Dialog Act Classification

using Word Embeddings & Acoustic Features

Jens Beck, Fabian Fey, Richard Kollotzek
Institute for Natural Language Processing, University of Stuttgart

Task Introduction

- Dialog Act Classification describes the process of automatically predicting the dialog act of the current speech and textual information
- We present an approach using a Convolutional Neural Network (CNN) which classifies utterances in four classes:
  - statement ("I think I read about that in the paper")
  - question ("Well where do you take those things")
  - opinion ("It was really good")
  - backchannel ("Uh-huh")
- Two kinds of features:
  - Lexical features
  - Acoustic features

Data

Switchboard

- We use a subset of the Switchboard Telephone Speech Corpus which consists of lexical and acoustic data
- Our subset includes 40,556 sentences
- The lexical dataset is divided in training, development and test data
- The acoustic dataset includes a recording for every utterance

Dataset\Channel	opinion	question	backchannel	statement	Sum
training	4984	2150	6792	14459	28385
development	1068	460	1455	3098	6081
test	1070	463	1458	3099	6090

MFCC features

- With OpenSmile we extract the MFCC features for every sentence
- The MFCC features are extracted every 10ms with a frame size of 25ms
- This results in 13 features for each measurement point

word2vec

- For the word embedding layer we use the pre-trained Google word2vector model
- Contains 3 million words representing one word as a 300-dimensional vector

Data Preprocessing

Embedding Matrix

1. All words from the training set are inserted into an embedding matrix
2. Each word is represented by it's corresponding vector from *word2vec*
3. If a word is not contained in *word2vec* it gets assigned a random vector
4. Unkown word and no word vectors added

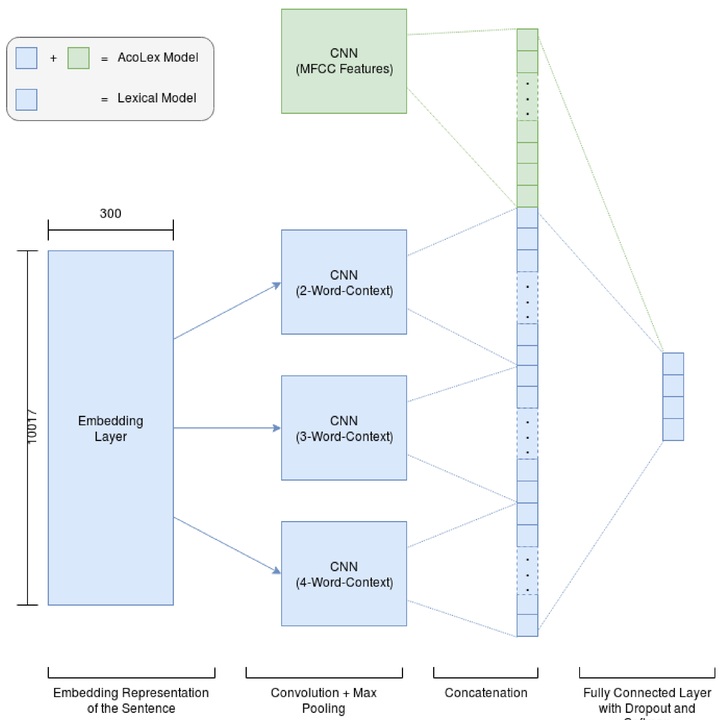
Lexical Features

1. Each sentence is converted into a sequence of indexes
2. Each index is the corresponding line in the embedding matrix for one word
3. The maximum sentence length is set to 100 words

Acoustic Features

1. The MFCC features of the first 10 second are used
2. The MFCC feature of the last 10 second are used
3. If audio file is shorter than 20 seconds the missing MFCC features are zeroized

System Architecture



Intermediate Results

	Epochs	Accuracy (%)	Trainable Embeddings	Learning Rate	Activation func.	Loss func.
Lexical	15	78.42	True	0.05	TanH	Hinge-Loss
	15	70.89	False	0.05	TanH	Hinge-Loss
	15		True	0.05	Relu	Hinge-Loss
	15	60.50	False	0.05	Relu	Hinge-Loss
	15		True	0.05	Sigmoid	Hinge-Loss
	15		False	0.05	Sigmoid	Hinge-Loss
	15	78.66	True	0.05	TanH	Cross Entropy
	15	69.62	False	0.05	TanH	Cross Entropy
	15		True	0.05	Relu	Cross Entropy
	15		False	0.05	Relu	Cross Entropy
	15		True	0.05	Sigmoid	Cross Entropy
	15		False	0.05	Sigmoid	Cross Entropy
Acolex	15	69.82	False	0.01	Sigmoid	Hinge-Loss
	15	77.90	True	0.01	TanH	Hinge-Loss

Potential Future Work

- What we plan next:
  - Varying MFCC feature amount
  - Using smoothed training data to better balance the classes
  - Using stop word filtering
  - Including words of the test and development set into the embedding layer
  - Insertion of an additional fully connected layer between the CNN output and the softmax layer

References

[1] Florian Eyben, Martin Wöllmer, and Björn 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] Yoon Kim. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882, 2014.

[3] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.