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 sodales, sollicitudin vel, wisi. Morbi auctor lorem 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 vestibulum turpis. Pellentesque cursus luctus mauris.

1 Introduction

The general task is to classify lexical and auditory speech into one of four predefined *dialog act classes*. A *dialog act*, in this context, represents informal 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 neural 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 with corresponding transcripts. Each of these recordings is assigned to one of 42 *dialog act classes*. For this project we reduced the amount of classes down to four which are *statement, opinion, question* and *backchannel*. These classes are supersets 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 imbalance between the *statement* class and the other three classes. However, we decided against reducing the data into equally distributed sets because this would exclude at least one third of the training 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.

	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.

2.2 Input Data Generation

Lexical and acoustic features were employed in our systems and had to be extracted and formatted into a machine readable format. For the lexical features we decided to use *Google's* freely accessible word embeddings which were trained on 100 billion words [4]. As for the acoustic features we relied on *Mel Frequency Cepstral Coefficient* (MFCC) features which were extracted with the *openSMILE* feature extraction tool [1]. To ensure 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 triple 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.

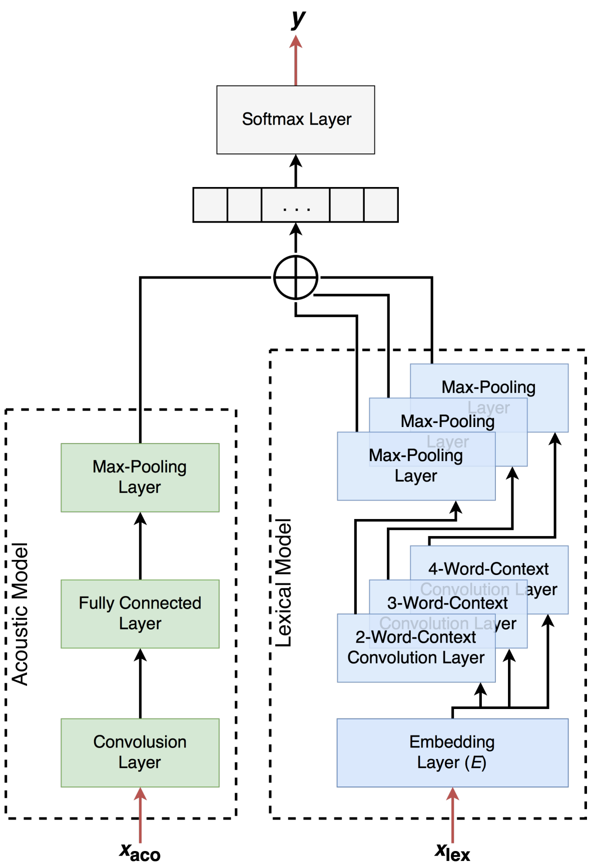

Figure 1: bla bla bla

3.1 Lexical Model

The lexical model (LM), depicted on the right side in Figure 1, consists of three layers. The first layer is an *embedding layer E* followed by a *convolution layer* which uses three different filter sizes and finally a *max-pooling layer*. The *embedding layer* yields the same *embedding matrix* as explained in Section 2.2. The filters of the *convolution layer* capture three 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. After the convolution the outputs are passed to a *max-pooling layer* which 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

3.3 AcoLex Model

4 Results

5 Research Question: None

6 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.