Report: Dialog Act Classification
using Word Embeddings & Acoustic Features

Jens Beck
jens.beckl@ims

Fabian Fey
fabian.fey@ims

Richard Kollotzek
richard.kollotzek@ims

Abstract

Nam dui ligula, fringilla a, euismod soda-les, sollicitudin vel, wisi. Morbi auctor lo-rem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vesti-bulum turpis. Pellentesque cursus luctus mauris.

1 Introduction

The general task is to classify lexical and auditory speech into one of four predefined *dialog act clas-ses*. A *dialog act*, in this context, represents in-formal information of how a dialog system should respond to a users input. The four provided classes are *statement, opinion, question* and *backchannel*. To solve this task we developed *convolutional neu-ral networks* (CNN) that use lexical and acoustic features. For the development and training of the systems a subset of the *Switchboard Dialog Act Corpus* was used. In next chapters we discuss the development of the systems and subsequently to that the research question **INSERT HERE**.

2 Data & Data Preperation

In this section we discuss the *Switchboard Dialog Act Corpus* and the extraction of the lexical and acoustic features.

2.1 The Switchboard Dialog Act Corpus

The *Switchboard Dialog Act Corpus* [2], from now on abbreviated as *SwDA*, consists of recordings

	training	dev	test
opinion (∼17%)	4984	1068	1070
question (∼8%)	2150	460	463
backchannel (∼24%)	6792	1455	1458
statement (∼51%)	14459	3098	3099
sum	28385	6081	6090

Table 1: Displays the distribution of the four classes in the three data sets.

with corresponding transcripts. Each of these re-cordings is assigned to one of 42 *dialog act clas-ses*. For this project we reduced the amount of classes down to four which are *statement, opinion, question* and *backchannel*. These classes are su-persets of the 42 *dialog act classes* defined in the *SwDA*. The distribution of the four classes within the training, development and test set are shown in Table 1. The numbers illustrate a huge imba-lance between the *statement* class and the other three classes. However, we decided against redu-cing the data into equally distributed sets because this would exclude at least one third of the trai-ning data. This is important to keep in mind for the evaluation of the systems because an educated guess would have an accuracy of around 51% by assigning all test examples to the *statement* class.

2.2 Input Data Generation

Lexical and acoustic features were employed in our systems and had to be extracted and format-ted into a machine readable format. For the lex-ical features we decided to use *Google's* freely accessible word embeddings which were trained on 100 billion words [4]. As for the acoustic fea-tures we relied on *Mel Frequency Cepstral Coeffi-cient* (MFCC) features which were extracted with the *openSMILE* feature extraction tool [1]. To en-sure that data of different utterances could not be mixed the lexical and acoustic inputs were always stored with their respective one-hot vector in a tri-

ple data structure.

Lexical Features

The word embedding matrix E was generated by assigning each word to its corresponding 300 dimensional vector of the *Google word2vec* model. If a word was not included in the model it was assigned a randomly generated 300 dimensional vector. Furthermore, we introduced a padding vector for the case that a sentence was shorter than our maximum sentence length. We decided to restrict the length of a single utterance to 100 words to not exclude to much lexical features for long utterances. The final size of the embedding matrix E was 11825×300.

$$E = \begin{bmatrix} v_{1,1} & \cdots & v_{1,300} \\ v_{2,1} & \cdots & v_{2,300} \\ \vdots & \ddots & \vdots \\ v_{i,1} & \cdots & v_{i,300} \end{bmatrix}$$

The lexical input for our systems is a vector x_{lex} were each word is represented by the index i of its corresponding vector in the embedding matrix E. The vector x_{lex} has a length of 100. Each element represents the index of a word in the utterance. The vector x_{lex} has the following shape: $x_{lex} = [i_1, i_2, ..., i_{100}]$ where i represents the index.

Acoustic Features

Each utterance had its acoustic features formatted into a matrix X_{aco} of shape 13×2000. This matrix was generated by arranging 2000 *MFCC-frames* into a matrix. Each *frame* hereby consisted of 13 coefficients. The chosen *frames* were the first and last thousand *frames* of the audio recording. If a recording had less than 2000 *MFCC-frames* the matrix was padded with zero vectors.

$$X_{aco} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,2000} \\ a_{2,1} & \cdots & a_{2,2000} \\ \vdots & \ddots & \vdots \\ a_{13,1} & \cdots & a_{13,2000} \end{bmatrix}$$

Afterwards, to use minibatch processing it was necessary to reformat X_{aco} into a vector x_{aco} with the shape 1×26000.

3 Baseline Systems

The architecture of the proposed *AcoLex* system is depicted in Figure 1. In this section we will explain the complete architecture of the system.

Furthermore, we will discuss its two core components: the lexical and the acoustic model.
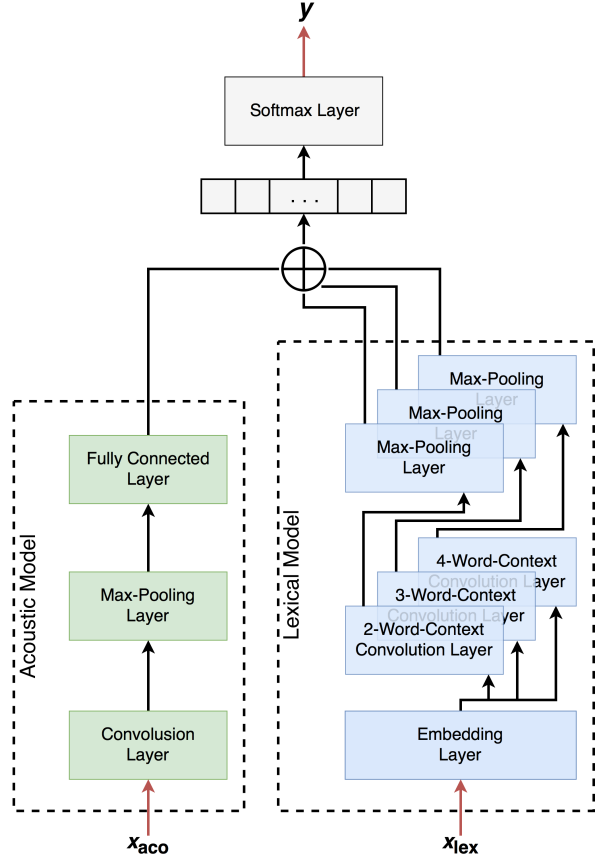

Figure 1: bla bla bla

3.1 Lexical Model

The *lexical model* (*LM*), depicted on the right side in Figure 1, consists of three layers. The architecture is based on the system introduced by Kim (2014) [3]. The first layer is an *embedding layer* which was initialized with the *embedding matrix* E described in Section 2.2. The benefits of using an embedding layer is a significant reduction of the time needed to import the input files due to storing the vector of each word only once. Furthermore it enables the tuning of the *embedding vectors* for the classification task. The second layer is a *convolution layer* which uses three different filter sizes to capture different word contexts, namely *2-word-*, *3-word-* and *4-word-contexts*. Overall 300 filters are applied in the LM, were each filter type is used 100 times. The last layer is a *max-pooling layer* that returns the highest value of each filter output. Therefore, the final output of the LM are three 100 element long vectors.

3.2 Acoustic Model

The *acoustic model* (*AM*), shown on the left side in Figure 1, is developed similarly to the acoustic model presented by Ortega et. al (2018) [5]. Like the *LM* it utilizes three different layers. The first layer is a *convolution layer* with one filter type that captures 100 *MFCC-frames* at a time. We decided to use a stride of 50 to significantly reduce the processing time. The second layer is a *max-pooling layer* similar to the one in the *LM*. This layer returns a 100 element long vector. This output is fed into a *fully connected layer* which returns the final output of the *AM* which is a 100 element long vector.

3.3 AcoLex Model

Our proposed *AcoLex Model* employs a *bi-CNN* that consists of the *LM* and the *AM* to simultaneously process lexical and acoustic features. The outputs are then concatenated into one 400 element long vector which is then passed into a *softmax layer* to perform the *dialog act classification task*. The final output is a four element long vector were each element represents the probability for one *dialog act class*.

4 Training

The system was trained for 21 epochs with the *stochastic gradient descent* optimization algorithm over mini-batches of size 100. The learning rate was fixed to 0.01. The tunable hyperparameters are shown in Table 2.

Hyperparameter	LM	AM
Filter width	2, 3, 4	100
Activation function	Relu	TanH
Word embeddings	word2vec	–
MFCC features	–	13
Dropout*	0.5	0.5
Learning Rate*	0.01	0.01
Mini-batch size*	100	100
# Feature maps*	100	100

Table 2: Hyperparameters marked with * are model independent and were set for the whole system.

Model	Dev Accuracy	Test Accuracy
Acoustic		
Lexical	77.52	77.40
AcoLex	80.24	79.93

Table 3: Hyperparameters marked with * are model independent and were set for the whole system.

5 Results

6 Research Question: None

7 Conclusion

References

[1] F. Eyben, M. Wöllmer, and B. Schuller. Opensmile: The munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 1459–1462, New York, NY, USA, 2010. ACM.

[2] J. J. Godfrey, E. C. Holliman, and J. McDaniel. Switchboard: Telephone speech corpus for research and development. In *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing - Volume 1*, ICASSP'92, pages 517–520, Washington, DC, USA, 1992. IEEE Computer Society.

[3] Y. Kim. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882, 2014.

[4] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

[5] D. Ortega and N. Thang Vu. Lexico-acoustic Neural-based Models for Dialog Act Classification. *ArXiv e-prints*, Mar. 2018.