ASR performance prediction on unseen broadcast programs using convolutional neurol networks

Zied Elloumi, Laurent Besacier, Olivier Galibert, Juliette Kahn, Benjamin Lecouteux

ASR PERFORMANCE PREDICTION ON UNSEEN BROADCAST PROGRAMS USING CONVOLUTIONAL NEURAL NETWORKS

Zied Elloumi [1] [2], *Laurent Besacier*[2], *Olivier Galibert* [1], *Juliette Kahn*[1], *Benjamin Lecouteux*[2]

[1] Laboratoire national de métrologie et d'essais (LNE) , France
[2] Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France
zied.elloumi@lne.fr

ABSTRACT

In this paper, we address a relatively new task: prediction of ASR performance on unseen broadcast programs. We first propose an heterogenous French corpus dedicated to this task. Two prediction approaches are compared: a state-of-the-art performance prediction based on regression (engineered features) and a new strategy based on convolutional neural networks (learnt features). We particularly focus on the combination of both textual (ASR transcription) and signal inputs. While the joint use of textual and signal features did not work for the regression baseline, the combination of inputs for CNNs leads to the best WER prediction performance. We also show that our CNN prediction remarkably predicts the WER distribution on a collection of speech recordings.

Index Terms— Performance Prediction, Large Vocabulary Continuous Speech Recognition, Convolutional Neural Networks.

1. INTRODUCTION

Predicting automatic speech recognition (ASR) performance on unseen speech recordings is an important Grail of speech research. From a research point of view, such a task helps understanding automatic (but also human) transcription performance variation and its conditioning factors. From a technical point of view, predicting the ASR difficulty is useful in applicative workflows where transcription systems have to be quickly built (or adapted) to new document types (predicting learning curves, estimating the amount of adaptation data needed to reach an acceptable performance, etc.).

ASR performance prediction from unseen documents differs from confidence estimation (CE). While CE systems allow detecting correct parts as well as errors in an ASR output, they are generally trained for a particular system and for known document types. On the other hand, performance prediction focuses on unseen document types and the diagnostic may be provided at broader granularity, at document level for instance[1]. Moreover, in performance prediction, we may not have access to the ASR system investigated (no lattices nor N-best hypotheses, no internals of the ASR decoding) which will be considered as a black-box in this study.

[1]Nevertheless this paper will analyze performance prediction at different granularities, from fine to broad grain: utterance, collection.

Contribution This paper proposes to investigate a new task: prediction of ASR performance on unseen broadcast programs. Our first contribution is methodological: we gather a large and heterogenous French corpus (containing *non spontaneous* and *spontaneous* speech) dedicated to this task and propose an evaluation protocol. Our second contribution is an objective comparison between a state-of-the-art performance prediction based on regression (engineered features) and a new strategy based on convolutional neural networks (learnt features). Several approaches to encode the speech signal are investigated and it is shown that both (textual) transcription and signal encoded in a CNN lead to the best performance.

Outline The paper is organized as follows. Section 2 is a brief overview of related works. Section 3 details our evaluation protocol (methodology, dataset, metrics). Section 4 presents our ASR performance prediction methods and 5 the experimental results. Finally section 6 concludes this work.

2. RELATED WORKS

Several works tried to propose effective confidence measures to detect errors in ASR outputs. Confidence measures were introduced for OOV detection by [1] and extended by [2] who used word posterior probability (WPP) as confidence measure for ASR. While most approaches for confidence measure estimation use side-information extracted from the recognizer [3], methods that do not depend on the knowledge of the ASR system internals were also introduced [4].

As far as the WER prediction is concerned, [5] proposed an open-source tool named *TranscRater* based on feature extraction (lexical, syntactic, signal and language model features) and regression. Evaluation was performed on CHiME-3 data and interestingly it was shown that *signal* features did not help the WER prediction.

One contribution of our paper is to encode signal information in a CNN for WER prediction. Encoding signal in a CNN has been done in several speech processing front-ends [6, 7, 8]. Some recent works directly used the raw signal for speech recognition [7, 9] or for sound classification [10].

3. EVALUATION FRAMEWORK

We focus on ASR performance prediction on unseen speech data. Our hypothesis is that performance prediction systems

should only use the ASR transcripts (and the signal) as input in order to predict the corresponding transcription quality. Obviously, reference (human) transcriptions are only available during training of the prediction system. A $Train_{pred}$ corpus contains many pairs {*ASR output, Performance*} (more than 75k ASR turns in this work), a $Test_{pred}$ corpus only contains ASR outputs (more than 6.8k turns in this work) and we try to predict the associated transcription performance. Reference (human) transcriptions on $Test_{pred}$ are used to evaluate the quality of the prediction.

3.1. French Broadcast Programs Corpus

The data used in our protocol comes from different broadcast collections in French:

- Subset of *Quaero*[2] data which contains 41h of broadcast speech from different French radio and television programs on various subjects.
- Data from *ETAPE* [11] project which includes 37h of radio and television programs (mainly spontaneous speech with overlapping speakers).
- Data from *ESTER 1 & ESTER 2* [12] containing 111h of transcribed audio, mainly from French and African radio programs (mix of prepared and more spontaneous speech: anchor speech, interviews, reports).
- Data from *REPERE* [13]: 54 hours of transcribed shows (spontaneous, such as debates) and TV news.

As described in Table 1, the full data contains non spontaneous speech (NS) and spontaneous speech (S). The data used to train our ASR system ($Train_{Acoustic}$) is selected from the non-spontaneous speech style that corresponds mainly to broadcast news. The data used for performance prediction ($Train_{pred}$ and $Test_{pred}$) is a mix of both speech styles (S and NS). It is important to mention that shows in $Test_{Pred}$ data set were unseen in the $Train_{Pred}$ and vice versa. Moreover, more challenging (high WERs) shows were selected for $Test_{Pred}$.

	$\text{Train}_{Acoustic}$	Train_{Pred}	Test_{Pred}
NS	100h51	30h27	04h17
S	-	59h25	04h42
Duration	100h51	89h52	08h59
WER	-	**22.29**	**31.20**

Table 1. Distribution of our data set between non spontaneous (NS) and spontaneous (S) styles

Our shows with spontaneous speech logically have a higher WER (from 28.74% to 45.15% according to the program) compared to the shows with non-spontaneous speech (from 12.21% to 25.41% according to the broadcast program)[3]. This S/NS division will allow us to compare our performance prediction systems on different types of documents with *non spontaneous* and *spontaneous* speech.

3.2. ASR system used

To obtain speech transcripts (ASR outputs) for the prediction model, we built our own French ASR system based on the KALDI toolkit [14] (following a standard Kaldi recipe). A hybrid HMM-DNN system was trained using $\text{Train}_{Acoustic}$ (100 hours of broadcast news from ESTER, REPERE, ETAPE and Quaero). A 5-gram language model was trained from several French corpora (3323M words in total - from EUbookshop, TED2013, Wit3, GlobalVoices, Gigaword, Europarl-v7, MultiUN, OpenSubtitles2016, DGT, News Commentary, News WMT, LeMonde, Trames, Wikipedia and transcriptions of our $Train_{Acoustic}$ dataset) using SRILM toolkit [15]. For the pronunciation model, we used lexical resource BDLEX [16] as well as automatic grapheme-to-phoneme (G2P)[4] transcription to find pronunciation variants of our vocabulary (limited to 80k).

3.3. Evaluation

The LNE-Tools [17] are used to evaluate the ASR performance. Overlapped speech and empty utterances are removed. We obtain 22.29% WER on Train_{pred} and 31.20% on Test_{pred} (see Table 1). In order to evaluate *WER* prediction task, we use *Mean Absolute Error* (*MAE*) metric defined as:

$$MAE = \frac{\sum_{i=1}^{N} |WER^{i}_{Ref} - WER^{i}_{Pred}|}{N} \quad (1)$$

where *N* is the number of units (utterances or files).

We also use Kendall's rank correlation coefficient τ between real and predicted WERs, at the utterance level.

4. ASR PERFORMANCE PREDICTION

4.1. Regression Baseline

An open-source tool for automatic speech recognition quality estimation, *TranscRater* [5], is used for the baseline regression approach. It requires *engineered features* to predict the WER performance. These features are extracted for each utterance and are of several types: *Part-of-speech (POS)* features capture the plausibility of the transcription from a syntactic point of view[5]; *Language model (LM)* features capture the plausibility of the transcription according to a N-gram model (fluency)[6]; *Lexicon-based (LEX)* features are extracted from the ASR lexicon[7]; *Signal (SIG)* features capture the difficulty of transcribing the input signal (general recording conditions, speaker-specific accents)[8].

[2]http://www.quaero.org

[3]Detailed results per broadcast programs not shown here due to space constraints

[4]`http://lia.univ-avignon.fr/chercheurs/bechet/download/lia_phon.v1.2.jul06.tar.gz`

[5]*Treetagger* [18] is used for POS extraction in this study

[6]We train a 5-gram LM on 3323M words text already mentioned

[7]A feature vector containing the frequency of phoneme categories in its prononciation is defined for each input word

[8]For feature extraction, *TranscRater* computes 13 MFCC (using *Opensmile*[19]), their delta, acceleration and log-energy, F0, voicing probability,

This approach, based on *engineered features*, can be considered as our baseline. One drawback is that its application to new languages requires to find adequate resources, dictionaries and tools which makes the prediction method less flexible. The next sub-section proposes a new (resource-free) prediction approach based on convolutional neural networks (CNNs) where features are learnt during training.

4.2. Convolutional Neural Networks (CNNs)

For WER prediction, we built our model using both *Keras* [20] and *Tensorflow*[9].

For pure textual input, we propose an architecture inspired from [21] (green in Figure 1). Input is an utterance padded to *N* words (N is set as the length of the longest sentence in our full corpus) presented as a matrix *EMBED* of size NxM (M is embedding size - our embeddings are obtained with Word2Vec [22]). The convolution operation involves a filter w which is applied to a segment of h words to produce a new feature. For example, feature c_i is generated from the words $x_{i:i+h-1}$ as:

$$c_i = f(w.x_{i:i+h-1} + b) \quad (2)$$

Where *b* is a bias term and *f* is a non-linear function. This filter is applied to each word segment in the utterance to produce a *feature map* $c = [c_1, c_2 \dots c_{n-h+1}]$. *Max-pooling* [23] then takes the 4 largest values of c, which are then averaged. W filters provide a W-sized input to two fully-connected hidden layers (256 and 128) followed respectively by dropout regularization (0.2 and 0.6) before WER prediction.

For signal input, we use the best architecture (*m18*) proposed in [10] (colored in red in Figure 1). This is a deep CNN with 17 conv+max-pooling layers followed by global average pooling and three hidden layers (512, 256 and 128 dimensions). A dropout regularization of 0.2 is added between the last two layers (256 and 128). We investigate several inputs to the CNN using *Librosa* [24]: raw signal, mel-spectrogram or MFCCs.

In order to predict the WER using CNN, we propose two different approaches:

- **CNN**$_{Softmax}$: we use $Softmax$ probabilities and an external fixed WER$_{Vector}$ to compute WER$_{Pred}$. WER$_{Vector}$ and *Softmax* output must have the same dimension. WER$_{Pred}$ is then defined as (expectation):

$$WER_{Pred} = \sum_{C=1}^{NC} P_{Softmax}(C) * WER_{Vector}(C) \quad (3)$$

  With *NC* as total number of classes. In our experiment, we use 6 classes with WER$_{Vector}$=[0%, 25%, 50%, 75%, 100%, 150%],

- **CNN**$_{ReLU}$: after the last FC layer, a *ReLU* function returning a float value between 0 and $+\infty$ estimates directly WER$_{Pred}$.

loudness contours and pitch for each frame. The SIG feature vector for the entire input signal is obtained by averaging the values of each frame

[9]https://www.tensorflow.org

For joint use of both speech and text, we merge the last hidden layers of both CNN *EMBED* and CNN *RAW-SIG (or MEL-SPEC or MFCC)* by concatenating and passing them through a new hidden layer before CNN$_{Softmax}$ or CNN$_{ReLU}$ (see dotted lines in the Figure 1) and we train the full network similarly.

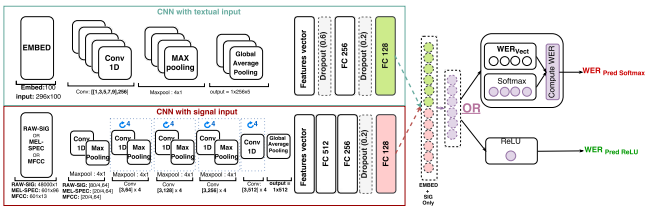


Fig. 1. Architecture of our CNN with text (green) and signal (red) inputs - dotted lines correspond to joint text+signal case

5. EXPERIMENTS AND RESULTS

In this section, *Regression* and *CNN* approaches are compared for ASR performance prediction. The *Regression* uses several engineered features extracted from the ASR output (POS, LEX, LM, SIG) while the *CNN* is based on features learnt from the ASR output and from the signal only. For the CNN, we randomly select 10% of the Train$_{Pred}$ data as a *Dev* set. 10 different model trainings (cross-validation) with 50 epochs are performed. Training is done with the $Adadelta$ update rule [25] over shuffled mini-batches. We use MAE as both loss function and evaluation metric. After training, we take the model (among 10) that led to the best *MAE* on *Dev* set and report its performance on Test$_{Pred}$. We investigate several inputs to the CNN:

- Textual (ASR transcripts) only (**EMBED**): input matrix has dimension 296x100 (296 is length of longest ASR hypothesis in our corpus ; 100 is dimension of word embeddings pre-trained on our large text corpus of 3.3G words)[10],

- Raw signal only (**RAW-SIG**): models are trained on six-second speech turns and sampled at 8khz (to avoid memory issues). Short speech turns ($< 6s$) are padded with zeros. Our input has dimension 48000 x 1[11],

- Spectrogram only (**MEL-SPEC**): we use same configuration as for raw signal ; we have 96-dimensional vectors (each dimension corresponds to a particular mel-frequency range) extracted every 10ms (analysis window is 25ms). Our input has a dimension 601x96[12],

- **MFCC** features only: we compute 13 MFCCs every 10ms to provide the CNN network with an input of dimension 601x13,

[10]We use filter window sizes *h* of [1, 3, 5, 7, 9] with 256 filters per size

[11]The detailed parameters of the filters are given in Figure 1

[12]The detailed parameters of the filters are given in Figure 1

- Joint (textual and signal) inputs (**EMBED+RAW-SIG**[13]): in that case, we concatenate last hidden layers of both textual and signal inputs (dotted lines of Figure 1).

5.1. Regression (baseline) and CNN performances

The lines *Regression* of table 2 show results obtained with combined features[14]. We can observe that the best performance is obtained with POS+LEX+LM features (MAE of 22.01%) while adding the SIG does not really improve the model (MAE of 21.99%). This inefficiency of SIG features in regression models was also observed in [5].

Model	Input	MAE	τ
Textual features			
Regression	POS+LEX+LM	22.01	**44.16**
CNN$_{Softmax}$	EMBED	**21.48**	38.91
CNN$_{ReLU}$	EMBED	22.30	38.13
Signal features			
Regression	SIG	25.86	23.36
CNN$_{Softmax}$	RAW-SIG	25.97	23.61
CNN$_{ReLU}$	RAW-SIG	26.90	21.26
CNN$_{Softmax}$	MEL-SPEC	29.11	19.76
CNN$_{ReLU}$	MEL-SPEC	26.07	24.29
CNN$_{Softmax}$	MFCC	**25.52**	**26.63**
CNN$_{ReLU}$	MFCC	26.17	25.41
Textual and Signal features			
Regression	POS+LEX+LM+SIG	21.99	45.82
CNN$_{Softmax}$	EMBED+RAW-SIG	**19.24**	**46.83**
CNN$_{ReLU}$	EMBED+RAW-SIG	20.56	45.01
CNN$_{Softmax}$	EMBED+MEL-SPEC	20.93	40.96
CNN$_{ReLU}$	EMBED+MEL-SPEC	20.93	44.38
CNN$_{Softmax}$	EMBED+MFCC	19.97	44.71
CNN$_{ReLU}$	EMBED+MFCC	20.32	45.52

Table 2. Regression *vs* CNN$_{Softmax}$ *vs* CNN$_{ReLU}$ evaluated at utterance level with MAE or τ on Test$_{pred}$

As for the use of textual features only, CNN$_{Softmax}$ and CNN$_{ReLU}$ are equivalent (better MAE but lower τ) than the regression model that uses engineered features. CNN$_{Softmax}$ shows better performance than CNN$_{ReLU}$. Concerning signal features only, ASR performance prediction is a difficult task with all MAE above 25%. However, among the different signal inputs to the CNN, simple MFCCs lead to the best performance both for MAE and τ. While the joint use of textual and signal features did not work for the regression baseline, the combination of inputs for CNNs lead to improved results. The best performance is obtained with CNN$_{Softmax}$ (EMBED + RAW-SIG) which outperforms a strong regression baseline (MAE is reduced from 21.99% to 19.24%, while τ is improved from 45.82% to 46.83%), *Wilcoxon Signed-rank Test*[15] confirms that the difference is significant with p-value of 4e-08.

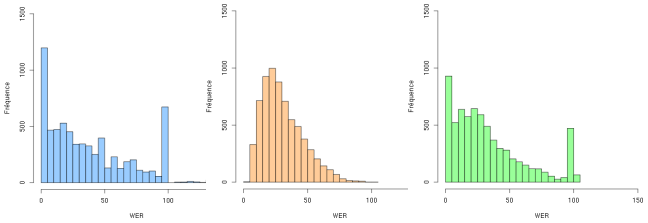
5.2. Analysis of predicted WERs

Table 3 shows the predicted WERs (at collection level) for both regression and (best) CNN approaches for different speaking styles (spontaneous and non spontaneous). Overall, the predicted WER on non spontaneous (NS) and spontaneous (S) speech is very good for the *CNN* approach. WER$_{Pred}$ is at -2.54% on non-spontaneous speech and at -4.84% on spontaneous speech. On the other hand, while efficient on non-spontaneous speech, regression fails to predict performance (-10,11%) on spontaneous speech.

	NS	S	NS + S
WER$_{REF}$	21.47	38.83	31.20
WER$_{Pred}$ Regression	**22.08**	28.72	25.82
WER$_{Pred}$ CNN$_{Softmax}$	18.93	**33.99**	**27.37**
#Utterances	3,1k	3,7k	6,8k
#Words$_{REF}$	49.8k	63.3k	113,1k

Table 3. Regression *vs* CNN$_{Softmax}$ predicted WERs (averaged over all utt.) per speaking style (NS/S) on Test$_{pred}$

Figure 2 analyzes WER prediction at utterance level[16] . It shows the distribution of speech turns according to their real or predicted WER. It is clear that CNN prediction allows to approximate the true WER distribution on Test$_{pred}$ while regression seems to build a gaussian distribution around the mean WER observed on training data. It is also remarkable that the two peaks at WER=0% and WER=100% can be predicted correctly by our CNN model.



(a) REF (c) Best REG (b) Best CNN

Fig. 2. Distribution of speech turns according to their WER: (a) real (b) predicted by regression (c) predicted by CNN

6. CONCLUSIONS

This paper presented an evaluation framework for evaluating ASR performance prediction on unseen broadcast programs. CNNs were very efficient encoding both textual (ASR transcript) and signal to predict WER. Future work will be dedicated to the analysis of signal and text embeddings learnt by the CNN and their relation to conditioning factors such as speech style, dialect or noise level.

[13] or EMBED+MEL-SPEC or EMBED+MFCC

[14] MAE using single POS, LEX, LM and SIG features is respectively 25.95%, 25.78%, 24.19% and 25.86%

[15] http://www.r-tutor.com/elementary-statistics/non-parametric-methods/wilcoxon-signed-rank-test

[16] Model outputs available on http://www.lne.fr/LNE-LIG-WER-Prediction-Corpus

7. REFERENCES

[1] Ayman Asadi, Richard Schwartz, and John Makhoul, "Automatic detection of new words in a large vocabulary continuous speech recognition system," *Proc. of International Conference on Acoustics, Speech and Signal Processing*, 1990.

[2] Sheryl R. Young, "Recognition confidence measures: Detection of misrecognitions and out-of-vocabulary words," *Proc. of International Conference on Acoustics, Speech and Signal Processing*, pp. 21–24, 1994.

[3] Benjamin Lecouteux, Georges Linarès, and Benoit Favre, "Combined low level and high level features for out-of-vocabulary word detection," *INTERSPEECH*, 2009.

[4] Matteo Negri, Marco Turchi, José GC de Souza, and Daniele Falavigna, "Quality estimation for automatic speech recognition.," in *COLING*, 2014, pp. 1813–1823.

[5] Shahab Jalalvand, Matteo Negri, Marco Turchi, José GC de Souza, Daniele Falavigna, and Mohammed RH Qwaider, "Transcrater: a tool for automatic speech recognition quality estimation," *Proceedings of ACL-2016 System Demonstrations. Berlin, Germany: Association for Computational Linguistics*, pp. 43–48, 2016.

[6] Karol J Piczak, "Environmental sound classification with convolutional neural networks," in *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*. IEEE, 2015, pp. 1–6.

[7] Tara N Sainath, Ron J Weiss, Andrew Senior, Kevin W Wilson, and Oriol Vinyals, "Learning the speech front-end with raw waveform cldnns," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[8] Ma Jin, Yan Song, Ian Mcloughlin, Li-Rong Dai, and Zhong-Fu Ye, "Lid-senone extraction via deep neural networks for end-to-end language identification," in *Proc. of Odyssey*, 2016.

[9] Dimitri Palaz, Mathew Magimai Doss, and Ronan Collobert, "Convolutional neural networks-based continuous speech recognition using raw speech signal," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4295–4299.

[10] Wei Dai, Chia Dai, Shuhui Qu, Juncheng Li, and Samarjit Das, "Very deep convolutional neural networks for raw waveforms," *CoRR*, vol. abs/1610.00087, 2016.

[11] Guillaume Gravier, Gilles Adda, Niklas Paulson, Matthieu Carré, Aude Giraudel, and Olivier Galibert, "The etape corpus for the evaluation of speech-based tv content processing in the french language," in *LREC-Eighth international conference on Language Resources and Evaluation*, 2012, p. na.

[12] Sylvain Galliano, Edouard Geoffrois, Djamel Mostefa, Khalid Choukri, Jean-François Bonastre, and Guillaume Gravier, "The ester phase ii evaluation campaign for the rich transcription of french broadcast news.," in *Interspeech*, 2005, pp. 1149–1152.

[13] Juliette Kahn, Olivier Galibert, Ludovic Quintard, Matthieu Carré, Aude Giraudel, and Philippe Joly, "A presentation of the repere challenge," in *Content-Based Multimedia Indexing (CBMI), 2012 10th International Workshop on*. IEEE, 2012, pp. 1–6.

[14] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[15] Andreas Stolcke et al., "Srilm-an extensible language modeling toolkit.," in *Interspeech*, 2002, vol. 2002, p. 2002.

[16] Martine De Calmès and Guy Pérennou, "Bdlex: a lexicon for spoken and written french," in *Proceedings of 1st International Conference on Langage Resources & Evaluation*, 1998, pp. 1129–1136.

[17] Olivier Galibert, "Methodologies for the evaluation of speaker diarization and automatic speech recognition in the presence of overlapping speech.," in *INTERSPEECH*, Frédéric Bimbot, Christophe Cerisara, Cécile Fougeron, Guillaume Gravier, Lori Lamel, François Pellegrino, and Pascal Perrier, Eds. 2013, pp. 1131–1134, ISCA.

[18] Helmut Schmid, "Treetagger— a language independent part-of-speech tagger," *Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart*, vol. 43, pp. 28, 1995.

[19] 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*, New York, NY, USA, 2010, MM '10, pp. 1459–1462, ACM.

[20] François Chollet et al., "Keras," `https://github.com/fchollet/keras`, 2015.

[21] Yoon Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.

[22] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.

[23] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. Aug, pp. 2493–2537, 2011.

[24] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, "librosa: Audio and music signal analysis in python," 2015.

[25] Matthew D. Zeiler, "ADADELTA: an adaptive learning rate method," *CoRR*, vol. abs/1212.5701, 2012.