Lexico-acoustic models with attention in Dialog Act Classification

Esra Dönmez, Christoph Schaller, Wei Zhou

Institute for Natural Language Processing (IMS)
University of Stuttgart
{st172173, st171455, st175345}@stud.uni-stuttgart.de

Abstract

Dialog act classification (DAC) is of significant importance in spoken dialog systems. Recent works have proposed several successful neural models for dialog act classification, many of which only explored the task by taking advantage of the transcripts of the audio files and building lexical models. In 2018, Ortega and Vu. [1] proposed a lexico-acoustic neural-based model for this task to utilize the acoustic information in combination with the lexical features. In their experiments, they use convolutional neural networks (CNNs) [2] to learn both the lexical and the acoustic features. Moreover, pretrained language models have been successfully used in recent years in various tasks to learn contextual embeddings from natural language input. In this paper, we experiment with both lexical and acoustic models in DAC. We implemented two models, one for the textual input and one for the acoustic input, and different ways to combine these modalities for multimodal learning in DAC. We evaluate our models on a subset of the SwDA (Switchboard Corpus) [3] and compare the results of the two models as well as the combined model. Furthermore, we conduct both quantitative and qualitative error analysis to investigate the strengths and weaknesses of these models. We also run an ablation study to examine the contributions of the individual components in our lexical model. Our results show that lexical model is sufficient to learn the features that are represented by both modalities, i.e. the acoustic model might not learn additional useful information that is not present in lexical features when combined with a powerful lexical model.

Index Terms: speech recognition, dialog act classification, spoken dialog systems, computational linguistics, multimodal learning

1. Introduction

Even before the rise of natural language processing (NLP) or machine learning, humans have envisioned to build smart spoken dialog systems to help assist them in various areas such as day-to-day life or industrial applications. Spoken language understanding (SLU) is an important part of these conversational intelligent systems. The pipeline for this traditionally consists of an automatic speech recognition (ASR) component to transcribe speech into text and an NLP component to extract meaning from the transcribed text [4]. In various domains such as music recommendation and weather information, intents and slots are designed in a domain-specific manner. Dialog acts, on the other hand, are domain-independent and a part of every dialog. Each utterance in a dialog carries a level of illocutionary act whose meaning induces an effect over the course of the dialog [1]. These are commonly referred to as dialog acts (DA). A DA can be understood as the intention of the speaker with that utterance in a given conversation, such as asking a question or making a statement. Analyzing these dialog acts can be beneficial in dialog modelling and other spoken dialog systems (SDSs). Table 1 shows some examples of these dialog acts.

Utterance	DA label
it was n't just a super bad storm	statement
Uh-huh	backchannel
it 's been it 's been cold	opinion
whereabouts is that	question
It 's very very rich	opinion
And they did n't used to be actually	statement
Oh yeah	question
but they came back later	statement
yeah	backchannel
So that 's still a real good show too	opinion
Oh	backchannel
wow eight kids	question

Table 1: *Example utterances and their corresponding dialog acts from SwDA Corpus*

In dialog systems, automatic DA tagging is used as a preprocessing step to extract the intent of the utterance that is spoken. To tag the utterances with their DAs, these systems include a DA classification component that can be formulated as a sequence of classification task. There are two main information sources that can be used in this task: lexical information that are extracted from the transcripts and acoustic information that are extracted from the audio signals. The reason why it is considered to be useful to encode information from both modalities, i.e. lexical and acoustic, is that the DAs can be highly ambiguous. This is especially the case when the context information is not available or when the transcripts provided by the ASR system do not contain punctuation marks. In such cases, acoustic information can be helpful to disambiguate between certain DAs such as between a statement and a question.

To extract acoustic cues, CNNs and recurrent neural networks (RNNs) have been proposed by several recent work. To extract lexical information, early models were developed using CNNs, RNNs or long short-term memory models (LSTMs). Although not in combined networks (i.e. lexical and acoustic) to our knowledge, pretrained language models have been found useful in learning the lexical cues in some recent works. In particular, Raheja et al., (2019) implemented a lexical classifier consisting of a pretrained language model as an encoder, an utterance-level RNN, a context-aware self attention, and a final conversation-level RNN that achieved impressive performance at the time of the publication.

Inspired by these recent works, we implemented a lexico-acoustic dialog act classifier with two main components: a lexical model (modified from [5]) consisting of a RoBERTa encoder with self attention, and an acoustic model (based on [1]). As we

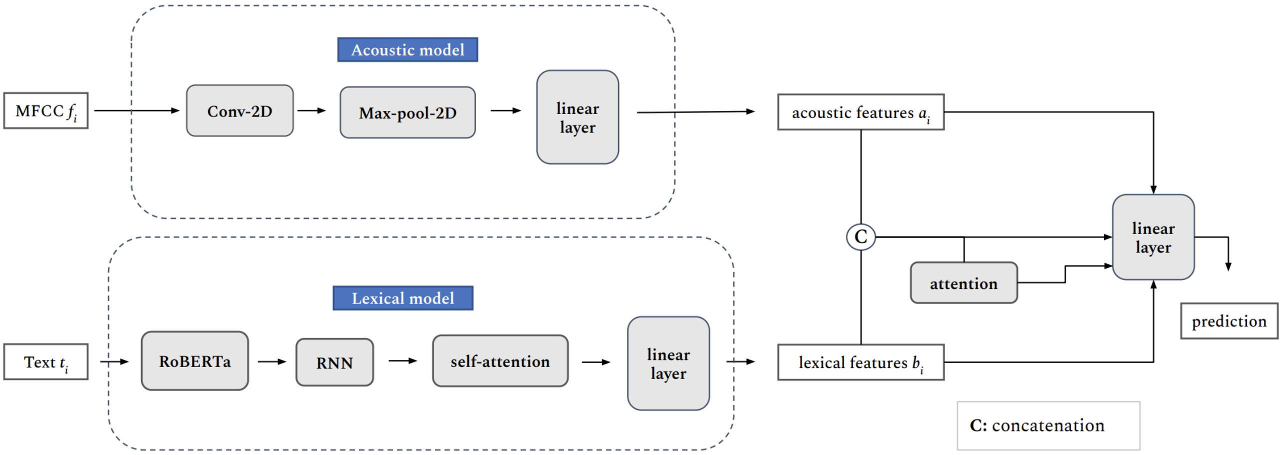


Figure 1: *Model architecture for this project. The original dataset provides audio files in wav format with their corresponding transcripts. We extracted MFCC features from these audio files and used them as our acoustic input.*

do not have the context information in our subset of the corpus, our work does not include context-aware self attention or any contextual information in either of the models. We train these models separately as well as in combination to classify DAs. To combine these modalities, i.e. the lexical and the acoustic modalities, we apply soft attention proposed by [6] to appropriately weight the contribution of each modality. We evaluate our model on a modified subset of the SwDA dataset [3] (details can be found in 3.1).

As our original dataset is highly unbalanced, we implemented two methods to mitigate this issue and run our experiments using these data rebalancing techniques as well as on the original unbalanced dataset. In 5.1, we discuss the model errors and the effects of the different attention implementations. We also run an ablation study to examine the contributions of the individual components in our lexical model. Our results show that a powerful lexical model is sufficient to learn the features that are represented by both modalities when the acoustic features are learned by using Mel-frequency-cepstral coefficients (MFCCs) through a single CNN. This suggests that the task might be efficiently solved by a single lexical model with a pretrained language model as textual encoder. We conclude that acoustic model should be explored further with different speech inputs and possibly a different model architecture.

2. Model

The model consists of two parts: a lexical model (described in 2.1) and an acoustic model (described in 2.2) that can be trained separately or combined using different techniques. Detailed descriptions are provided in the following subsections. The overall model architecture is depicted in Figure 1.

2.1. Lexical model

The lexical model is based on the structure of model proposed by [5] that consists of a pretrained RoBERTa language model (LM), an RNN and a self-attention layer. Different from [5], we remove the layer encoding context information, as our dataset does not contain this. Contextualized embeddings learned by RoBERTa provide general token meanings, and the RNN layer can capture and amplify task related information and filter out those unimportant. We also apply self-attention to enable tokens to better relate to each other, e.g. pronouns and their references. The output of the attention layer is then passed into a linear layer so to generate embeddings for each sentence. These embeddings are then passed into a softmax layer for final classification or combined with the lexical features.

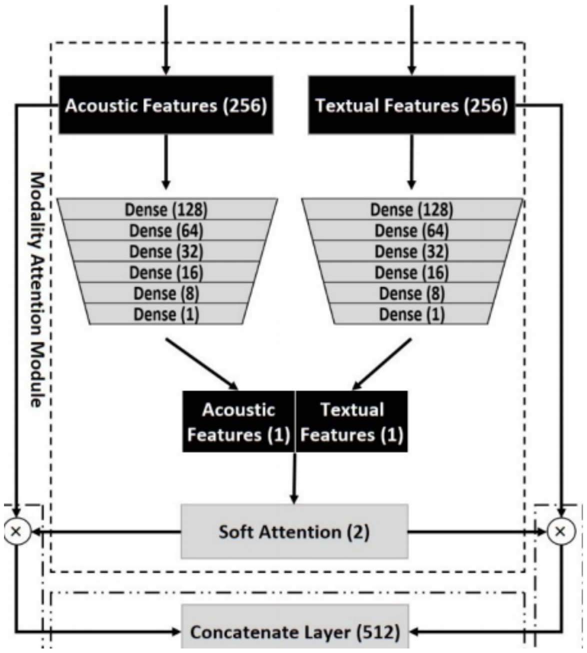


Figure 2: *Attention module from [6]. In our work, the size of the acoustic and lexical features are 128. We therefore remove the first dense layer.*

Model	balanced dataset					original dataset					balanced loss				
	ba	st	qu	op	**acc**	ba	st	qu	op	**acc**	ba	st	qu	op	**acc**
acoustic	0.91	0.79	0	0.04	0.72	0.92	0.78	0.04	0.10	0.71	0.92	0.78	0	0.04	0.72
lexical	0.96	0.85	0.57	0.52	0.81	0.98	0.85	0.65	0.60	0.82	0.98	0.84	0.57	0.47	0.81
combined	0.97	0.84	0.57	0.27	0.79	0.98	0.84	0.63	0.58	0.81	0.97	0.85	0.54	0.47	0.81

Table 2: *f_1-score of different classes and accuracy of different settings.* ***ba*** *stands for backchannel,* ***st*** *for statement,* ***qu*** *for question,* ***op*** *for opinion and* ***acc*** *for accuracy.*

class	number	proportion
backchannel	6792	0.24
opinion	4984	0.18
question	2150	0.07
statement	14459	0.51

Table 3: *Distribution of different classes*

2.2. Acoustic model

We based our implementation of the acoustic model on [1]. We use a CNN to learn the acoustic features. The acoustic model consists of a single 2D convolutional layer followed by a 2D max-pooling layer and a final fully connected layer to produce the features. Just like the lexical embeddings, the acoustic embeddings are then fed into a softmax layer for final classification or combined with the lexical features.

2.3. Lexico-acoustic model

To combine the lexical and acoustic features, we experiment with two options. Replicating the implementation from [1], in the first option, we concatenate these two learned vectors and pass them directly into a softmax layer for classification. By doing so, the combined model uses the information from both modalities equally.

In the alternative to the simple concatenated representation, we experiment with two different attention implementations to weigh the contribution of lexical and acoustic embeddings. First one is based on the work of [6], where features from different modules are passed into multiple dense layers and are compressed into 1. Then a soft attention is applied to get scores for each module. We get the final representations from different modules, weigh them through multiplication with the output of the soft attention, and then concatenate these to obtain the final representation. The structure of this approach is shown in Figure 2. This final representation is then fed into a softmax layer for classification, just like in the first option.

The second attention approach is based on [7], where we use a simple weight matrix instead of multiple dense layers to calculate the attention scores. The results shown in Table 2 are obtained using this method as our attention module.

3. Experiments

3.1. Data

We test our model on **SwDA**: NXT-format Switchboard Corpus [3], a dialog corpus of 2-speaker conversations. In our experiments, we use a 4-tag subset of the data that consists of the tags *backchannel*, *statement*, *opinion*, and *question* (all question types (bh, qwd̂, qo, br, qh, ĝ, qw, qyd̂, qy) are aggregated into a single class).

In our split of the dataset, these 4 classes are highly unbalanced, shown in Table 3. We address this in 3.2. We run our experiments both with unbalanced and balanced data using different techniques which we discuss in 4.1.

The utterances in our dataset are not associated with their contexts as the dataset has been sampled from the original corpus irrespective of this information. Also, the transcripts contain erroneously recognized words, redundant spaces, and no punctuation marks as these transcripts were obtained using an ASR system. Lastly, the utterances are pre-split into *train*, *dev* and *test* sets.

The audio files have a maximum duration of 24 and a minimum duration of 0.06 seconds. As a preprocessing step, we extract Mel-frequency-cepstral coefficients (MFCC features) using python_speech_features library [8] with default settings of 16000 sample rate, 0.025-second window and 13 cepstrum to return. We pad the MFCC features with 0s to a maximum length of 3361 to get a fixed-length input.

To obtain the lexical input, we tokenize the text with roberta-base tokenizer [9], encode them with input IDs and attention mask, and pad them to a maximum length of 128. As as result, we obtain contextual embeddings as our lexical input that reflect the information in their local context, i.e. the words in the utterance, but not the global context, i.e. the context of the conversation.

3.2. Data Rebalancing

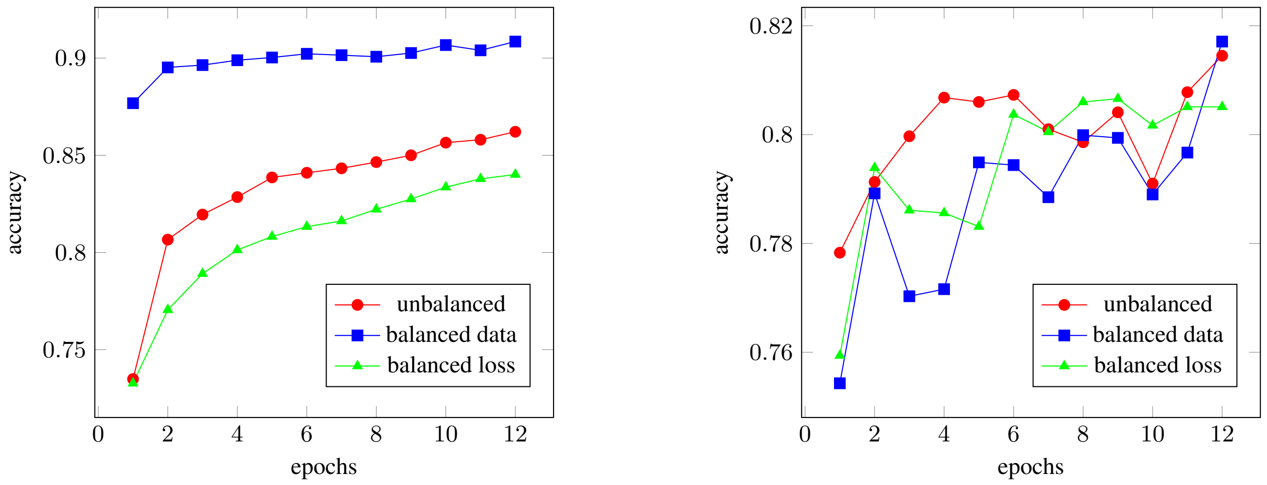
As discussed earlier, the distribution of classes over the documents in our dataset is highly unbalanced, as shown in Table 3. We implement two methods to try to mitigate the effects that might be caused by this data imbalance.

Resampling the dataset is the first and more basic approach we implemented. We upsample the minority classes in the train set, after deriving our dataset splits, by making use of sklearn[1]. This samples existing data of minority classes according to a random distribution to bolster the amount of documents in minority classes until all classes are equally distributed.

Weighted Loss also takes the distribution of classes in the dataset into account. For this, we pass this distribution of the classes to our loss function in the PyTorch[2] implementation of the model. This allows for a manual rescaling of each class's weight in the loss and helps to mitigate the effects of the unbalanced distribution of classes in the dataset.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.utils.resample.html

[2] https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html?highlight=losstorch.nn.CrossEntropyLoss



(a) ***Training Accuracy*** *between different methods of mitigating the issue of unbalanced data.*

(b) ***Validation Accuracy*** *between different methods of mitigating the issue of unbalanced data.*

Figure 3: *Performance difference between* ***Training Accuracy****(a) and* ***Validation Accuracy****(b) during training. Compared between unbalanced data, data rebalanced while employing data upsampling and rebalancing by applying weights according to the class distribution to the loss function.*

3.3. Hyperparameters and Training

We use a default learning rate of 0.0001, a maximum epoch number of 50, and a batch size of 32 for all models. During training, the logger waits 3 epochs before recording loss to allow the model to warm up. Early stopping waits for 6 epochs before terminating when the model performance does not improve.

To exam the influence of unbalanced dataset and mitigate the possible issues caused by this, we implement a data rebalancing option (explained in 3.2) that can be passed to the training as an argument. We conduct 3 independent runs (i.e. model trained on unbalanced dataset, balanced dataset and with balanced loss on unbalanced dataset) for each model and report on these runs in Table 2.

3.4. Evaluation

Quantitatively, we evaluate the performance of the models (i.e. both the lexical and the acoustic model separately as well as the combined model) based on accuracy and f_1 scores. We also carry out an error analysis to provide a qualitative evaluation.

4. Results

Table 2 displays the overall results. We train the acoustic and lexical models separately as well as in combination using different configurations. As shown in the table, our lexical model achieves the highest performance followed by the combined model both in terms of accuracy and f_1 scores for each class. Scores show that combining the lexical model with the acoustic model harms the model performance both in terms of accuracy and f_1 scores, with the only exception being the f_1 score in statement class with balanced loss. Acoustic model achieves around 72% accuracy both when the data is balanced and when the loss is balanced, and 71% accuracy when trained with unbalanced data without mitigation methods. We see that the acoustic model achieves this performance by mainly predicting *backchannel* or *statement* as the f_1 scores for *question* and *opinion* are close to 0 under all conditions. The lexical model performs the best on the original dataset achieving better f_1 scores in each class as well as higher overall accuracy with 82%.

4.1. Data Rebalancing

We describe the details of our data rebalancing implementations in 3.2. As seen in figure 3*(a)*, balancing the data by resampling it offers a significant advantage to the training accuracy compared to training on the originally unbalanced dataset. The balanced loss, on the other hand, is unable to achieve better scores on the training accuracy, performing significantly worse than both the model trained on the unbalanced data and the model trained on the resampled data.

On the validation split, the best results are achieved when the model is trained on the resampled dataset, whereas it shows only 0.3% performance drop when trained on the original unbalanced dataset. The model trained with the weighted loss function performs 1.2% worse than the best model, offering worse performance than the other two models.

Furthermore, the performance progress on the validation data appears highly unstable, with both data rebalancing approaches starting at a noticeably lower accuracy than the original model and showing unsteady progress throughout the training process. This probably stems from the implementation of the balancing method as the training dataset is the one being rebalanced, and depending on the distribution of classes between the training and validation set the weighted loss can also be skewed.

5. Analysis

5.1. Error Analysis

In this section, we carry out both quantitative and qualitative error analysis. The errors are from the default combined model with unbalanced dataset. We obtain the confusion matrix displaying the distribution of the errors in each class shown in Figure 4 using the default settings.

In general, *backchannel* is the easiest class to predict, with fewer examples predicted wrong. However, there is a huge confusion between *opinion* and *statement*, which is not a big surprise as there are no obvious features for either of these classes. Their differences are more of discourse-wise: statements are “descriptive, narrative, or personal” and opinions are other-directed [10]. In other words, opinions are often countered with further opinions while statements elicit continuers or backchannels.

To investigate this further, we look at 50 data examples that are mistakenly predicted as *opinion* (gold label *statement*) and 50 as *statement* (gold label *opinion*). We categorize the errors into five types. These are:

- **Implicit data:** Errors of this type lack detectable features. As a result, they might be too implicit for the model to predict. For instance, the following sentence is an opinion, however it is predicted as a statement:
  ——”You probably wouldn’t like my favorite team in college then.”
  If we append ”I think” at the start of the sentence, the meaning of the sentence remains the same. This tells us that the speaker is expressing something about the other person from the perspective of themselves. However, it is probably hard for the model to tell that as the sentence contains both ”you” and ”my”.

- **Lack of context**: This is the second most frequent error type. As stated before, the biggest difference between an opinion and a statement is that they elicit a different response. Since there is no contextual information associated with these utterances, it is hard to distinguish between the two types, even for humans. Consider the following example:
  ——”He’s been doing real good.”
  This example can be an opinion of the speaker and not a fact, stating what they believe to be true. In that sense, an opinion is subjective. Contrarily, if the speaker is stating a fact, then this is a statement.

- **Mixer**: For this type, there are too many possible features that would confuse the model. For instance:
  ——”But I mean I don’t think so, it is just that I am lazy.”
  The first part of the sentence has two obvious features that would lead to the prediction of opinion. However, ”it is just” from the last part somehow suggests that the sentence is a statement.

- **Abbreviation (new expressions)**: If the model is trained to find specific patterns for *opinion* and *statement*, we would expect abbreviations or less frequent expressions to fail the model as their training samples are not sufficient.

- **Wrong label**: There are some instances that are labeled wrongly, which could again confuse the model. Consider the following example, which is labeled as *statement*, but is in fact an opinion:
  ——”I don’t think you have to crush the other ones for them to get to take that.”

Our second focus is the poor performance of the model in *question* class. The questions in our dataset are mostly wrongly predicted as *statement*. Part of the reason why this happens, especially for declarative questions, might be that there are no obvious lexical features to help the model as the dataset does not contain question marks. Our intuition tells us that the acoustic features could especially be helpful in disambiguating between the statements and the wrongly classified questions as also shown by [1]. However, for us this is not the case as shown in Table 2. Looking at the f_1 scores of the *question* class from the acoustic model, we see that the acoustic features that are learned by the model are not helpful in predicting questions.

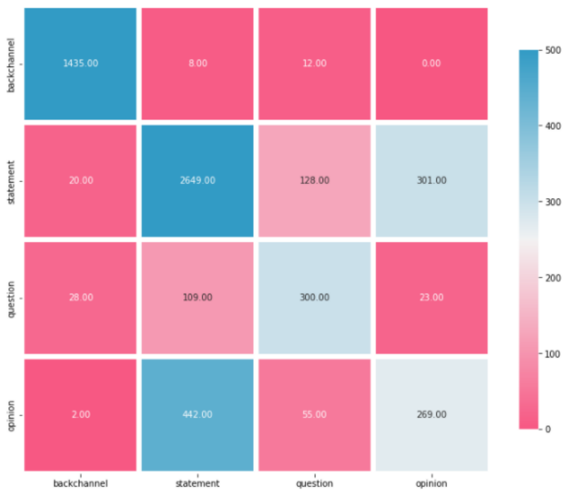


Figure 4: *Confusion matrix of the combined model.*

The poor performance of the acoustic model for the *question* class can either come from the data or the model. Regarding the data, one hypothesis could be that the questions do not have enough correlating features for the model to gain confidence as the dataset contains by far the least amount of examples in this class, also shown in Table 3. As for the model, it might be that 1) the MFCC features do not capture prosody information, which is very likely, as MFCCs are extracted from very short frames while prosody features are usually longer, 2) MFCC features contain prosodic features, however, these features are lost during convolution and other operations.

To take a closer look into this, we randomly sampled 20 data points from the question class. After listening the audios, we qualitatively found that most of them indeed have distinguished question features. To find out if MFCC features are responsible for the poor model performance of the *question* class, we carry out a statistic analysis to see if the MFCCs of *question* are significantly different from those of other classes. To do this, we only use the entire feature set and categorize these into two, i.e. question and others. We use Mann–Whitney U test [11] to see whether the distribution is different. The statistics and p values are 224777645299667.0 and 3.6771353526458307e-23 respectively. This means that MFCC features from *question* do differ from those of other classes. This might suggest that prosody information is actually captured in MFCC features, and it might be lost during the model training. However, this is only a speculation as these statistical correlations might not be related to the prosody information.

5.2. Ablation Study of Lexical Model

The lexical model is a stack of a pretrained RoBERTa LM, an RNN layer and a self attention layer. To investigate which part of the model contributes most/least to the final performance, we perform an ablation study. Table 4 shows the results of this study.

model	ba	st	qu	op	acc
RoBERTa+RNN+atten	0.98	0.84	0.64	0.62	0.82
RoBERTa+RNN	0.97	0.84	0.62	0.62	0.82
RoBERTa-only	0.97	0.83	0.60	0.61	0.81

Table 4: *Ablation study on the lexical model*

Looking at the table, we see that except for *question* the scores for the classes do not differ much across different models. It seems that the pretrained RoBERTa LM contributes the most to the performance. Neither the RNN layer nor the self attention contributes significantly to the performance. However, we observe that the full lexical model achieves the highest f_1 score in each class compared to RoBERTa-only. This might suggest that using an RNN and a self-attention layer help the model better encode information compared to RoBERTa-only. We see this contribution especially in *question* class. One postulation is that the question information is ”enhanced” when an RNN and a self-attention layer are used as we are passing the question information to other related tokens in the sentence as well. As a result of this, the information might be retained better.

5.3. Attention in Combined Model

As mentioned previously, we experimented with two different attention implementations. In the first approach, shown in Figure 2, acoustic and lexical features are fed into multiple linear layers respectively in order to get the hidden representations for calculating the attention weights. The second attention implementation is just a simple linear layer that takes a concatenation of acoustic and lexical features and outputs a tensor of size 2. We find that the overall accuracy for the combined model does not differ between these two attention approaches.

6. Discussion and Future Work

In this work, we experimented with self-attention in lexical models and proposed different ways to combine acoustic and lexical modalities in DA classification. Our experiments show that a powerful lexical model performs better when trained alone than when combined with an acoustic model. Ablation study on the lexical model shows that, although not so much in terms of performance, a RNN followed by a self-attention might help stabilize the training when stacked on top of a pretrained language model.

Deeper analysis on the comparison between the two mitigation methods, i.e. the data rebalancing and the weighted loss, revealed that resampling from the dataset according to the data distribution offers a significantly better advantage both to the training and to the validation accuracy. Weighted loss, on the other hand, not only works worse than the resampling approach but also harms the training as the model performs better on the original unbalanced data.

Unlike our original intuition, in our case, the acoustic model did not help disambiguate between the classes in which one of the greatest confusion occurs, i.e. between *question* and *statement*. We believe that this might be due to the MFCC features not being able to capture enough prosody information for the model to learn from. We also suspect that this is amplified by the lack of sufficient training instances in this class as the dataset contains the least number of examples labeled as *question*. One possible future direction here could be the extraction of prosodic features, such as F0, and training the acoustic model on these features instead of the MFCCs or in combination as prosody is believed to be helpful in cases like ours. We also suggest training a deeper and more powerful CNN, possibly with residual connections, to capture more prominent correlations in the speech data.

Furthermore, we conclude that it is rather hard to disambiguate between a statement and an opinion when the model is not provided with the conversational context as the cues for these classes can be misleading or sometimes not even present. As mentioned before, the biggest difference between a statement and an opinion lies in what they elicit in the following utterances. Here, lack of context can mean lacking significant information to detect the intent of the speaker with a given utterance. To know whether an utterance is a statement or an opinion, one must either be aware of the context leading up to that utterance or the following utterance. Although we did not investigate training with context in this work, this still remains an interesting aspect to study, especially in regard to the confusion between the *statement* and the *opinion* classes, but also to the *question* overall. We leave this comparison study to future work.

In this work, we did not find that attention among different modules could help boost the overall performance, which is quite out of our expectations. Looking closer into the attention weights, we found that the weights for the lexical model are much higher than the weights for the acoustic model. This again confirms that the combined model mostly uses the lexical information in its predictions, and discards most of the acoustic information. We believe that this behavior is mainly attributed to the poor performance of the acoustic model.

Lastly, as mentioned in 3.1, the audio transcripts in our dataset were rather erroneous and did not contain punctuation marks. As we showed how powerful just a pretrained language model can be in classifying DAs, we believe that it would be worth investigating how the performance of the model changes when trained on non-erroneous data (or at least on data that contains less systematic errors) and possibly with punctuation marks as we are obtaining better and better transcripts with the advancements in ASR technology.

7. References

[1] D. Ortega and N. T. Vu, “Lexico-acoustic neural-based models for dialog act classification,” 2018.

[2] N. Kalchbrenner and P. Blunsom, “Recurrent convolutional neural networks for discourse compositionality,” *Workshop on CVSC*, 06 2013.

[3] S. Calhoun, J. Carletta, J. M. Brenier, N. Mayo, D. Jurafsky, M. Steedman, and D. Beaver, “The nxt-format switchboard corpus: a rich resource for investigating the syntax, semantics, pragmatics and prosody of dialogue,” *Language Resources and Evaluation*, vol. 44, pp. 387–419, 2010.

[4] V.-T. Dang, T. Zhao, S. Ueno, H. Inaguma, and T. Kawahara, “End-to-end speech-to-dialog-act recognition,” 2020.

[5] V. Raheja and J. R. Tetreault, “Dialogue act classification with context-aware self-attention,” in *NAACL*, 2019.

[6] Y. Gu, K. Yang, S. Fu, S. Chen, X. Li, and I. Marsic, “Hybrid attention based multimodal network for spoken language classification,” in *Proceedings of the 27th International Conference on Computational Linguistics*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 2379–2390. [Online]. Available: https://aclanthology.org/C18-1201

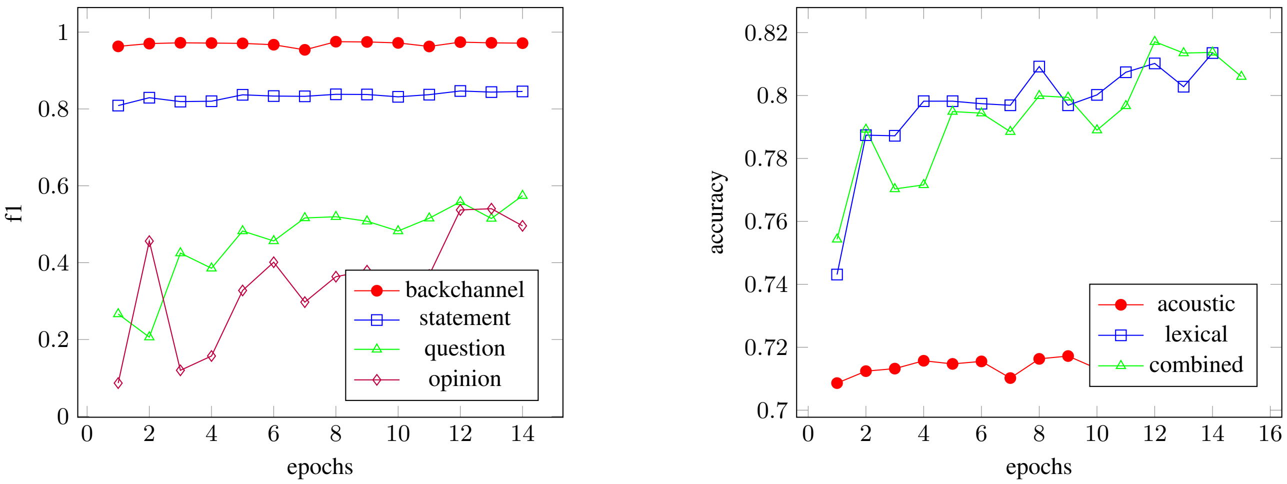
[7] L. Weng, “Attention? attention!” *lilianweng.github.io/lil-log*, 2018. [Online]. Available: http://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html

[8] J. Lyons, D. Y.-B. Wang, Gianluca, H. Shteingart, E. Mavrinac, Y. Gaurkar, W. Watcharawisetkul, S. Birch, L. Zhihe, J. Hölzl, J. Lesinskis, H. Almér, C. Lord, and A. Stark, “jameslyons/python_speech_features: release v0.6.1,” Jan. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3607820

[9] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” 2019.

[10] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. V. Ess-Dykema, and M. Meteer, “Dialogue act modeling for automatic tagging and recognition of conversational speech,” *Computational Linguistics*, vol. 26, pp. 339–373, 2000.

[11] M. Neuhäuser, *Wilcoxon–Mann–Whitney Test*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1656–1658. [Online]. Available: https://doi.org/10.1007/978-3-642-04898-2_615



(a) f_1 score over epochs between targets

(b) Accuracy over epochs between models for balanced data.