# Joint Learning of Dialog Act Segmentation and Recognition in Spoken Dialog Using Neural Networks

Tianyu Zhao[1,a) Tatsuya Kawahara[1,b)

**Abstract:** Dialog act segmentation and recognition are basic natural language understanding tasks in spoken dialog systems. This paper investigates a unified architecture for these two tasks, which aims to improve the model's performance on both of the tasks. Compared with past joint models, the proposed architecture can (1) incorporate contextual information in dialog act recognition, and (2) integrate models for tasks of different levels as a whole, i.e. dialog act segmentation on the word level and dialog act recognition on the segment level. Experimental results show that the joint training system outperforms the simple cascading system and the joint coding system on both dialog act segmentation and recognition tasks.

## 1. Introduction

Recently the burst of interactive assistants and chatbots leads to an increasing interest of dialog systems. Natural language understanding (NLU), as an important component of dialog system, is usually responsible for dialog act (DA) or dialog intent tagging, where text classification techniques are necessary. Dialog act (also speech act) is a representation of the meaning of a sentence at the level of illocutionary force [1]. For instance, a sentence *"How is the weather?"* belongs to the dialog act class *Question*. For another sentence, *"The weather is quite good today."*, if it follows a previous *Question* sentence, it should be an *Answer* to the question. Otherwise it is likely to be a *Statement*. Therefore, DA recognition requires us to understand the sentence from semantic, pragmatic and syntactic aspects, and its context plays an important role as well.

The prerequisite for DA recognition is to split a sequence of words into segments, each of which corresponds to one DA unit. Especially for spoken dialog systems, NLU is based on Automatic Speech Recognition (ASR) hypotheses or transcripts, in which we cannot make any assumption of punctuation such as periods, commas and question marks for segmentation. Therefore DA segmentation becomes essential for spoken dialog systems. As the example given in Table 1 shows, a long utterance is firstly split into two segments *"hi"* and *"my name is Erica"*, to which DA tags *"Greeting"* and *"Statement"* are assigned afterwards. DA segmentation is a sequence labeling task and an "IE" tag coding scheme is adopted to describe segment boundaries, where "I" denotes "inside" of a segment and "E" denotes the "end". While the DA segmentation is a pre-process of DA recognition, recognition of DA in the sequence helps segmentation. Thus, DA segmentation and recognition are two highly related tasks. We can expect that joint learning of these two tasks can improve the performance of models. In this paper we investigate architectures of joint learning of DA segmentation and recognition, and analyze their performances. DA segmentation is a sequence labeling task on the word level and DA recognition is a classification task on the DA segment level. Our model is flexible and can be applied to tasks of different levels.

**Table 1** DA segmentation and recognition.

| **Words** | hi | my | name | is | Erica |
|---|---|---|---|---|---|
| **Segment** | E | I | I | I | E |
| **DA** | Greeting | Statement | | | |

The rest of this paper is structured as follows. Section 2 gives a literature review. Section 3 describes a hierarchical neural network model for DA recognition and explains its ability to model sequential short texts. Section 4 focuses on joint learning of DA segmentation and recognition, and explores three models for joint tasks. Section 5 shows experimental results on three tasks using the proposed models. Lastly in Section 6 and 7 we discuss the results and give a conclusion.

## 2. Related Works

In the task of DA recognition, Shriberg et al. [2] applied decision tree using rich features and emphasized the importance of prosodic features. Stolcke et al. [1] used HMM to capture the intrinsic patterns of DA sequence. Variants of neural network have been recently used in this task. In [3], a recurrent convolutional neural network is applied to text classification, which consists of a RNN layer and a max-pooling layer over it. Ji et al. [4] proposed a latent variable RNN for modeling discourse relations between sentences. Khanpour et al. [5] investigated RNNs with different settings of hyperparameters. A hierarchical neural network was introduced by Lee and Dernoncourt [6]. It firstly uses a RNN layer or a CNN layer to generate vector representations of short

[1] School of Informatics, Kyoto University
Sakyo-ku, Kyoto 606-8501, Japan
[a] zhao@sap.ist.i.kyoto-u.ac.jp
[b] kawahara@i.kyoto-u.ac.jp

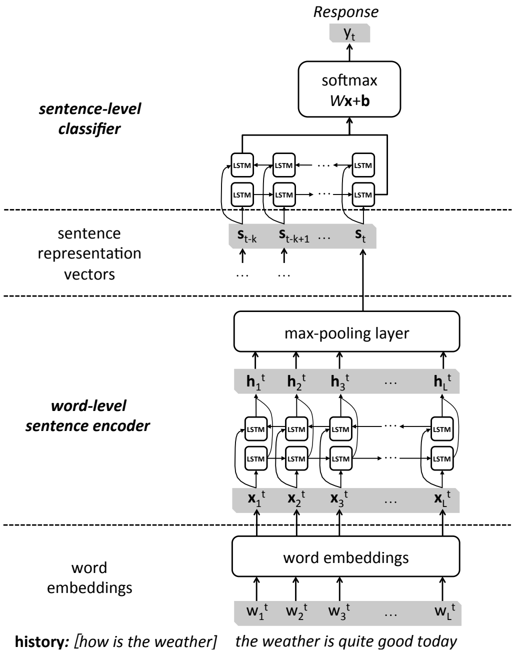

Response
y_t
softmax
Wx+b
sentence-level classifier
LSTM
sentence representation vectors
s_t-k s_t-k+1 … s_t
max-pooling layer
h_1^t h_2^t h_3^t … h_L^t
word-level sentence encoder
x_1^t x_2^t x_3^t … x_L^t
word embeddings
w_1^t w_2^t w_3^t … w_L^t
history: [how is the weather] the weather is quite good today

**Fig. 1** Hierarchical neural network: an example of input and output is given in the figure.

texts. Then a two-layer feedforward ANN takes a sequence of these vector representations to predict the probability distribution of output labels. Li and Wu [7] used gated RNN for both vector representation generation and classification in their hierarchical model. We base our work on hierarchical neural network for DA recognition and propose a unified architecture for joint DA segmentation and recognition, which is discussed in Section 4.

## 3. Hierarchical Neural Network

When we regard DA recognition as a text classification task, there are two difficulties in accurately recognizing a DA. In the first place, texts in DA recognition are often limited to a small number of words while tasks such as sentiment analysis and news topic categorization aim to classify fairly long documents and can exploit mainly n-gram models. Compared with long documents, dialog utterances have much fewer words and it is difficult to extract enough information from simple word co-occurrence features. Secondly, it is of great importance to consider contexts in DA recognition. For instance, a sentence *"the weather is quite good today"* is regarded as an *Answer* if it appears after another speaker questioning the weather. Otherwise, it is a *Statement*.

In order to address aforementioned problems, we use a hierarchical neural network for DA recognition. The hierarchical model firstly takes distributed word representation as input, which contains richer semantic information than n-gram features do. Secondly, it exploits history information to recognize DA tags of ambiguous utterances such as *"the weather is quite good today"*. The general architecture of the hierarchical neural network consists of a sentence encoder and a classifier. A sentence encoder neural network encodes a sequence of words into a vector (sentence representation vector) of a fixed length, which will be explained in Section 3.1. A classifier neural network predicts the label given representation vectors of the corresponding sentence and its preceding sentences, which will be explained in Section 3.2. The architecture of hierarchical neural network used in our work is shown in Figure 1.

### 3.1 Sentence Representation

A sentence encoder encodes a sequence of words into a fixed-length vector. By training the encoder, it obtains the ability to mine useful task-related information from a word sequence. We choose Bidirectional Long Short-Term Memory (BiLSTM) - a variant of RNN. LSTM [8] can better avoid the vanishing gradient problem compared with normal RNNs, thus it is suitable for processing information through many time steps.

Input words $\mathbf{w}^t_{1:L}$ are firstly converted to word embeddings through a lookup table in word embedding layer. Given word embeddings $\mathbf{x}^t_{1:L}$, BiLSTM outputs hidden states $\mathbf{h}^t_{1:L}$, where an output hidden state $\mathbf{h}_i$ is the concatenation of forward hidden state $\overrightarrow{\mathbf{h}}_i$ and backward hidden state $\overleftarrow{\mathbf{h}}_i$. We use a max pooling layer to extract the most informative features over time, and produce a single vector $\mathbf{s}_t$ as the encoding of word sequence $\mathbf{w}^t_{1:L}$.

### 3.2 Sequence Classification

Given sentence encoding vectors $\mathbf{s}_{t-k:t}$, we use the second neural network to predict the label of the $t$-th sentence. We use a history of length $k$ instead of the whole history since dialog act usually depends on the very late history and it also accelerates training. Again we use BiLSTM and it works in a similar manner as the BiLSTM sentence encoder does but without max pooling layer, since the latest sentence provides the most information for DA recognition. Given the final hidden state, a fully-connected layer with a softmax function outputs the predicted label $y_t$.

## 4. Joint Learning

Joint learning (multi-task learning) is an approach to learn from several related tasks in parallel so that it improves the model's ability to generalize features, and promotes its performance on the different tasks. In natural language processing (NLP), many higher-level tasks usually depend on outputs from lower-level tasks, for example named entity recognition (NER) relies on part-of-speech (POS) tagging. Hence there are many cases where models can benefit from joint learning.

Collobert and Weston [9] introduced a neural network-based joint architecture making minimal assumption of feature engineering, and also concluded three kinds of joint model, i.e. cascading features, shallow joint training, and deep joint training. A cascading model, however, does not include any joint learning procedure, and shallow joint training is actually to convert tags of different tasks into one tag. In the rest of this paper, we will name them cascading model, joint coding model, and joint training model for accuracy. Zheng et al. [10] applied a joint coding method to Chinese word segmentation and POS tagging by changing POS labels using a "BIES" tag coding scheme. Peng and Dredze [11] improved NER by word representation learnt in word segmentation task using a LSTM-CRF model. Yang et al. [12] proposed a multi-task cross-lingual model for sequence labeling tasks using RNN-CRF structures. These approaches to joint learning mostly attempt to learn shared representation of words and characters from different tasks.

**Table 2** Joint "IE" tag coding scheme, where "I" and "E" refer to "inside" and "end" respectively. We concatenate segmentation tag with DA tag to produce coded tags. For example, "E_S" denotes the end of a *Statement* segment.

| **Words** | hi | my | name | is | Erica | nice | to | meet | you |
|---|---|---|---|---|---|---|---|---|---|
| **Segmentation** | E | I | I | I | E | I | I | I | E |
| **DA** | Greeting | Statement | | | | Greeting | | | |
| **Joint tag coding** | E_G | I_S | I_S | I_S | E_S | I_G | I_G | I_G | E_G |

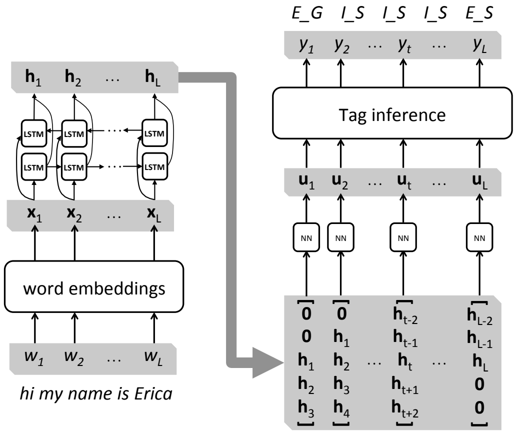


**Fig. 2** Joint coding model: an example of input and output is given in the figure.

Unlike aforementioned NLP tasks, DA segmentation and recognition deal with units of different levels, i.e. word level and DA segment level. Joint learning of these two tasks does not naturally fit in the architectures above. Previous works usually use joint coding methods. Zimmermann 2006 [13] used a hidden-event language model for sequence labeling of DA type and its boundary on word level. It also exploits prosodic features in classification. Zimmermann [14] and Quarterono et al. [15] applied conditional random field (CRF) and the former also investigated how different tag coding schemes affect the model's performance. Granell et al. [16] incorporated syntactic features and used a combination of a HMM at the lexical level and a Language Model (LM) at the DA level. Hakkani-Tür et al. [17] used a single sequence labeling LSTM model for joint semantic frame parsing, where sentence-level intent and domain tags are predicted at the last token of the sentence. An encoder-decoder-pointer framework was used for chunking in [18], where segmentation was done by a pointer network and labeling was done by a decoder LSTM.

To our knowledge, there is no previous work on neural network based joint model applied to joint learning of DA segmentation and recognition. In this section, we investigate cascading model, joint coding model, and joint training model using neural networks. In the joint coding model, joint tag coding is used to combine segmentation and recognition tags and leads to a word-level sequence labeling task. For cascading and joint training models, the proposed architectures can deal with tasks of different unit levels and the hierarchical model introduced in Section 3 is integrated to make use of contextual information.

## 4.1 Joint Coding Model

### 4.1.1 Joint Coding

In the joint coding method, one single model predicts labels of DA segmentation and recognition at the same time, so that the units of these two tasks should keep consistency. We use a joint tag coding scheme to combine labels of segmentation and recognition and make them a word-level sequential labeling problem as shown in Table 2. This allows us to use one single sequential labeling model to solve two tasks simultaneously. The proposed joint coding model is given in Figure 2.

### 4.1.2 Tag Score

A sequence of words $\mathbf{w}_{1:L}$ is firstly mapped to word embeddings $\mathbf{x}_{1:L}$, then we feed them to the following RNN layer which produces hidden states $\mathbf{h}_{1:L}$. In order to provide contextual information explicitly, when we predict a label for the $i$-th step, we also make use of hidden states of preceding steps ($\mathbf{h}_{i-1}$, $\mathbf{h}_{i-2}$, etc.) and succeeding steps ($\mathbf{h}_{i+1}$, $\mathbf{h}_{i+2}$, etc.). The concatenated vector $[\mathbf{h}_{i-2}, \mathbf{h}_{i-1}, \mathbf{h}_i, \mathbf{h}_{i+1}, \mathbf{h}_{i+2}]$ is then fed to a neural network layer that computes a tag score $\mathbf{s}_i$, where $\mathbf{s}_i \in \mathbb{R}^{|C|}$, the $t$-th element in $\mathbf{s}_i$ indicates the score of choosing the $t$-th tag at the $i$-th step, and $|C|$ is the number of tag classes.

### 4.1.3 Tag Inference

Since there often exists invalid tag sequences such as an "E_S" following an "I_Q" and we would like to penalize such invalid tag transitions, we apply a post process to compute the optimal tag sequence considering the transition probabilities by using a transition score matrix $A_{mn}$, which indicates the score of jumping from $m$-th tag to the $n$-th tag. Let $\mathbf{s}_i^t$ denotes the $t$-th element of $\mathbf{s}_i$, the score of a tag sequence $\mathbf{t}_{1:L}$ is defined as:

$$score(\mathbf{t}_{1:L}) = \sum_{i=1}^{L} (A_{t_{i-1}t_i} + \mathbf{s}_i^t), \qquad (1)$$

and we use the Viterbi algorithm to find the optimal tag sequence $\mathbf{t}^*_{1:L}$ that maximizes the sequence score:

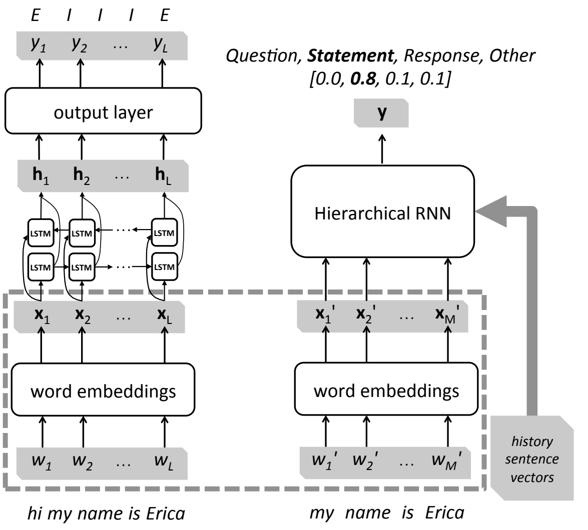
$$\mathbf{t}^*_{1:L} = \arg\max_{\forall \mathbf{t}_{1:L}} score(\mathbf{t}_{1:L}). \qquad (2)$$

## 4.2 Cascading Model and Joint Training Model

DA segmentation splits a sequence of words into segments, and DA recognition assigns a dialog act type to them. Thus, these two tasks are naturally conducted in a cascading manner. The proposed cascading model is shown in Figure 3.

The left part is a sequential labeling model for segmentation. Similar to the joint coding model, a word embedding layer and a layer of RNN are used to produce a sequence of hidden states $\mathbf{h}_{1:L}$. Then the top layer outputs predicted labels $\mathbf{y}_{1:L}$. The right part of Figure 3 uses the hierarchical neural network model introduced in Section 3. In order to provide contextual information, we also maintain a history of sentence representation vectors of previous sentences using the same hierarchical neural network.

Joint training model can be seen as a cascading model with



**(a) Segmentation component** **(b) Recognition component**

**Fig. 3** Cascading model and joint training model: in cascading model, part (a) on the left is a component for segmentation and part (b) on the right is for DA recognition. In joint training model, word embedding layers in dashed line rectangle are shared by both segmentation and recognition models. An example of input and output is given in the figure.

shared components. As shown in Figure 3, the proposed model uses only one set of word embeddings compared with the cascading one, while other task-specific parts are still separated. By updating the shared word embeddings using errors from both tasks, the model is expected to learn features from DA segmentation and recognition and improve its ability of generalization.

## 5. Experimental Evaluations

In order to evaluate our models, we conducted three sets of experiments:

- **Segmentation task**: evaluate DA segmentation performance.
- **Recognition task**: evaluate DA recognition performance given correct segments.
- **Joint segmentation and recognition task**: evaluate DA segmentation and recognition jointly, where predicted segments are given for DA recognition.

For each set of experiments, we also vary the length of history *k* for the cascading model and the joint training model.

**Table 3** Corpus statistics.

| Corpus Statistics | |
|---|---|
| # of classes | 4 |
| avg. # of segments per session | 165 |
| avg. # of segments per turn | 1.76 |
| # of training sessions | 30 |
| # of test sessions | 8 |

### 5.1 Data Set

We use a one-to-one Japanese chatting corpus collected using a conversational android ERICA [19], [20]. It is annotated with 4 DA tags (i.e. *Question*, *Statement*, *Response* and *Other*) following standards in [21]. Table 3 presents related statistics.

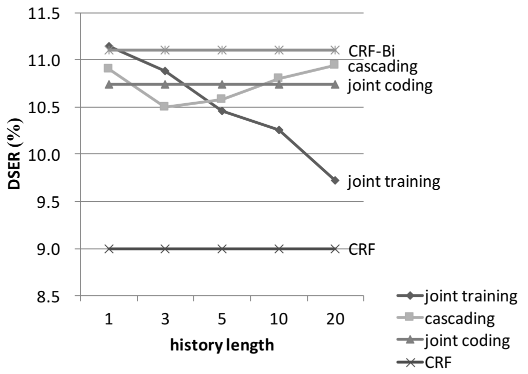


**Fig. 4** Results of the segmentation task.

### 5.2 Implementation

We implemented the proposed models with *TensorFlow*. Neural networks are trained using the Adam optimizer [22]. We use an initial learning rate of 0.0001 which decays in half when the objective loss does not decrease. A dropout layer of 0.25 dropout probability is added before every RNN layer for regularization. We choose 128 as the word embedding dimension since it works well in most NLP tasks. To find out how history length *k* affects the models, we experimented on *k* of 1, 3, 5, 10, 20. We also test CRFs for comparison in our experiments.

### 5.3 Evaluation Metrics

For the segmentation task, we use the DA Segmentation Error Rate (DSER) in [23]. The DSER is the percentage of segments that are incorrectly segmented, i.e. its left or right boundary differs from the reference boundaries. Accuracy, Macro F1 measure, and Weighted F1 measure are used for evaluation of the recognition task since it is a text classification problem. For the joint task, we use the DA Error Rate (DER) which is the same as the DSER but also considers the DA type for correctness. Table 4 demonstrates the calculation of DSER and DER.

### 5.4 Segmentation Result

In this task, a set of experiments are conducted to evaluate the model performances on DA segmentation specifically. We apply our models to segment every turn in dialogs, where one turn refers to a sequence of consecutive words uttered by one speaker without being interrupted by another speaker. We compare the cascading model, the joint coding model and the joint training model. Two CRFs are used for comparison. A simple CRF uses unigrams and bigrams of $w_{t-2}$, $w_{t-1}$, $w_t$, $w_{t+1}$, $w_{t+2}$ as features, where $w_t$ is the current word. Another CRF additionally uses the last tag as a feature and is named CRF-Bi in the following experiments.

Figure 4 shows the results of the segmentation task. The joint training model achieves 9.7% DSER at a history length of 20, which is comparable to CRF's 9.0% (without a significant difference), and it outperforms the cascading model's 10.5% at a history length of 3. The joint coding model lags behind slightly with the DSER of 10.7% and CRF-Bi gets a DSER of 11.1%.

### 5.5 Dialog Act Recognition Result

In the task of DA recognition, we evaluate the model perfor-

**Table 4** An example of DSER and DER metrics. The DSER equals 0.5 (2/4) and the DER equals 0.75 (3/4).

<table>
<tr><th>Reference</th><th>E_G</th><th>I_S</th><th>I_S</th><th>E_S |</th><th>I_Q</th><th>I_Q</th><th>E_Q</th><th>I_S</th><th>E_S</th></tr>
<tr><th>Prediction</th><th>E_G</th><th>I_S</th><th>I_S</th><th>I_S</th><th>E_S |</th><th>I_Q</th><th>E_Q</th><th>I_R</th><th>E_R</th></tr>
<tr><td><b>DSER</b></td><td>√</td><td colspan="3">×</td><td colspan="3">×</td><td colspan="2">√</td></tr>
<tr><td><b>DER</b></td><td>√</td><td colspan="3">×</td><td colspan="3">×</td><td colspan="2">×</td></tr>
</table>

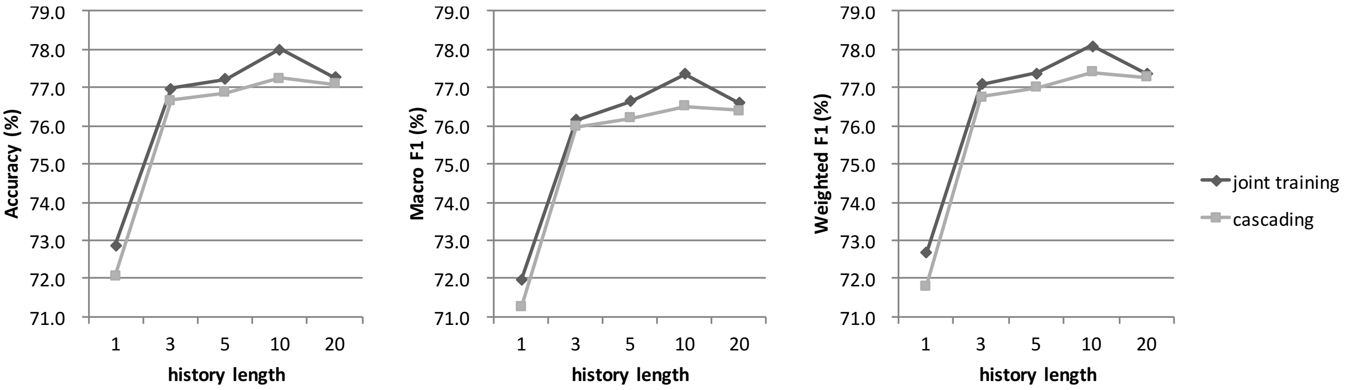


**Fig. 5** Recognition results evaluated using Accuracy, Macro F1 measure and Weighted F1 measure.

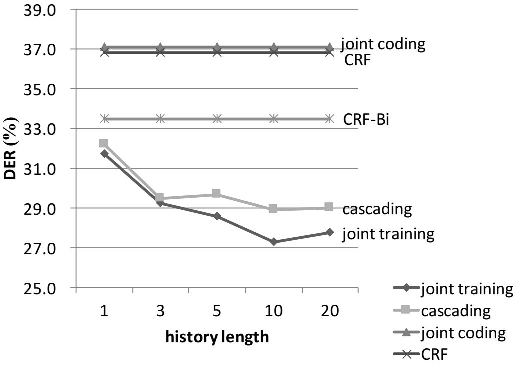


**Fig. 6** Results of the joint task.

mances of DA recognition. Therefore, we directly use ground-truth segments as inputs and predict a DA label for these segments. We only compare the cascading model and the joint training model in the recognition task because the joint coding model and CRFs are sequence labeling models and they do not naturally fit the text classification task.

As shown in Figure 5, the joint training model gets better results than the cascading model according to all three metrics. The joint training model achieves the best results of 78.0% accuracy, 77.4% Macro F1 measure and 78.1% Weighted F1 measure at the history length of 10, while the cascading model reaches 77.2%, 76.5% and 77.4% respectively. A significant improvement is obtained when we increase the history length from 1 to 3.

### 5.6 Joint Segmentation and Recognition Result

In the joint task, segmentation and recognition performances are evaluated jointly. We firstly segment each turn in dialogs into segments, then use the predicted segments as inputs of DA recognition. As in the segmentation task, we compare our proposed models and two CRFs.

Figure 6 shows the results of the joint task. The joint training model has the lowest error rate of 27.3% at the history length of 10, gaining an absolute improvement of 1.6% compared with the cascading model's 28.9%. CRF-Bi, CRF and the joint coding model got the DSER of 33.5%, 36.8% and 37.1% respectively.

## 6. Discussion

In the segmentation task, CRF using only n-gram features obtains a result of 9.0%, though there is not a statistically significant difference in performance between CRF and the joint training model, whose DSER is 9.7%. We conjecture that the result is due to that the boundaries of DA segments in Japanese are usually marked with words such as *"masu"*, *"desu"*, *"kedo"*, and considering simple n-gram features can cover most of the cases in DA segmentation. CRF-Bi, which uses a previous tag as additional feature, has a higher error rate of 11.1% compared with CRF using n-grams. It implies that extra information (previous segmentation tag and dialog act tag) may hurt the model's performance in the sense of segmentation. However, by comparing the results of the cascading joint model and the joint training model (10.5% DSER of the cascading model and 9.7% DSER of the joint training model), we can see that our joint training method is able to extract useful information from DA recognition task for segmentation and improve our models' ability of segmentation.

In the DA recognition task, we compare the best results of the cascading model and the joint training model at a history length of 10. Similar to aforementioned conclusion, joint training helps the joint training model outperform the cascading model by 0.8% in accuracy, 0.9% in Macro F1 measure and 0.7% in Weighted F1 measure. Thus we can conclude that joint training can also learn features from segmentation task to help recognize DA tags.

In the joint evaluation, we observe that even though CRFs obtain fairly good results in the segmentation task, they significantly lag behind the proposed cascading model and the joint training model. The hierarchical neural network introduced in Section 3 makes use of contexts and contributes to the improvement. Similarly, CRF-Bi gets a better result of 33.5% DER than CRF's 36.8%, which implies that contextual information (previous tag) plays an important role in DA recognition.

Finally the joint coding model has an acceptable result of segmentation but a very low performance in the joint task. There are three possible reasons: (1) The joint coding model only uses surrounding words as context instead of previous sentences, thus it fails to capture DA relations while the proposed hierarchical neural network is able to. (2) Failure in learning from the recognition task can degrade the model's performance on the segmentation task. (3) Lastly but the most important, DA recognition requires understanding of the whole segment. In the architecture of joint coding model, however, it has to predict the type of DA tag at the very beginning of a segment. Although a short context (i.e. 5 words in our experiments) is available, the model still cannot make use of complete information of the corresponding segment.

## 7. Conclusion

In this work, we explored joint learning of dialog act (DA) segmentation and recognition. To exploit contextual information in DA recognition, we introduced a hierarchical neural network architecture that incorporates history utterances. Based on the hierarchical neural network, we investigated three models for joint DA segmentation and recognition, i.e. cascading model, joint coding model, and joint training model. Our proposed models can (1) integrate the hierarchical neural network and (2) combine tasks of different levels (word level and DA segment level) in a unified architecture.

Three sets of experiments were carried out to evaluate the proposed models' performances on the segmentation task, the DA recognition task and the joint task. Experimental results showed that (1) contextual information plays an important role in DA recognition; (2) the cascading model and the joint training model outperform CRF baselines significantly (4.6% and 6.2% in DER respectively) in the joint task while having comparable performances in the segmentation task; (3) the joint training model outperforms the cascading model in all three tasks. The result demonstrates that joint training can learn useful generalized features.

## Acknowledgments

This work was supported by JST ERATO Ishiguro Symbiotic Human-Robot Interaction program (Grant Number JPMJER1401), Japan.

### References

[1] Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Van Ess-Dykema, C. and Meteer, M.: Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech, *Computational Linguistics*, Vol. 26, No. 3, pp. 339–373 (2000).

[2] Shriberg, E., Stolcke, A., Jurafsky, D., Coccaro, N., Meteer, M., Bates, R., Taylor, P., Ries, K., Martin, R. and Van Ess-Dykema, C.: Can prosody aid the automatic classification of dialog acts in conversational speech?, *Language and speech*, Vol. 41, No. 3-4, pp. 443–492 (1998).

[3] Lai, S., Xu, L., Liu, K. and Zhao, J.: Recurrent Convolutional Neural Networks for Text Classification, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 2267–2273 (2015).

[4] Ji, Y., Haffari, G. and Eisenstein, J.: A Latent Variable Recurrent Neural Network for Discourse Relation Language Models, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 332–342 (2016).

[5] Khanpour, H., Guntakandla, N. and Nielsen, R. D.: Dialogue Act Classification in Domain-Independent Conversations Using a Deep Recurrent Neural Network, *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers*, pp. 2012–2021 (2016).

[6] Lee, J. Y. and Dernoncourt, F.: Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 515–520 (2016).

[7] Li, W. and Wu, Y.: Multi-level Gated Recurrent Neural Network for dialog act classification, *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers*, pp. 1970–1979 (2016).

[8] Hochreiter, S. and Schmidhuber, J.: Long short-term memory, *Neural computation*, Vol. 9, No. 8, pp. 1735–1780 (1997).

[9] Collobert, R. and Weston, J.: A unified architecture for natural language processing: deep neural networks with multitask learning, *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008)*, ACM International Conference Proceeding Series, Vol. 307, pp. 160–167 (2008).

[10] Zheng, X., Chen, H. and Xu, T.: Deep Learning for Chinese Word Segmentation and POS Tagging, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013*, pp. 647–657 (2013).

[11] Peng, N. and Dredze, M.: Improving Named Entity Recognition for Chinese Social Media with Word Segmentation Representation Learning, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016* (2016).

[12] Yang, Z., Salakhutdinov, R. and Cohen, W. W.: Multi-Task Cross-Lingual Sequence Tagging from Scratch, *CoRR*, Vol. abs/1603.06270 (2016).

[13] Zimmermann, M., Stolcke, A. and Shriberg, E.: Joint Segmentation and Classification of Dialog Acts in Multiparty Meetings, *2006 IEEE International Conference on Acoustics Speech and Signal Processing, ICASSP 2006*, pp. 581–584 (2006).

[14] Zimmermann, M.: Joint segmentation and classification of dialog acts using conditional random fields, *INTERSPEECH 2009, 10th Annual Conference of the International Speech Communication Association*, pp. 864–867 (2009).

[15] Quarteroni, S., Ivanov, A. V. and Riccardi, G.: Simultaneous dialog act segmentation and classification from human-human spoken conversations, *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011*, pp. 5596–5599 (2011).

[16] Granell, R., Pulman, S. G. and Martínez-Hinarejos, C. D.: Simultaneous Dialogue Act Segmentation and Labelling using Lexical and Syntactic Features, *Proceedings of the SIGDIAL 2009 Conference, The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 333–336 (2009).

[17] Hakkani-Tür, D., Tür, G., Çelikyilmaz, A., Chen, Y., Gao, J., Deng, L. and Wang, Y.: Multi-Domain Joint Semantic Frame Parsing Using Bi-Directional RNN-LSTM, *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association*, pp. 715–719 (2016).

[18] Zhai, F., Potdar, S., Xiang, B. and Zhou, B.: Neural Models for Sequence Chunking, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 3365–3371 (2017).

[19] Glas, D. F., Minato, T., Ishi, C. T., Kawahara, T. and Ishiguro, H.: ERICA: The ERATO Intelligent Conversational Android, *25th IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN 2016*, pp. 22–29 (2016).

[20] Inoue, K., Milhorat, P., Lala, D., Zhao, T. and Kawahara, T.: Talking with ERICA, an autonomous android, *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 212–215 (2016).

[21] Bunt, H., Alexandersson, J., Carletta, J., Choe, J., Fang, A. C., Hasida, K., Lee, K., Petukhova, V., Popescu-Belis, A., Romary, L., Soria, C. and Traum, D. R.: Towards an ISO Standard for Dialogue Act Annotation, *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010* (2010).

[22] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, *CoRR*, Vol. abs/1412.6980 (2014).

[23] Zimmermann, M., Liu, Y., Shriberg, E. and Stolcke, A.: Toward Joint Segmentation and Classification of Dialog Acts in Multiparty Meetings, *Machine Learning for Multimodal Interaction, Second International Workshop, MLMI 2005*, Vol. 3869, pp. 187–193 (2005).