Using Context Information for Dialog Act Classification in DNN Framework

Yang Liu and **Kun Han** and **Zhao Tan** and **Yun Lei**
Applied Machine Learning, Facebook

Abstract

Previous work on dialog act (DA) classification has investigated different methods, such as hidden Markov models, maximum entropy, conditional random fields, graphical models, and support vector machines. A few recent studies explored using deep learning neural networks for DA classification, however, it is not clear yet what is the best method for using dialog context or DA sequential information, and how much gain it brings. This paper proposes several ways of using context information for DA classification, all in the deep learning framework. The baseline system classifies each utterance using the convolutional neural networks (CNN). Our proposed methods include using hierarchical models (recurrent neural networks (RNN) or CNN) for DA sequence tagging where the bottom layer takes the sentence CNN representation as input, concatenating predictions from the previous utterances with the CNN vector for classification, and performing sequence decoding based on the predictions from the sentence CNN model. We conduct thorough experiments and comparisons on the Switchboard corpus, demonstrate that incorporating context information significantly improves DA classification, and show that we achieve new state-of-the-art performance for this task.

1 Introduction

Dialog act (DA) represents a function of a speaker's utterance in either human-to-human or human-to-computer conversations. Correct identification of DAs is important for understanding human conversations, as well as for developing intelligent human-to-computer dialog systems (either written or spoken dialogs). For example, recognizing DAs can help identify questions and answers in meetings, customer service, online forum, etc. Many machine learning techniques have been investigated and shown reasonable performance for DA classification, for example, (Ang et al., 2005; Ji and Bilmes, 2005; Kalchbrenner and Blunsom, 2013; Ribeiro et al., 2015), just to name a few. Intuitively we would expect that leveraging dialog context can help classify the current utterance. For example, if the previous sentence is a question, then there is a high probability that the current sentence is a response to that question. Such context information has been explored in some previous methods, for example, hidden Markov models (HMM), conditional random fields (CRF), dynamic Bayesian networks (DBN). Given the recent success of the deep learning framework in various language processing tasks, in this work we also employ neural networks for DA classification. In fact, such models have been used in some recent studies for DA classification, e.g., (Rojas-Barahona et al., 2016; Kalchbrenner and Blunsom, 2013; Zhou et al., 2015); however, previous work has not thoroughly evaluated the use of context information for this task, and there is still a lack of good understanding about how we can use context information and how useful it is. This is the question we aim to answer in this work.

The contributions of this paper are: 1) We propose several ways to incorporate context information for DA classification over the baseline method of using convolutional neural networks (CNN) for sentence classification, including: (a) a hierarchical RNN/LSTM and CNN to model the utterance sequence in the conversation, where the input to the higher level LSTM and CNN unit is the sentence vector from the sentence level CNN model;

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2170–2178
Copenhagen, Denmark, September 7–11, 2017.

(b) a two-step approach where the predicted DA results for the previous utterances, either labels or probability distributions, are concatenated with the sentence CNN vector for the current utterance as the new input for classification; (c) sequence level decoding based on the predicted DA probabilities and the transition probabilities between DA labels. Some of these methods have not been exploited previously for this task. 2) We perform a detailed and thorough analysis of different modeling approaches and some impacting factors in the models (such as the context length, representations and quality of the predictions). This is the first study with such kind of comparisons. 3) We achieve new state-of-the-art results.

2 Related work

Previous work has investigated different machine learning techniques for DA classification such as Maximum entropy, DBN, HMM, and SVM (Ang et al., 2005; Ji and Bilmes, 2005; Venkataraman et al., 2003; Webb et al., 2005; Fernandez and Picard, 2002; Mast et al., 1996; Liu, 2006; Kral and Cerisara, 2014). Different features have been explored in these models, including lexical, syntactic features, prosodic cues, and speaker interactions. In particular, context information has been previously used in some methods. For example, some early studies used HMMs (Venkataraman et al., 2003; Stolcke et al., 2000), where the "hidden" states are the DA tags, which generate the sequence of words as observations. The observation probabilities are obtained by DA specific word-based language models, and a DA tag based n-gram language model provides the transition probabilities between the DA tags. (Ji and Bilmes, 2005; Dielmann and Renals, 2008) used DBN for sequence decoding and examined both the generative and the conditional modeling approaches. CRF, as a powerful sequence labeling method, has also been widely used to incorporate context information for DA classification (Kim et al., 2010; Quarteroni et al., 2011; Chen and Eugenio, 2013; Dielmann and Renals, 2008). It is worth noting that (Ribeiro et al., 2015) used different configurations to capture information from previous context in the SVM classifiers, such as n-grams or DA predictions. This is similar to our work in that we also evaluate using the previous utterances, and the predicted DAs for them. However, our modeling approaches are all based on DNNs, as described in more details in Section 3, and the interaction between utterances and DA labels is modeled in the hierarchical models in a more principled way.

Recently deep learning has been widely adopted in many language processing tasks, including DA classification. Context or sequence information is also explored in this framework. For example, (Rojas-Barahona et al., 2016) proposed to use DNN for DA classification and slot filling, and evaluated on two different sets. They showed that their proposed CNN+LSTM model has negligible gain on one data set, and significant improvement on the other one for the joint DA classification and slot filling task. (Kalchbrenner and Blunsom, 2013) proposed methods for discourse decomposition, and investigated using recurrent CNN for DA classification, reporting some positive results, e.g., 2.9% improvement over the LM-HMM baseline. In this paper we propose different methods in the deep learning framework to incorporate context information. Our hierarchical LSTM and CNN method has some similarities to that used in (Rojas-Barahona et al., 2016; Kalchbrenner and Blunsom, 2013), but unlike those that focus on just one method, we propose a few approaches and perform comparisons among them for a deeper understanding of different methods and their contributing factors.

The discussions above are limited to DA classification using speech/text data. Other knowledge sources have also been used in a multimodal setting (e.g., haptic actions in (Chen and Eugenio, 2013)). In this study we just rely on textual information. Also note that in some scenarios, for example, speech conversations where transcripts are from speech recognition systems, DA segmentation is also needed. This problem has been addressed in some previous work, for example, (Lendvai, 2007; Quarteroni et al., 2011; Ang et al., 2005), which often uses a classification or sequence labeling setup for the segmentation task, or performs joint DA segmentation and classification. We use pre-segmented utterances and focus just on the DA classification task in this work.

3 DA Classification Methods

3.1 Task

Our task is to classify each utterance in a conversation into a predefined DA tag set. We use Switchboard data in our experiments (see Section 4.1 for

DA type	speaker	sentence
statement-opinion	B	I always kind of think it would be neat to be able to watch them and be there for them all the time
back-channel	A	Uh-huh
question-yes-no	B	Is that what you do?
yes-answer	A	Uh yeah
statement	A	Actually I teach my kids at home
statement	A	so I'm here all the time
summarize/reformulate	B:	Oh so they don't go to school

Table 1: An example of Switchboard conversation with the DA labels.

additional information on the data). There are different granularities of the tag sets. In this work we use 42 tags (Jurafsky et al., 1997), which has been widely used in previous studies of DA classification on this data set. Table 1 shows an example of some utterances in a Switchboard conversation. We can see that the 'answer' DA follows the 'question' one, which is quite intuitive. Our goal is thus to model such sequential information for DA classification. Again in this work we only use the transcriptions of the utterances along with the speaker information (i.e., if the current utterance is from the same or different speaker as the previous one), without any speech related features.

3.2 CNN for utterance classification

All of our methods are built based on the basic CNN sentence representation, which has been widely used recently in sentence as well as document classification (Collobert et al., 2011; Kim, 2014), therefore we first briefly describe this baseline. Figure 1 shows the context independent CNN-based classification method. Let $\mathbf{w}_{[1...n]}$ represent the word embedding sequence for a sentence with n words, where $\mathbf{w}_i \in \mathbf{R}^d$ is the d-dimensional embedding vector for the i^{th} word. A temporal convolution operation is applied to the sentence:

$$c_{[1...n]} = \tilde{\mathbf{w}}_{[1...n]} * \mathbf{f}$$

where $\tilde{\mathbf{w}}_{[1...n]}$ denotes the sequence $\mathbf{w}_{[1...n]}$ with zero padding, and f is a filter map for the convolution operation. A max pooling layer is then applied over the resulting sequence $c_{[1...n]}$ to obtain one value for the sentence. If we use l window sizes and k filters for each window, then $l \times k$ convolutional sequences are generated for each sentence, and after max pooling, we obtain a fixed-length vector $\mathbf{s}$ with a dimension of $l \times k$. This is the feature vector representation for the sentence, which is then used as the input in a multi-layer perceptron (MLP) or feedforward neural network for sentence classification. We only use one layer MLP in this work.

This baseline CNN model learns textual information in each sentence for DA classification. We can incorporate additional features into this model, for example, if the current sentence is from the same speaker as the previous one. Figure 1 shows the use of such additional features – they are concatenated with the CNN-based textural vector, and then fed to the MLP for DA classification. In the rest of the paper, when there is no confusion, we also use CNN for the cases when additional features are concatenated with the standard CNN for sentence-level representation. We use this CNN model as a baseline, and in the following will explore several methods using context information for DA classification.

3.3 Use history DA information

As discussed earlier, we expect there is valuable sequential information among the DA tags, therefore in the first approach, we combine the history DA information with the current utterance to classify its DA tag. This is represented as additional features concatenated with the CNN sentence representation, as shown in Figure 1. We evaluate different configurations in this framework.

- Use DA labels. We compare using reference and system predicted DA labels in training and testing. Note that using reference labels in testing is not a real testing setup. This is just meant to provide an upper bound and understand the performance degradation due to prediction errors.
- Use probabilities for system predictions. Instead of taking the hard decisions from the system's predictions, we evaluate using the

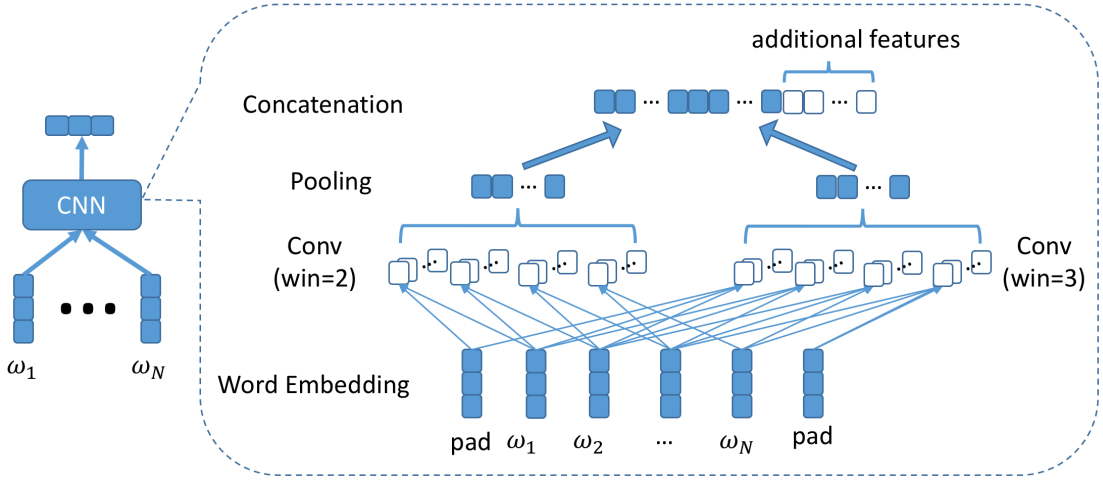

Figure 1: Baseline context-independent CNN-based DA classification method.

posterior probabilities from the system in order to capture more information.

- History length. We compare using DA information from different number of previous utterances.

Note that for most of these setups above when system's predicted DA information is used, we need to go through the following procedure:

- train a context-independent sentence CNN model
- use it to generate predictions, for training and test data
- add the corresponding history DA information in the training set to retrain a model
- add the history DA information in the test set and apply the new model

The only scenario where these steps are not required is when reference DA tags are used in both training and testing. There is one additional caveat that is worth pointing out – when generating the DA predictions for the training data, ideally we need to perform cross validation for the training set such that all the training sentences are labeled by a model trained from data that does not include this sentence, and thus we have matched information used in training and testing; however, we noticed that our model does not overfit the training data very much, and the training accuracy is not significantly different from the test accuracy, therefore we simply apply the trained CNN model to the training set itself to obtain the DA predictions for all the training sentences, and train the new model.

3.4 CNN + DA transition decoding

In this approach, we perform conversation level decoding that combines the probabilities from the context-independent CNN model and the DA tag transition probabilities. The DA classification problem can be represented as:

$$\hat{Y} = argmax P(Y|X) = argmax P(Y)P(X|Y)$$
$$= argmax P(Y) \prod_i P(x_i|y_i)$$

where Y is the DA tag sequence, and X contains the entire conversation, i.e., sequence of sentences. $P(Y)$ can be computed for the DA tag sequence (similar to word-based n-gram language model, here "words" are DA tags), and the probability of a tag given the utterance ($P(x_i|y_i)$) can be obtained from the rescaled probability from the CNN model (that is $P(y_i|x_i)$). For decoding, we can use either Viterbi decoding to find the most likely DA sequence (as shown above) or forward-backward decoding to determine the best tag for each utterance in the sequence. This model is similar to the HMM model used previously for this task (Stolcke et al., 2000), and the difference is in that the probability of a DA given the sentence is estimated by the CNN model, a discrim-

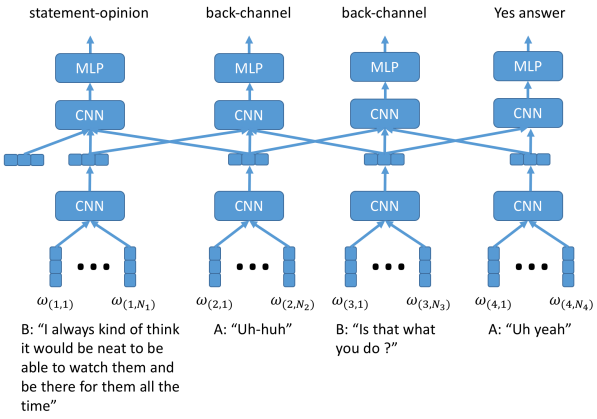

Figure 2: Hirarchical CNN: sequence CNN on top of sentence CNN for DA classification.

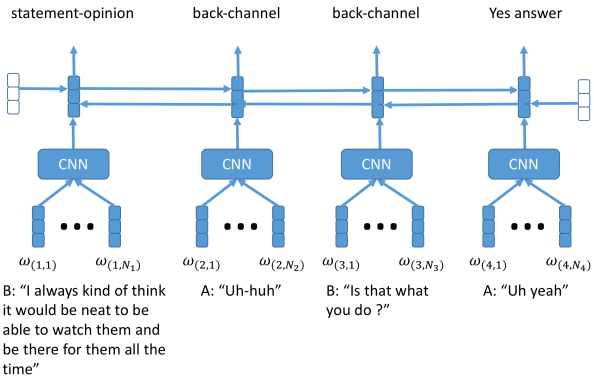

Figure 3: RNN/Bi-LSTM on top of sentence CNN for DA classification.

inative model, in contrast to the word-based language model that is a generative model.

3.5 Hierarchical model: CNN+CNN

Once we have the sentence vector representation built based on the baseline CNN model, we use another CNN to incorporate context information of an utterance for its classification. Figure 2 shows this method. The sequence of sentences is represented by a sequence of fixed length vectors $\mathbf{s}_{[1...m]}$, where m is the number of sentences in the conversation, and s_i is the vector representation for sentence i from the baseline CNN model. Similar to the CNN model for word sequence, we apply a temporal convolutional layer with different filters to $\mathbf{s}_{[1...m]}$. Different from the sentence CNN model for word sequences, here we do not perform pooling for the entire dialog sequence, as the classification task is for each sentence, not the whole conversation (sentence sequence). Instead, for each sentence, the output of every convolutional filter is concatenated to form the sentence's representation, and then an MLP is used for its classification. This approach can be thought as a hierarchical neural network, where the high level CNN is used to capture context information.

3.6 Hierarchical model: CNN+RNN

The hierarchical CNN method uses the neighboring sentences to learn the dependencies among consecutive utterances. A different method to model the sequential information is via an RNN that is intrinsically capable of learning the temporal dynamics, which is suitable for the problem. In this hierarchical model, the representation for each sentence is still learned by the CNN as in the baseline, while the dialog-level sequence information among sentences is modeled by the RNN. Here, we use bidirectional-LSTM (BLSTM) to learn the context before and after the current sentence. The left-to-right LSTM output and the one from the reverse direction are concatenated and input to a hidden layer for classification. BLSTM has been widely used recently for various sequence labeling problems (such as part-of-speech tagging, named entity recognition) and achieved state-of-the-art performance. Figure 3 shows the structure of the model. Note that the difference between these last two models and the one using history DA information is in that DA labels are not explicitly represented in these hierarchical models.

4 Experiments

4.1 Data

We use Switchboard data in our experiments. This corpus has been widely used in the community for DA classification. In this data, two people talked over the phone about a given topic for several minutes. 1155 conversations have been manually labeled with DAs. 40 conversations were held out for testing and development. However, there is no standard as to what are the test ones (it is unknown from the earliest paper using this data (Stolcke et al., 2000)). Therefore we randomly split the set into two, 20 conversations in each, with similar amount of utterances. We use one set as the development set and evaluate on the other set. As mentioned earlier, we do not use speech features, and only use textual information and speaker change feature in this study. For all the experiments, we use human transcripts. This setup is expected to be applicable to written conversations/dialogs. Ta-

ble 2 shows the basic statistics of the data.

	conversations	sentences
training	1115	196,753
test set 1	20	3,764
test set 2	20	3,771

Table 2: Data information.

4.2 Results

4.2.1 Baseline CNN

For all the DNN models, we did not tune model parameters very much. Most of the parameters were chosen based on literature or our experience with other DNN-based text classification tasks. We used pretrained embeddings (dimension 200) to initialize word vectors to use in CNN, and then update them during training.[1] To avoid overfitting, we use a dropout of 0.5. The baseline CNN uses three windows: 1, 2, and 3, and 100 filter maps for each. The output hidden layer dimension is 100. For learning, we use Adagrad with a learning rate of 0.01.

Table 3 shows the baseline classification accuracy results when no context information is used, for three setups: the baseline sentence CNN model with the pretrained embeddings, when speaker change information is added, and when no pretrained embeddings are used. We can see the slight performance change because of the added speaker change feature. When no pretrained embeddings are used, i.e., no additional information is used from other resources, there is a performance degradation of 2-3%. Note that these results are better or at least comparable to state-of-the-art performance. In fact, we also implemented a CRF tagging model for this data set, where we used bag-of-word features for each utterance, therefore the information is similar to that used in the DNN framework (but the CRF does model DA tag sequential information). This CRF model has an accuracy of about 74% for the two sets combined. The CNN model without using pretrained embeddings has worse results than the CRF system that is trained just using the Switchboard data, confirming that when using word embeddings as word representations, pretrained embeddings are beneficial when the training size is small. However, the CNN model can effectively leverage word embedding information (obtained from unlabeled data), whereas it is not straightforward to use such information in the CRF classifiers. This shows an advantage of the DNN-based method.

[1]The embeddings we used are generated based on our collected web data. We compared it to other embeddings, e.g., Senna, and found the performance difference is very small.

	set 1	set 2
CNN	74.47	76.88
+ speaker change	74.73	77.12
no pretrained embedding	71.81	74.49

Table 3: DA classification accuracy (%) when using the baseline CNN without context information.

4.2.2 Hierarchical models: CNN+CNN/RNN

For the hierarchical models described in Section 3.5 and 3.6, i.e., adding CNN and BLSTM on top of the baseline sentence CNN, we kept the same model parameters in the sentence CNN part. The dimension is 64 for both the higher level CNN and LSTM. For these sequence labeling tasks, we use stochastic gradient descent (SGD), with a learning rate of 0.01. We observed this yielded better performance than Adagrad learning. Table 4 shows the results for different setups in these two models to evaluate the impact of context information. For LSTM, we compare using LSTM and BLSTM; for CNN, we show results when using different context window sizes in the top layer.

		set 1	set 2
baseline CNN		74.73	77.12
CNN+CNN	window 2	76.2	79.16
	window 3	76.78	79.05
	window 4	77.15	79.74
CNN+RNN	BLSTM	76.91	79.71
	LSTM	76.35	79.71

Table 4: DA classification results (%) when using the hierarchical structure: sentence CNN followed by dialog sequence level CNN or RNN/BLSTM.

From the table we can see that using LSTM and CNN to model context information for DA classification is effective, both models significantly outperforming the baseline. Regarding the effect of context, in general there is slightly more gain when more context is used, as in BLSTM, or larger windows in CNN. For CNN, when we increase the window more, to beyond 4, there is no further improvement. The greatest difference comes from using context vs. not using it at all.

history	DA = ref or sys		DA representation	set 1	set 2
	training	testing			
one	ref	ref	label	78.19	80.96
	ref	sys	label	75.72	77.94
	sys	sys	label	76.78	79.26
	sys	sys	probabilities	76.62	79.45
two	ref	ref	label	78.93	81.54
	sys	sys	label	76.41	79.98
	sys	sys	probabilities	76.51	80.14
three	ref	ref	label	79.62	81.76
	sys	sys	label	76.54	80.06
	sys	sys	probabilities	76.73	79.9
baseline CNN				74.73	77.12

Table 5: DA classification results (%) when incorporating history DA information in the current utterance in the CNN method. Three factors are examined: context history length, DA representations, and where DA information is from.

4.2.3 CNN + DA prediction

As described in Section 3.3, another method to incorporate context information is to use the DAs from previous utterances. We perform a detailed analysis to examine three factors under this framework:

- context history: we use a window of up to 3, i.e., information from the previous one, two, or three utterances;
- representation of the DA information, whether it is DA label or probabilities;
- reference vs. system predicted DA labels during training and testing.

Using the reference DA labels in testing is expected to give an oracle or upper bound performance for this set of experiments. Table 5 shows the results for these setups. The predictions for the utterances are generated using the baseline CNN model, with the pretrained embeddings and speaker information (i.e., the best utterance classification model). The model parameters in the second-round CNN training (when additional history DA information is included) are the same as the baseline CNN.

From Table 5 we can see that in terms of the representation of the history DA information, using hard labels and soft predictions achieves similar performance. For model training, it is better to have matched information in training and testing. Using reference DA labels during training and system predictions in testing (second row in the results) is less effective compared to using both system predictions in training and testing. The quality of the prediction also affects the usefulness of the DA prediction information, as demonstrated by the better performance when the reference labels are used compared to using system predicted DAs, which is expected. The immediate previous utterance has the largest impact on the prediction of the current utterance (comparing to not using context at all), and adding longer context helps less. In addition, using the reference previous DA labels (ref train and ref test condition) benefits more than using system predicted DA labels when longer history is used, suggesting that more predicted DAs, when used together, become more noisy and bring less gain.

4.2.4 Overall results

Table 6 summarizes the results for different systems, including the baseline CNN model without using context information (this baseline uses pretrained embeddings and speaker change feature), and four different ways of using context: (a) predicted DA information (posterior probabilities) is combined with the current sentence's CNN-based representation; (b) applying a BLSTM on top of the sentence CNN representation; (c) hierarchical CNN that combines the current sentence's CNN representation with its neighbors; (d) sequence decoding by combining CNN posteriors with DA transition scores.

From the results, we can see the positive effect

	set 1	set 2
CNN baseline, no context	74.73	77.12
CNN + DA predictions	76.73	79.9
CNN + RNN/BLSTM	76.91	79.7
CNN + CNN	77.15	79.74
CNN prob + DA transition	76.70	79.69

Table 6: DA classification results (%) using different systems.

when context information is used. All the methods using context yield significant improvement over the baseline (statistically significant based on t-test). Comparing representing context information via the DA labels of the previous utterances vs. using the hierarchical CNN or RNN model, we see there is not much difference. This observation is somewhat different from that found in (Ribeiro et al., 2015; Kim et al., 2010) where using previous DA predictions yields more gain than adding n-gram features from the previous utterances. We believe one reason for this difference is the use of the DNN framework to model the utterance sequences. Given the current data size and the oracle performance in Table 5, we expect that when more data is available, using larger neural networks will further improve the performance. Furthermore, we want to mention that overall these results represent new state-of-the-art performance for this task ((Kalchbrenner and Blunsom, 2013) reported 73.9% accuracy using recurrent CNN, though the results are not directly comparable since they only evaluated on 19 test conversations).

4.2.5 Final remarks

As expected, our experimental results demonstrate that we can effectively incorporate context information to improve DA classification. We conducted some analyses to see what errors are corrected when we use the context models compared to the baseline results. Due to space limit, we show one positive example below where adding context changes the prediction from 'backchannel' to 'answer'.

- Example:
  - Is this a mail order parts house that specializes in parts for parts for uh old imports?
  - right

It is clear that using context can help disambiguate and better predict the DAs for the current utterance. In fact, we noticed that close to 5% of errors are correctly changed from 'back channel' to 'reply' when context information is used.

One of the most frequent errors we notice the system makes is the mislabels between 'statement' and 'statement-opinion'. To correctly identify statement-opinion DAs, we can perform some opinion or subjectivity recognition, but that is out of the scope of this study. Another frequent error is the confusion between backchannel and agreement. For example, 'right' and 'yeah' are common words for both categories, and even with context information, they are still hard to disambiguate for the current models.

Finally it is worth pointing out that our work uses an offline setting where we perform DA tagging for the entire conversation. In real world applications, an online setting may be needed; however, information from previous utterances can still be used there. In fact, most of the performance gain from incorporating context information comes from the previous utterances (e.g., the difference between the hierarchical LSTM and BLSTM is very small). Our findings about the effectiveness of context information are applicable to the online setting.

5 Conclusions

We proposed several approaches to incorporate context information in the deep learning framework for DA classification in conversations, including expanding the sentence CNN vector with the predicted DA information from previous utterances to train another model, hierarchical models based on CNN or LSTM to model the DA sequence on top of the sentence CNN representation, or dialog level decoding once the sentence CNN generates its hypothesis. Compared to the baseline using CNN for utterance classification, our proposed methods effectively leverage context information and achieve significantly better performance. We observe that there is very small difference among different approaches. Our results represent the state-of-the-art for DA classification on the Switchboard data. We conducted thorough evaluations to understand the impact of different factors, and our results shed lights on the use of context information for similar tasks. In our future work, we plan to apply these approaches to other tasks, such as intent recognition and slot filling in language understanding.

Acknowledgments

The authors thank Yandi Xia for preparing the Switchboard data, Xian Qian, Antoine Raux and Benoit Dumoulin for various discussions.

References

Jeremy Ang, Yang Liu, and Elizabeth Shriberg. 2005. Automatic dialog act segmentation and classification in multiparty meetings. In *Proc. of ICASSP*.

Lin Chen and Barbara Di Eugenio. 2013. Multimodality and dialogue act classification in the robohelper project. In *Proceddings of Sigdial*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12(Aug):2493–2537.

Alfred Dielmann and Steve Renals. 2008. Recognition of dialogue acts in multiparty meetings using a switching dbn. *IEEE Transactions on Audio, Speech and Language Processing* 16.

Raul Fernandez and Rosalind Picard. 2002. Dialog act classification from prosodic features using support vector machines. In *Proc. of Speech Prosody*.

Gang Ji and Jeff Bilmes. 2005. Dialog act tagging using graphical models. In *Proc. of ICASSP*.

D. Jurafsky, L. Shriberg, and D. Biasca. 1997. Switchboard swbd-damsl shallow-discourse-function annotation coders manual. Technical report, University of Colorado at Boulder.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent convolutional neural networks for discourse compositionality.

Su Nam Kim, Lawrence Cavedon, and Timothy Baldwin. 2010. Classifying dialog acts in one-on-one live chats. In *Proceedings of EMNLP*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proc. of EMNLP*. pages 1746–1751.

Pavel Kral and Chrisophe Cerisara. 2014. Automatic dialogue act recognition with syntactic features. In *Proceedings of LREC*.

Piroska Lendvai. 2007. Token-based chunking of turn-internal dialogue act sequences.

Yang Liu. 2006. Using svm and error-correcting codes for multiclass dialog act classification in meeting corpus. In *Interspeech*.

Marion Mast, Ralf Kompe, Stefan Harbeck, Andreas Kiebling, Heinrich Niemann, and Elmar Noth. 1996. Dialog act classification with the help of prosody. In *Proc. of ICSLP*.

Silvia Quarteroni, Alexei V. Ivanov, and Giuseppe Riccardi. 2011. Simultaneous dialog segmentation and classification from human-human spoken conversations. In *Proceedings of ICASSP*.

Eugenio Ribeiro, Ricardo Ribeiro, and David Martins de Matos. 2015. The influence of context on dialogue act recognition .

Lina M. Rojas-Barahona, Milica Gasic, Nikola Mrksic, Pei-Hao Su, Stefan Ultes, Tsung-Hsien, and Steve Young. 2016. Exploiting sentence and context representation in deep neural models for spoken language understanding.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialog act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics* .

Anand Venkataraman, Lucianna Ferrer, Andreas Stolcke, and Elizabeth Shriberg. 2003. Training a prosody based dialog act tagger from unlabeled data. In *Proc. of ICASSP*.

Nick Webb, Mark Hepple, and Yorick Wilks. 2005. Dialog act classification based on intra-utterances features. In *AAAI workshop on Spoken Language Understanding*.

Yucan Zhou, Qinghua Hu, Jie Liu, and Yuan Jia. 2015. Combining heterogeneious deep neural networks with conditional random fields for chinese dialogue act recognition. *Neurocomputing* 168.