs in MI.C sentences. The second stage is sequence-labeling-based MID (SeqMID) which attempts to detect MIs in MI.N sentences. The main advantage of our approach over previous studies is that the ConjMID and SeqMID can be implemented when only SI-labeled training data are available; ie our approach requires neither collection of actual MI-labeled training data nor manual annotation of extra labels onto SI-labeled training data.

The rest of the paper is organized as follows: In the following section, we briefly introduce related work. Section 3 shows a method for Automatic Speech Recognition (ASR) error correction which is indispensable for SDS system. Section 4 describes the detailed method of our two-stage approach to MID. Section 5 demonstrates the experimental design and results. Finally, Section 6 draws conclusions.

2 Related work

In traditional natural language processing (NLP), one approach is sentence boundary disambiguation. One study reported an F_1 score of 98.37 % for written input [20]; this study cannot be applied to MID, because the method cannot successfully process ASR output which does

Table 1 Examples of the three types of input sentences

Type	Example
SI	"*show me the list of action movies*" → search-program
MI.C	"*record ocn news and play big bang theory*" → record-program, play-program
MI.N	"*what is the genre of big bang theory tell me the story about it*" → search-genre, search-introduction

not have punctuation marks. In the NIST Rich Transcription Fall 2004 Evaluation (RT-04 F),[1] the minimum error score was 38.46 % for spoken input, and a later study reported an error score of 35.6 % for spoken input [14]; these studies are not sufficiently accurate to be applied to MID.

In traditional NLP research, another approach is to use clause identification. In the CoNLL-2001 shared task, the best F_1 score was 84.36 % for written input [16]. A later study reported an F_1 score of 89.04 % for written input [12]. However, these studies are neither verified for spoken input nor sufficiently accurate to be applied to MID.

In SDS research, one study approached MID as a classification problem [23]. This study limited the maximum number of intents per sentence to two, and therefore regarded a combination of double intents (DIs) as a class. The study added hidden variables to identify segments belonging to each intent. The main focus of this study was to overcome the sparsity of training data, so it used hidden-state conditional random fields that exploit shared intents across different intent combinations. However, the method requires collection of actual DI-labeled training data.

In SDS research, another study approached MID as a sequence labeling [17]. The study regarded MID as a detecting user intent indicator (UII) in an input sentence. Each UII in an input sentence represents an individual intent. The study used conditional random fields (CRFs) for sequence labeling of UII [4]. However, the method requires manual annotation of UII onto SI-labeled training data.

The proposed two-stage approach to MID consists of two sequential processes. The first stage is ConjMID. If fewer than two intents are detected in this stage, the second stage is performed. The second stage is SeqMID. If fewer than two intents are detected in this stage, traditional SI determination (SID) is performed.

We used an in-house Korean POS tagger that is based on Hidden Markov model and that was trained using Korean Sejong Corpus. We used Maximum Entropy (ME) and Conditional Random Fields (CRF) which performs either classification or sequence labeling.

One of the strength of the ME paradigm is the ability to incorporate arbitrary knowledge sources while avoiding fragmentation. So, the ME-based language models can combine n-gram features and other higher level linguistic knowledge in one unified framework [15, 22]. For this reason, we used ME to deal with words, part-of-speech(POS) and intents in our single and multiple intent detection model.

To achieve accurate multi-intent detection, we adopt a linear-chain Conditional Random Fields (CRF) classifier [4] of which classification performance has been approved. Although the CRF classifier has good performance in the classification tasks, it is important to choose the required features to be obtained from the given dataset.

3 ASR error correction

Because the input of SLU system has to be fed from ASR system, it is necessary to reduce the error of ASR system. To reduce error of ASR system, our method consists of two parts: ASR error detection and correction. First, the error detection part detects errors in the input sentence. Next, the correction part replaces or removes words that were identified as errors by the

[1] http://itl.nist.gov/iad/mig/tests/rt/2004-fall/

detection part. All the models that are needed to process the method are constructed from only the text corpus that is to train the dialog system.

3.1 ASR error detection

ASR error detection is the problem of labeling a word as an error. However, this detection cannot be treated as a supervised classification problem because no parallel corpus that includes ASR results and their transcripts is provided. The errors are essentially detected by voting from each of the detection component modules that independently identify error candidates.

POS pattern based detection An erroneous sentence may have an incorrect POS pattern, such as a grammatical error pattern. With a correct POS pattern, we could detect the erroneous words. POS pattern based error detection model includes several sentence-level POS label sequences. After tagging the ASR output sentence, the system searches for the most similar POS pattern from the model. To find the most similar POS pattern, we use the Levenshtein distance to calculate a similarity score:

$$\mathrm{s} = \frac{\mathrm{Levenshtein\ Distance(t, p)}}{\#\ of\ words\ of\ o}, \tag{1}$$

where t is a POS pattern in an ASR output, p is a POS pattern of the POS pattern model, and o is the ASR output. The lowest scored pattern among all POS patterns in the POS pattern model is selected for error part detection in the ASR output. Aligning the POS label sequence of the ASR output with the selected POS label pattern, any word that does not have a matching POS label in the POS pattern is regarded as an error candidate.

Word dictionary by POS label based detection Out of vocabulary (OOV) words in the dialog corpus have the possibility of being incorrect words. To construct a word dictionary by POS label, we consider valuable POS labels for the application: ie, nouns and verbs. If a word in the input sentence is tagged with a valuable POS, the component searches for the word in the dictionary of the tagged POS label. A word that is not present in the dictionary is regarded as an error candidate.

Word Co-occurrence based detection Word co-occurrence based detection model includes the target word and its sentence level co-occurring words, which are sorted by co-occurrence frequency.

For each word in the ASR output, a set of co-occurrence words that includes the word itself is constructed by searching the co-occurrence model. The co-occurrence score c_i is calculated by comparing the sets:

$$c_i = \sum_{j \in N} \frac{n\left(S_i \bigcap S_j\right)}{n(S_i)} \times \frac{1}{n(I)}, \tag{2}$$

where S_i is a set of co-occurrence words for word i, N is a set of ASR output words except word i, I is a set of ASR output words, and the function $n(A)$ is the number of elements of A.

The numbers of elements of S are equivalent for all i and are determined by a configuration option of the detection component. The words with comparatively low scores in relation to the other words in the ASR output may be possible errors. Then, k words with low c_i are regarded as error candidates. The number of error candidates, k, is determined by a configuration option of the detection component based on the ASR accuracy.

RNNLM based detection The RNNLM is trained to generate the word probability distribution given previous context, so the model can be used for evaluation of the appropriateness of each word in an input sentence. The equation of RNNLM score r of the word at position i in the input sentence is

$$\text{RNNLM score } r_i = p\left(w_i \middle| w_{i-1}, \ldots, w_1\right), \tag{3}$$

where the probability p is the output of the RNNLM. In the same way as the word co-occurrence based detection, k low scored words are regarded as error candidates.

3.2 Syllable prediction RNN-based error correction (SPREC)

Before the correction process, the words near the detected erroneous words are also labeled as errors because the neighbor words of the detected erroneous words also have high potential to be incorrect. The error correction method uses a syllable prediction based on RNN. Our method continuously predicts syllables at the detected error position, and the length of the prediction depends on the length of the detected error position. To select a correct word, each generated word replaces detected erroneous word and each revised sentence is evaluated by a word-level likelihood score produced by a language model based on RNN [9]. The sentence with the highest score is selected as the correction.

The syllable prediction network in our method (Fig. 1) has input layer x, syllable context layer h, predicted pronunciation layer p, and output syllable layer y. In syllable position t, the input layer to the network is $x(t)$, the syllable context layer is $h(t)$, the predicted pronunciation layer $p(t)$, and the output syllable layer is $y(t)$. Input layer $x(t)$ is formed by concatenating layer

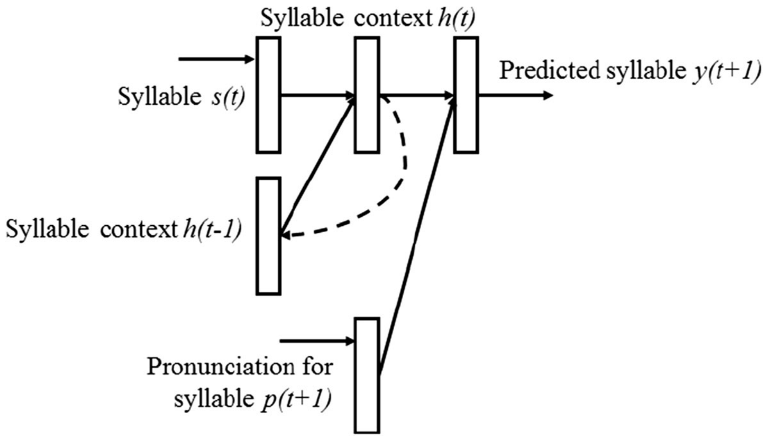

Fig. 1 RNN for syllable prediction

s(*t*) that represents a current syllable with 1-of-N coding and the previous syllable context layer *h*(*t* + *1*). To predict a syllable in position *t* + *1*, the layers are calculated as

$$\mathrm{x}(t) = s(t) + h(t{-}1) \tag{4}$$

$$h_j(t) = f\left(\sum{}_i x_i(t)u_{ij}\right) \tag{5}$$

$$y_k(t+1) = \mathrm{g}\left(\sum_j h_j(t)v_{kj} + \sum_l p_l(t+1)w_{kl}\right), \tag{6}$$

where *f* is a sigmoid activation function and *g* is a softmax function. The predicted pronunciation layer *p* is an additional layer that is included for accurate prediction and is provided in two different ways. First, if the position of the prediction *t* + *1*, provides the pronunciation information, the pronunciation layer represents a confused phoneme sequence of a syllable of the error position *t* + *1*, and the layer is calculated from the pronunciation confusion matrix [3]. Second, if the position of the prediction *t* + *1* cannot provide the pronunciation information, then the pronunciation layer is calculated by the pronunciation RNN. The network for the pronunciation RNN has input layer x_p, pronunciation context layer h_{p_n}, and predicted output pronunciation layer p_o. Input layer $x_p(t)$ is formed by concatenating layer $p_c(t)$ which represents a current syllable pronunciation with 1-of-N coding and previous pronunciation context layer $h_p(t+1)$. To predict syllable pronunciation in position *t* + *1*, the layers are calculated as

$$x_p(t) = p_c(t) + h_p(t{-}1) \tag{7}$$

$$h_{p_n}(t) = f\left(\sum{}_m x_{p_m}(t)u_{p_{mn}}\right) \tag{8}$$

$$p_o(t+1) = f\left(\sum{}_n h_{p_n}(t)v_{p_{no}}\right), \tag{9}$$

where *f* is a sigmoid activation function. The output layer p_o is activated by the sigmoid function, not the softmax function, because this layer is also an input layer to the output syllable layer *y*, so p_o should be scaled the same as the syllable context layer *h*. To train weights *u*, *v* and *w* of the syllable prediction network, a standard back-propagation algorithm is applied with the 1-of-N coding syllable vector to ensure that the output syllable layer represents the next syllable. The syllable pronunciation prediction RNN is trained independently. To train weights u_p and v_p of the pronunciation RNN, a standard backpropagation algorithm is also applied with the 1-of-N coding pronunciation vector to ensure that the output pronunciation layer represents the next syllable pronunciation. To train the RNNs of correction model, weights are initialized to small values as $-0.1 \sim 0.1$. The networks are trained in several epochs. The weights are trained with 0.1 of initial learning rate, and after each epoch the networks is tested on validation data which is the training data. If improvement on the validation data is not significant, then the learning rate is halved and start new epoch [9]. Training process is finished when no significant improvement on the validation data is again [9].

4 Two stage multi-intent detection method

The proposed two-stage approach to MID consists of two sequential processes (Fig. 2). The first stage is ConjMID. If fewer than two intents are detected in this stage, the second stage is performed. The second stage is SeqMID. If fewer than two intents are detected in this stage, traditional SI determination (SID) is performed.

We used an in-house Korean POS tagger that is based on Hidden Markov model and that was trained using Korean Sejong Corpus. We used fastCRF library which performs either classification or sequence labeling.

4.1 Conjunction-based multi-intent detection

ConjMID attempts to detect MIs from MI.C sentences. ConjMID consists of three sequential processes: generation of MI hypotheses, evaluation of MI hypotheses, and selection of the best MI hypothesis. In ConjMID, we limited the maximum number of intents per sentence to two as in the previous study [23]. In our experiments, this assumption is realistic because users rarely express more than two intents.

4.1.1 Generation of multi-intent hypotheses

ConjMID generates a set H of MI hypothesis by analyzing the conjunctions in the input sentence (Fig. 3). A conjunction entry is a sequence of word/POS pairs; e.g. "even/RB though/IN". An MI hypothesis $h \in H$ is represented as $< h_{left}, h_{conj}, h_{right} >$, where h_{left} is the left-side clause; h_{conj} is the conjunction; h_{right} is the right-side clause.

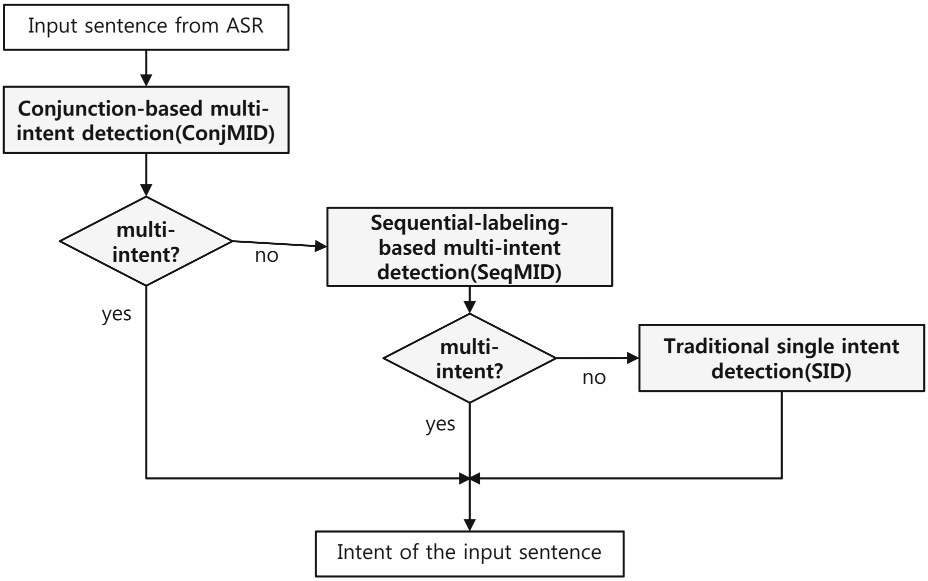
Input sentence from ASR
Conjunction-based multi-intent detection(ConjMID)
multi-intent?
no
yes
Sequential-labeling-based multi-intent detection(SeqMID)
multi-intent?
no
yes
Traditional single intent detection(SID)
Intent of the input sentence

Fig. 2 Overall process of multi-intent detection

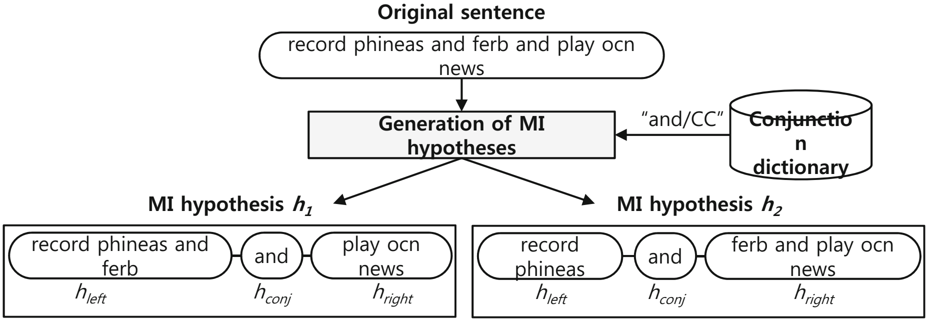

Fig. 3 Example of generation of multi-intent hypotheses

4.1.2 Evaluation of multi-intent hypotheses

ConjMID evaluates H. Given $h \in H$, traditional SID is performed on h_{left} and h_{right}. The SID score $score_{SID}(s)$ of sentence s is the confidence score of classification of s, and the score is calculated using maximum entropy model. The score of h is

$$score_{MIH}(h) = min\left\{score_{SID}\left(h_{left}\right), score_{SID}\left(h_{right}\right)\right\} \tag{10}$$

To train SID from SI-labeled training data, we used maximum entropy (MaxEnt) classifier [13]. We used two features in SID: word-n-gram and word/pos-n-gram.

4.1.3 Selection of the best multi-intent hypothesis

ConjMID compares the top-scored MI hypothesis h^*, where

$$h^* = argmax_{h \in H} score_{MIH}(h), \tag{11}$$

to the score $score_{SID}(original)$ of original sentence. ConjMID selects h^* if

$$\frac{score_{MIH}\left(h^*\right)}{score_{SID}(original)} > threshold_{conj\_mid}, \tag{12}$$

where $threshold_{conj\_mid}$ was set empirically to 1. The condition means if the score of hypothesis is greater than the score of original sentence, the hypothesis would be selected. If h^* is selected, the output is the combination of the single intent of ${h_{left}}^*$ and ${h_{right}}^*$. If the condition is not satisfied, ConjMID rejects all H.

4.2 Sequence-labeling-based multi-intent detection

SeqMID attempts to detect MIs from MI.N sentences. SeqMID adopts traditional begin, inside, and outside (BIO) tagging. Our main contribution point here is that we trained sequence labeling model when only SI-labeled training data are available.

4.2.1 Generation of multi-intent-labeled training data

MI-labeled training data can minimize the errors of MID, but we had only SI-labeled training data. So we automatically generated MI-labeled training data by concatenating all combination of two SI sentences (Fig. 4a). We generated MI.N sentences up to the square of the number of SI sentences.

4.2.2 Computation of TF-IDF values

To annotate intent-BIO tags in both original SI-labeled training data and automatically generated MI-labeled training data, we should determine whether a word is related to an intent. We regarded a word *w* to be related to intent *i* if *w* satisfies all of the following conditions:

1) *w* is not a named entity. We do this because we assumed that named-entity words are not related to intents.

2) At least one *n*-gram token *t* includes *w*, and the term frequency–inverse document frequency (TF-IDF) value, $tfidf_n(t, i, I)$ of *t* exceeds a specified threshold (Fig. 4b). We do this because we assumed that frequent and relevant words are related to intents. We computed TF-IDF values as

$$f_n(t,i) = \text{frequency of } n\text{−gram term } t \text{ in intent } i \tag{13}$$

$$tf_n(t,i) = 0.5 + \frac{0.5 \times f_n(t,i)}{max\{f_n(w,i)\ : w \in i\}} \tag{14}$$

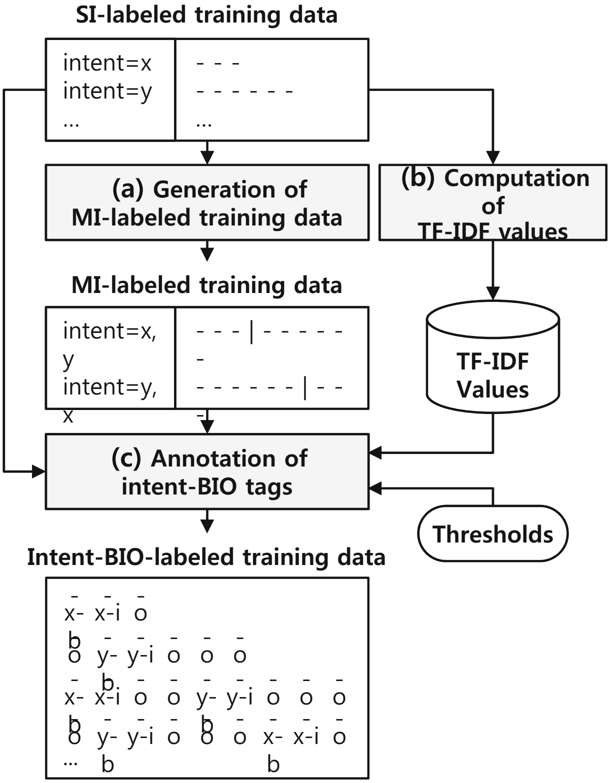

Fig. 4 Example of preparing intent-BIO-labeled training data

$$idf_n(t, I) = log \frac{|I|}{|\{i \in I \ : t \in i\}|} \tag{15}$$

$$tfidf_n(t, i, I) = tf_n(t, i) \times idf_n(t, I) \tag{16}$$

TF-IDF thresholds are different in each *n*-gram. Based on grid search technique, we found the best thresholds that minimized the errors in a development set.

4.2.3 Annotation of intent-BIO tags

We assigned intent-B or intent-I tags on to the words that are related to intent. If the word is the first word that is related to its intent, we assigned a intent-B tag to the word. Otherwise, we assigned it an intent-I tag. We assigned an O tag to the words that are not related to the intent (Fig. 4c, Table 2).

4.2.4 Extraction of features

To train SeqMID, we used CRF [4]. We used six features for sequence labeling of intent-BIO tags: word-n-gram, pos-n-gram, word/pos-n-gram, distant-n-word, is-foreign-word, and is-number.

4.2.5 Detection of Multi-intent using sequence label

The trained CRF is used in the SeqMID stage (Fig. 2). We used the same six features as described in the feature extraction section.

5 Experiments

5.1 Data

We collected a Korean-language corpus for the TV guide domain. In our TV guide domain, the size of the intent set is 33. The training and development sets consist of 5,180 and 561 SI sentences respectively. In the test set, we limited the maximum number of intents per sentence

Table 2 Example of intent-BIO tagging on "hmm what time is [the simpsons]$_{\mathrm{TITLE}}$ playing [today]$_{\mathrm{TIME}}$" (intent = search-start-time)

Word	Intent-BIO tag
hmm	O
what	search-start-time-B
time	search-start-time-I
is	search-start-time-I
the	O
simpsons	O
playing	search-start-time-I
today	O

to two as in the previous study [23]. We prepared both written and spoken test sets for three sentence types. The written test set consists of 816 sentences and the spoken test set consists of 407 sentences. To prepare the spoken test set, we used ASR in Android 4.1 Jelly Bean on a Samsung Galaxy S III Device. In our TV guide domain, the measured word error rates (WERs) were 8.89, 9.14, and 17.90 % for SI, MI.C, and MI.N sentences respectively.

To realize conjunction-based MID, we manually constructed a Korean conjunction dictionary which consists of 16 conjunction entries.

5.2 Experimental design

We defined three sentence types. So we computed weighted average by weighting type SI, MI.C, and MI.N as 0.7, 0.15, and 0.15 respectively, which are the proportions of the three sentence types in a dialog log of TV guide domain.

We compared F_1 scores of following four MID methods:

1) Baseline is traditional SID which is explained in Section 4.1.
2) ConjMID is conjunction-based MID which is proposed in Section 4.1.
3) SeqMID is sequence-labeling-based MID which is proposed in Section 4.2.
4) TwoStageMID is our final method that exploits both ConjMID and SeqMID.

5.3 Experimental results

In the written test set, each method showed the following results against Baseline (Table 3):

ConjMID had no change in SI and MI.N sentences; it reduced errors in MI.C sentences by 38.99 %. These results indicate that ConjMID can process MI.C sentences without increasing errors in the other sentences types.

SeqMID increased errors in SI sentences by 1.37 %, and reduced errors in MI.C and MI.N sentences by 37.43 and 34.31 % respectively. These results indicate that SeqMID can process both MI.C and MI.N sentences, and cause little error increase in SI sentences. We expected that SeqMID could successfully process only MI.N sentences; so it achieved more than we expected.

TwoStageMID increased errors in SI sentences by 1.37 %, but it reduced errors in MI.C and MI.N sentences by 50.77 and 34.41 % respectively. Compared to SeqMID, TwoStageMID reduced errors in MI.C sentences by 20.54 %.

These results indicate that ConjMID and SeqMID are complementary in MI.C sentences, so combining the two methods can achieve the best accuracy in MID.

Table 3 MID performance (%) on ***written*** test set for each sentence type

Method	SI	MI.C	MI.N	Avg.
Baseline	83.21	60.94	60.94	76.53
ConjMID	83.21	76.17	60.94	78.82
SeqMID	82.98	75.56	74.34	80.57
TwoStageMID	82.98	80.77	74.34	81.35

Table 4 MID performance (%) on *spoken* test set for each sentence type. (WERs are 8.89, 9.14, and 17.90 % for type SI, MI.C, and MI.N sentences respectively)

Method	SI	MI.C	MI.N	Avg.
Baseline	83.44	57.29	54.17	75.13
ConjMID	83.44	71.29	54.55	77.29
SeqMID	83.44	70.16	65.27	78.72
TwoStageMID	83.44	74.94	65.27	79.44

In the previous research, 83.6 % of intent detection accuracy was achieved on DI (Double Intent) test set using trained DI training set with class model [23]. Also, they acquired 82.1 and 80.6 % of intent detection accuracy on DI test set trained on SI (Single Intent) data plus DI data, and on SI test set trained on SI data plus DI data, respectively. However, because the experiments were performed with their own written test set, direct comparison is not meaningful.

In the spoken test set, results obtained using Baseline, ConjMID, SeqMID, and TwoStageMID showed F_1 scores of 75.13, 77.29, 78.72, and 79.44 % respectively (Table 4). In summary, TwoStageMID reduced errors in the spoken test set by 17.34 %.

6 Conclusion

In this paper, we focused on solving the MID task when only SI-labeled training data is available. First we defined three sentence types: SI, MI.C, and MI.N. Then we proposed a two-stage approach that consists of ConjMID and SeqMID. In the first stage, the system generates MI hypotheses based on conjunctions in the input sentence, then evaluates the hypotheses and then selects the best one that satisfies specified conditions. In the second stage, the system applies sequence labeling to mark intents on the input sentence. The sequence labeling model is trained based on SI-labeled training data. In experiments, the proposed two-stage MID method reduced errors for written and spoken input by 20.54 and 17.34 % respectively.

The experimental results show the proposed method is effective in solving the MID task when only SI-labeled training data is available. We are looking for several ways to improve the performance of this method.

Acknowledgments This paper is supported by ATC(Advanced Technology Center) Program - “Development of Conversational Q&A Search Framework Based On Linked Data: Project No. 10048448”

References

1. Elmir Y, Elberrichi Z, Adjoudj R (2014) Multimodal biometric using a hierarchical fusion of a Person’s face, voice, and online signature. J Inf Process Syst 10:555–567. doi:10.3745/JIPS.02.0007
2. Hakkani-Tur D, Tur G, Heck L, Fidler A (2012) A discriminative classification-based approach to information state updates for a multi-domain dialog System. in Proc. Interspeech
3. Han D, Choi K (2007) A study on error correction using phoneme similarity in post-processing of speech recognition. in The Journal of The Korea Institute of Intelligent Transport Systems. The Korean Institute of Intelligent Transport Systems (Korean ITS, p 77–86

4. Lafferty JD, McCallum A, Pereira FCN (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. in Proc. ICML
5. Lee C, Jung S, Kim K, Lee D, Lee GG (2010) Recent approaches to dialog management for spoken dialog systems. J Comput Sci Eng 4(1):1–22
6. Lee C, Jung S, Kim K, Lee GG (2010) Hybrid approach to robust dialog management using agenda and dialog examples. Comput Speech Lang 24(4):609–631
7. Liu J, Li X, Acero A, Wang Y (2011) Lexicon modeling for query understanding. in Proc. ICASSP
8. Liu J, Pasupat P, Wang Y, Cyphers S, Glass J (2013) Query understanding enhanced by hierarchical parsing structure. in Proc. ASRU
9. Mikolov T, Karafi'at M, Burget L, Cernock'y J, Khudanpur S (2010) Recurrent neural network based language model. in INTERSPEECH, p 1045–1048
10. Noh H, Ryu S, Lee D, Lee K, Lee C, Lee GG (2012) An example-based approach to ranking multiple dialog states for flexible dialog management. IEEE J Sel Top Sign Process 6(8):943–958
11. O'Neill I, Hanna P, Liu X, Greer D, McTear M (2005) Implementing advanced spoken dialogue management in Java. Sci Comput Program 54(1):99–124
12. Ram VS, Devi SL (2008) Clause boundary identification using conditional random fields. in Proc. CICLing
13. Ratnaparkhi A, Marcus MP (1998) Maximum entropy models for natural language ambiguity resolution. Ph. D. Thesis, UPenn
14. Roark B, Liu Y, Harper M, Stewart R, Lease M, Snover M (2006) Reranking for sentence boundary detection in conversational speech. in Proc. ICASSP
15. Ronald R (1996) A maximum entropy approach to adaptive statistical language modeling. Comput Speech Lang 10:187–228
16. Sang EFTK, Déjean H (2001) Introduction to the CoNLL-2001 shared task: clause identification. in Proc. CoNLL
17. Seo H (2013) Multiple user intent understanding for spoken dialog system. MS Thesis, POSTECH
18. Vanus J, Smolon M, Martinek R, Koziorek J, Zidek J, Bilik P (2015) Testing of the voice communication in smart home care. Human-centric Comput Inf Sci 5:15. doi:10.1186/s13673-015-0035-0
19. Verma P, Singh R, Singh AK (2013) A framework to integrate speech based interface for blind web users on the websites of public interest. Human-centric Comput Inf Sci 3:21. doi:10.1186/2192-1962-3-21
20. Walker D, Clements D, Darwin M, Amtrup J (2001) Sentence boundary detection: a comparison of paradigms for improving MT quality. in Proc. MT Summit
21. Williams JD, Young S (2007) Scaling POMDPs for spoken dialog management. IEEE Trans Audio Speech Lang Process 15(7):2116–2129
22. Wu J (2002) Maximum Entropy Language Modeling with Non-Local dependencies. Ph.D. thesis, Johns Hopkins University
23. Xy P, Sarikaya R (2013) Exploiting shared information for multi-intent natural language sentence classification. in Proc. Interspeech
24. Yang Z, Levow G, Meng H (2012) Predicting user satisfaction in spoken dialog system evaluation with collaborative filtering. IEEE J Sel Top Sign Process 6(8):971–981

Byeongchang Kim He got his M.S and Ph.D. degree at Pohang University of Science and Technology (POSTECH), Pohang, South Korea. From 2004, he is an associate professor in Catholic University of Daegu, South Korea

Seonghan Ryu He is a Ph.D/M.S. candidate from 2012 in Pohang University of Science and Technology (POSTECH), Pohang, South Korea.

Gary Geunbae Lee, Ph.D Gary Geunbae Lee has been a professor at CSE department of POSTECH in Korea since 1991. He is a director of Intelligent Software (ISoft) Laboratory which focuses on natural language technology researches including spoken dialog processing, intelligent computer assisted language learning, speech synthesis and web/text mining. Professor Lee holds a Ph.D. in computer science from UCLA, and BS/MS in computer engineering from Seoul National University.