Interspeech 2018
2-6 September 2018, Hyderabad

Word Emphasis Prediction for Expressive Text to Speech

Yosi Mass, Slava Shechtman, Moran Mordechay, Ron Hoory, Oren Sar Shalom, Guy Lev, David Konopnicki

IBM Research, Haifa, Israel

{yosimass,slava,moranm,hoory,orensr,guylev,davidko}@il.ibm.com

Abstract

Word emphasis prediction is an important part of expressive prosody generation in modern Text-To-Speech (TTS) systems. We present a method for predicting emphasized words for expressive TTS, based on a Deep Neural Network (DNN). We show that the presented method outperforms machine learning methods based on hand-crafted features in terms of objective metrics such as precision and recall. Using a listening test, we further demonstrate that the contribution of the predicted emphasized words to the expressiveness of the synthesized speech is subjectively perceivable.

Index Terms: word emphasis, speech synthesis, expressive text to speech, prosody, deep learning

1. Introduction

Generating natural and expressive prosody is considered to be one of the most important challenges in Text-To-Speech (TTS) systems. There are numerous perceptually distinct prosodic realizations of the same input text. The prosody determines the emotional state and attitude of speakers and helps bring clear messages to listeners by distinguishing the more important speech portions. The latter is usually realized by means of word emphasis [1].

Word emphasis patterns are speaker- and domain-specific. The word emphasis is either applied deliberately to convey a certain speaking style or used pragmatically to focus attention on particular words or the ideas associated with them. Doing so can clarify or even modify the meaning of a sentence. The mixed nature of word emphasis makes its prediction a challenging task that relies heavily on a reliably annotated and consistent voice corpus.

In this work, we explore the usability of fully automatic word emphasis prediction for expressive TTS system, when used to generate persuasive speech for an audience. Two capabilities are required for such a system. First, the system needs a prediction module that marks emphasized words within a text. Second, it must have the capabilities to realize that word emphasis. In this paper, we focus on the word emphasis prediction module. This module can be integrated in word-emphasis-enabled TTS systems such as [2, 3] to achieve a fully automated TTS system with emphasized words.

Emphasized words in speech can be used in two different scenarios: i) *detection* from spoken data and ii) *prediction* for TTS. While the former can exploit both acoustic and textual features [4, 5, 6], the latter can only use textual features. In general, word emphasis prediction models are rule-based [7] or trained on hand-crafted features using classic machine learning (ML) approaches [8]. In this work, we propose harnessing the power of Deep Neural Networks (DNN) for predicting emphasized words. Our network uses a combination of fully connected layers and a Recurrent Neural Network (RNN) with Bidirectional Long Short-Term Memory (BiLSTM) [9].

One of the difficulties in training a DNN lies in collecting large amounts of labeled data. Unfortunately, there are currently no available datasets for the discussed task. Although several resources with TOnes and Breaks (ToBI) labeling [10] are available, this data cannot be translated directly to emphasized word labels [11]. In [7, 8], the authors used a relatively small proprietary dataset comprising less than total of $6,500$ words. with word emphasis labeling.

We describe a larger labeled dataset that we created for training a DNN. The dataset contains about 20 hours of recorded data, which represent 9,461 sentences with a total of 168,409 words. Each sentence was annotated for word-emphasis binary labels by 4 professional labelers. We trained and tested a DNN on the labeled data using k-fold cross validation, and demonstrate that it outperforms a classic ML technique with hand-crafted features. To further enhance research in this domain, we release a subset of the dataset on the IBM Debater Datasets webpage [1].

To verify the effectiveness of the model, we performed a subjective listening test [12] by creating a test dataset with 50 random sentences, to which we applied our model for predicting emphasized words. We then used the TTS system of [2] to synthesize the sentences, once without emphasized words and once with the predicted emphasized words. We report the results of this listening test and show that the spoken data with emphasized words was significantly better in terms of expressiveness, while maintaining the naturalness of the original TTS.

To summarize, the contributions of the paper are the following. First, a DNN based model for predicting emphasized words from text. Second, a large speech corpora, labeled for emphasized words, and third, an expressive TTS system with prosody generation for emphasized words.

2. Related Work

Most previous work that detect emphasized words, are based on acoustic and prosodic features that exist in spoken data [4, 5, 6]. Our focus in this paper is different. We develop a model for predicting emphasized words from text, thus we can use text-based features only.

There are relatively few previous works that predict emphasized words. In one, the authors present a rule-based approach [7]. In another [8], a ML classifier is used. To predict emphasized words, the classifier uses text-based features such as part-of-speech (POS), information content, and position in the sentence. Strom et al. [12] proposed using a simple pitch accent predictor based on the ratio between the number of times a word was pitch-accented in some large labeled corpus, and the total number of its appearances in the corpus. We show that our

[1] http://www.research.ibm.com/haifa/dept/vst/debating_data.shtml

10.21437/Interspeech.2018-1159

method, which is based on DNN with word embedding [13], outperforms approaches that are based on the above-mentioned features.

Some of the other works cited here [14, 15] also use DNN, but they use the technique for detecting emphasized words in spoken data, and not for prediction.

Obtaining labeled data of emphasized words for training a model is subjective, error prone, and labor intensive. Usually, several labelers are required to increase the annotation reliability. For example, three labelers annotated a corpus of 6,434 words with syllable prominence in [7]. The word emphasis was then derived by taking the maximum syllable prominence in each word. Another smaller-scale emphasis prediction work [8] is based on an annotated corpus of four childrens stories (with a total of 2,906 words), using just one labeler. In that work, the labels included pitch accents, while the emphasized words were determined by some heuristics on the pitch accents of the word and its neighbors.

As opposed to the cited prior art, we used a much larger dataset with 168,409 words, fully annotated by four distinct labelers.

3. Word Emphasis Prediction

The proposed architecture (Figure 1) receives a batch of sentences as input and processes each sentence as follows.

The **Word Embedding Layer** extracts a feature vector for each word using a word embedding matrix. We use the Google's pre-trained w2v [13] that represents the semantic meaning of the words. The pre-trained matrix allows us to benefit from training on a very large unlabeled data set.

The **Fully Connected (FC) Layer** translates the original word embeddings into new representations to better fit the task at hand. It applies a linear transformation to the word embeddings, followed by *tanh* as a non-linear activation function.

The **Bidirectional RNN Layer** captures the context of each word when predicting whether it should be emphasized. Clearly, emphasizing a word depends on its context [16]. We use LSTM [9] to capture the consecutive elements in a sequence (in our case, words in a sentence). The learned representation of each word is dependent on the elements that precede it. To capture subsequent words, we use bidirectional LSTM (BiLSTM). As a result, the output of this layer captures the meaning of each word together with its relevant context.

The **Prediction Layer** is a fully-connected layer used to translate the representation computed in previous layers into a probability score that represents the probability of the word being emphasized. This is done by computing the sigmoid on the inner product between a learned weight vector β_1 and the output of the previous layer x plus a bias term. Namely, $sigmoid(\beta_1 x + \beta_0)$.

Figure 1: *Model architecture*

The network can be trained on an annotated voice corpus with binary word emphasis labels attached to each word (i.e., the emphasized words are labeled with 1, and the rest with 0). The labeled data that we used is described in Section 5 below. The loss function is defined as the weighted cross entropy between the predictions and the actual labels.

$$loss = \sum_{x \in X} [label_x \cdot (-log(prediction_x)) \cdot \mathtt{pw} + (1 - label_x) \cdot (-log(1 - prediction_x))] \quad (1)$$

where X represents all words in all training sentences and the hyperparameter `pw` is used as a weight for compensating the positive (i.e., emphasized) words, due to their unbalanced ratio among all words (see Table 1). Another method for handling unbalanced data is to apply over/under sampling in the training set and fix the prediction bias [17] as we did in Section 6 for the Logistic Regression classifier (Equation 3). However, since our DNN model depends on context, it is not feasible to over/under sample words inside a sentence.

Once the model is trained, it can be used for predicting emphasized words in a new sentence as follows. A sentence is fed into the network, which outputs a prediction value for each word, as described above. All words with $prediction \geq 0.5$ are then defined as emphasized words. More details on the network and the selection of the hyper-parameters (e.g., sizes of each layer) are given in Section 6 below.

4. Prosody Generation

The predicted emphasized words are given as input to a speech-synthesis engine. In this work we use the IBM concatenative unit-selection system [2, 18] which is adapted to utilize the predicted emphasized words.

The engine generates word-emphasis prosody using two stacked BiLSTM networks. The first one generates a generic fine-grained prosody at sub-phonemic resolution. The second one generates a rough piecewise linear prosodic trajectory at syllable resolution to realize the word emphasis, augmenting the generic sub-phonemic prosody model [2]. The predicted prosody targets serve both for unit selection and for post-selection signal modification by Pitch-Synchronous Overlap and Add (PSOLA) [18].

The control of word-emphasis strength in the system is enabled by adopting a special data normalization technique [2]. During the training, a moving average is extracted from the prosodic target components in the vicinity L_m around each syllable m. During the prediction phase, the weighted sum of the moving average and the maximum in L_m (denoted by $s^\alpha(m)$), is added back as a post-processing stage at the m-th syllable. Formally,

$$s^\alpha(m) = \alpha \cdot avg_{q \in L_m}(s_{neu}(q)) + (1 - \alpha) \cdot max_{q \in L_m}(S_{neu}(q)) \quad (2)$$

where $s_{neu}(q)$ is a corresponding predicted neutral component, obtained from the generic fine-grained prosody prediction, and L_m is a subset of indices in the vicinity of the m-th syllable. We used $\alpha = 0.6$ and $\alpha = 0.8$ in our experiments (Section 6.1).

The word-emphasis prosody model was trained on the same labeled voice corpus as described in Section 5, but with the agreement of one out of four labelers to maximize the amount of data for prosody model training. Originally, the model was developed for user-controlled word emphasis [2]. To adapt it to

the error-prone automatic word-emphasis prediction, additional constant-window Gaussian smoothing was applied on the predicted pitch trajectory (with window size of about 120 ms).

5. Labeled Speech Corpora

We recorded about 20 hours of a professionally native female US English speaker. The recorded corpus is comprised of topic-specific claims and evidences [19, 20, 21], which the speaker was instructed to read in a persuasive and lively manner. In turn, based on the recorded speech, each sentence was annotated by 4 professional labelers for emphasized words. The labelers got the original text and the recorded data with the following guidelines. i) Emphasized words are words that clearly stand out in the speech relatively to most of the words in the sentence. ii) Label only based on the speech, and not based on the importance of the word in the text.

The total corpus contains 9,461 sentences, with a total of 168,409 words. The kappa agreement [22] between the labelers was 0.35. This indicates the subjectivity of the task. Table 1 shows the statistics regarding the different levels of agreement between the labelers. The number and ratio of emphasized words agreed upon by even three labelers is quite low, so we used agreement level 2 as the ground-truth.

Agreement	Total words	Emphasized	Ratio
1	168,409	26,307	15.6
2	168,409	9,093	5.4
3	168,409	4,196	2.5
4	168,409	1,530	0.01

Table 1: *Labeling statistics*

Table 2 lists some examples of labeled emphasized words from the ground-truth (shown in bold).

1	The policy is considered as a **great** success in helping to implement China's **current** economic growth
2	The pursuit of **doping** athletes has turned into a modern day **witch** hunt
3	Qualifications should be the **only** determining factor

Table 2: *Example labeled emphasized words*

Table 3 shows the top labeled emphasized words, sorted by absolute counts (left column) and by relative counts (right column). Each word is presented with the number of times it was emphasized, and with the ratio to its total number of occurrences in the collection. In the right column we show only words that appear at least 20 times in the collection. Note that words are presented with their original case, since our prediction model is case-sensitive. The top emphasized words in term of absolute counts are mainly negative terms and adjectives, while the top emphasized words by ratio are adjectives and numbers (e.g., thousands, millions).

6. Experiments

To verify the quality and expressiveness of the predictors of the emphasized words when used in the TTS system [2], we conducted a listening test as described below. As a preliminary step for tuning the parameters, we ran a five-fold cross-validation on the 9,461 labeled sentences. In each fold, 20% were dedicated

Absolute (count, ratio)	Relative (count, ratio)
not (**252**, 0.26%))	thousands (19, **0.73**%)
all (**216**, 0.43%))	completely (17, **0.68**%)
no (**101**, 0.3%))	half (22, **0.66**%)
very (**89**, 0.33%))	All (25, **0.59**%)
do (**85**, 0.15%))	millions (20, **0.58**%)
should (**78**, 0.2%))	everything (15, **0.57**%)
any (**69**, 0.33%))	No (19, **0.57**%)
is (**47**, 0.01%))	absolutely (12, **0.57**%)
one (**46**, 0.1%))	ever (24, **0.56**%)
every (**44**, 0.41%))	everyone(13, **0.54**%)

Table 3: *Top emphasized words*

for test, while the other 80% were divided between training (70%) and dev (10%). The model was implemented in TensorFlow[2].

We first fine-tuned the weight parameter `pw` (Equation 1), which controls the trade-off between recall and precision. A higher value for `pw` is expected to increase recall (since it boosts the positive examples), but in exchange it decreases precision. We used the full DNN as depicted in Figure 1. For the pre-trained embeddings, we used Google's word2vec embedding [13] of dimension 300, that were trained on a corpus of 100B words[3]. We used the FC layer with output dimension 200, followed by a *tanh* activation. We used hidden size of 128 for the LSTM in the third layer and, since we use bidirectional LSTM, the output of this layer is 256. The prediction layer converts from 256 to 1. We further used a dropout of 0.75 and the Adagrad optimizer [23] with an initial learning rate of 0.5. We applied early termination and take the model parameters, which achieved the best results on the dev set. We refer to this configuration as FC200-BiLSTM128.

In all experiments of all methods in this section, we filtered out pronouns and prepositions from being emphasized, even if they were predicted to be so. In addition, we filtered very frequent words such as *all* and *very* from being emphasized. This is because those words are already caught and realized to some extent with the basic prosody generation model [18]. We discovered that the explicit emphasis prediction of those words does not contribute much on average, but occasionally sounds exaggerated. Furthermore, we avoided emphasizing consecutive words.

Figure 2 shows the recall, precision, and F1-measure for various values of the `pw` parameter for the selected hyper parameters. The results are averaged over the five folds. As ex-

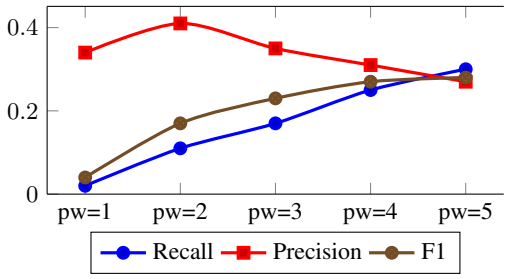

Figure 2: *Effect of `pw`. Network uses three layers: FC (200d), BiLSTM(128d) and another FC (1d)*

[2] `https://www.tensorflow.org`
[3] `https://code.google.com/archive/p/word2vec`

pected, increasing pw results in a higher recall and a lower precision. For example, pw=2 results in recall of 0.11 and precision of 0.41, whereas for pw=3, recall increases to 0.18 but precision drops to 0.33. Note that the specific task at hand is more precision oriented (we prefer emphasizing less words but with higher accuracy). Therefore, we selected pw=2.

Next we tried different configurations of the network as shown in Figure 3. All methods use pw=2. The full network (FC200-BiLSTM128) outperforms partial networks (where the embedding and prediction layers are fixed and the internal two layers are changed), both in terms of precision and recall. For example, replacing the BiLSTM layer with an LSTM (FC200-LSTM128) achieved a precision of 0.40 and recall 0.08. Removing the BiLSTM layer completely (FC200) achieved precision of 0.38 and recall 0.09, while removing the FC layer completely (BiLSTM128) achieved precision of 0.35 and recall of 0.05. We also tried other sizes (e.g., FC128-BiLSTM50) and more than one BiLSTM layers (not shown in the figure) but they were a bit inferior to the FC200-BiLSTM128 configuration.

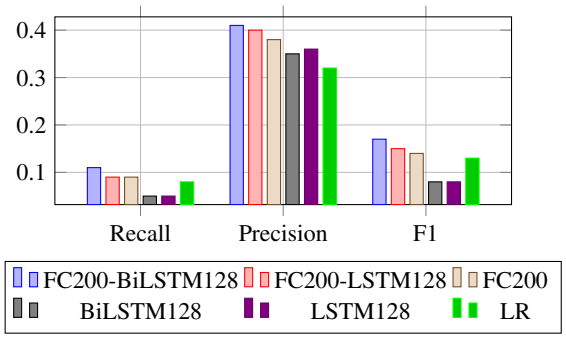

Figure 3: *Compare different networks,* pw=2

The various configurations of the DNN, were compared to a logistic regression (LR) classifier (last bars in Figure 3), which was trained with the following features [8]. POS tagging (14 categories, coded with 1-hot vectors), information content, word offset in sentence, and a word negation indicator (extracted from a dictionary of negation shifters such as not, shouldn't, never, etc.). We replaced the ID of the word as used in [8] with a 300-D word embedding, as we used in the DNN method, to further improve the performance of [8]. For each word, we extracted the above features for the word itself, as well as for the previous and subsequent words in the sentence. Overall, the number of features for each word are 951 (317 for each of the word itself, previous word and next word).

Due to the imbalanced number of positive vs negative examples (the emphasized words are only 5.4% of the total words), we down-sampled the negative examples. We tried various sampling rates, $s \in [1, 20]$, where sample rate s means that the number of negative examples in a sentence is s times the number of positive examples. In this way, at test time we keep all learned parameters, and only modify the bias term β_0 [17]

$$\beta_0 = \beta_0 + \log(\frac{\pi}{1-\pi}) - \log(\frac{\widetilde{\pi}}{1-\widetilde{\pi}}) \quad (3)$$

where π is the original ratio of the positive examples (0.054 as appears in Table 1) and $\widetilde{\pi}$ is the modified ratio, given by $1/(1+s)$. The best result was obtained for s=10 with a precision rate of 0.32 and a recall level of 0.08, which are much lower than the rates achieved by the DNN – precision 0.41 and recall 0.11.

6.1. Listening Test

To evaluate the proposed systems within the expressive TTS framework, a subjective listening evaluation was conducted on Amazon Mechanical Turk (AMT), in the form of a mean opinion score (MOS) test [24]. The MOS test included 50 out-of-corpus stimuli per system and 25 votes per stimulus, provided by 105 paid anonymous native speakers. A single subject was removed as a result of the outlier rejection [24]. In addition to the neutral prosody reference model (NOEMPH), the prosody prediction systems with various emphasis strength strengths (EMPH06 for $\alpha = 0.6$ and EMPH06 for $\alpha = 0.8$, in Equation 2 above) were applied using our word prediction model with the best parameters (FC200-BiLSTM128), trained on 90% of the labeled corpus (Section 5) with 10% left for dev for early termination of the training.

In addition to the standard MOS test in which the subjects were asked to rate the quality and naturalness of the synthesized speech on a five-grade qualitative scale (Poor, Bad, Fair, Good, Excellent), the users were asked to assess the expressiveness of the synthesized samples. A five-grade scale was utilized for this test too, with its values explained to the subjects (very non-expressive, non-expressive, neutral, expressive, very expressive). The evaluation scores are reported along with their 95% confidence interval and p-values against the reference (NOEMPH) in Table 4. The bold results are statistically significant ($p < 0.05$) compared to the reference system (NOEMPH). Both word emphasis systems significantly improve subjective expressiveness, while preserving the original quality and naturalness. The system with stronger emphasis (EMPH08) resulted in slightly higher expressiveness and slightly lower quality than the system with weaker emphasis (EMPH06), but the differences were not found to be statistically significant.

MOS	NOEMPH	EMPH06	EMPH08
Expressiveness	3.65 ± 0.05	**3.72 ± 0.05 (p = 0.017)**	**3.74 ± 0.05 ($p < 0.01$)**
Quality	3.82 ± 0.05	3.84 ± 0.05	3.82 ± 0.05

Table 4: *MOS results for word emphasis synthesis with* $\mu \pm 95\%$ *confidence and p-value against NOEMPH*

7. Summary and Future Work

In this work, we presented a fully automated TTS system for improving the perceived expressiveness of synthesized speech. The system is built from two components: i) a word emphasis prediction model and ii) a prosody generation model that utilizes the predicted emphasized words. Subjective experiments demonstrated that the synthesized speech based on this model indeed was perceived as more expressive, while preserving the quality and naturalness of the original.

For future work, inspired by [25], we plan to learn a personalized model for each labeler. Another possible area of study is to explore multi-voice training of the proposed word emphasis models and their application to unseen voices.

8. Acknowledgment

We would like to thank Zvi Kons, Raul Fernandez and Michal Jacovi for their support in creating the labeled speech corpora.

9. References

[1] S. Pan and K. R. McKeown, "Word informativeness and automatic pitch accent modeling," in *Proceedings of EMNLP*, 1999.

[2] S. Shechtman and M. Mordechay, "Emphatic speech prosody prediction with deep lstm networks," in *Proceedings of ICASSP, IEEE*, 2018.

[3] Z. Malisz, H. Berthelsen, J. Beskow, and J. Gustafson, "Controlling prominence realisation in parametric dnn-based speech synthesis," in *Proceedings of Interspeech*, 2017.

[4] R. Fernandez and B. Ramabhadran, "Automatic exploration of corpus-specific properties for expressive text-to-speech: A case study in emphasis." in *6th ISCA Workshop on Speech Synthesis*, 2007.

[5] T. Mishra, V. R. Sridhar, and A. Conkie, "Word prominence detection using robust yet simple prosodic features," in *Proceedings of Interspeech*, 2012.

[6] Y. Chen and R. Pan, "Automatic emphatic information extraction from aligned acoustic data and its application on sentence compression," in *Proceedings of the 31th AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[7] C. Widera, T. Portele, and M. Wolters, "Prediction of word prominence," in *EUROSPEECH*, 1997.

[8] J. M. Brenier, D. M. Cer, and D. Jurafsky, "The detection of emphatic words using acoustic and lexical features," in *Proceedings of Eurospeech*, 2005.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural compu- tation*, vol. 9, pp. 1735–1780, 1997.

[10] K. Silverman, M. Beckman, J. Pitrelli, M. Osten-dorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirschberg, "Tobi: A standard for labeling english prosody," in *2nd International Conference on Spoken Language Processing (ICSLP)*, 1992, pp. 867–870.

[11] J. Hirschberg, "Pitch accent in context: Predicting intonational prominence from text," *Artificial Intelligence*, vol. 63, 1993.

[12] V. Strom, A. Nenkova, R. Clark, Y. Vazquez-Alvarez, J. Brenier, S. King, and D. Jurafsky, "Modeling prominence and emphasis improves unit-selection synthesis," in *Proceedings of Interspeech*, 2007.

[13] T. Mikolov, K. Chen, G. Corrado, and J.-f. Dean, "Efficient estimation of word representations in vector space," *ArXiv e-prints*, Mar. 2013.

[14] A. Rosenberg, R. Fernandez, and B. Ramabhadran, "Modeling phrasing and prominence using deep recurrent learning," 09 2015.

[15] A. Schnall and M. Heckmann, "Comparing speaker independent and speaker adapted classification for word prominence detection," in *IEEE Spoken Language Technology Workshop (SLT)*, 2016.

[16] K. Lee, "Sentence stress in information structure," *Oenoehag*, vol. 27, 2013.

[17] G. King and L. Zeng, "Logistic regression in rare events data," *Political analysis*, vol. 9, no. 2, pp. 137–163, 2001.

[18] R. Fernandez, A. Rendel, B. Ramabhadran, and R. Hoory, "Using deep bidirectional recurrent neural networks for prosodic-target prediction in a unit-selection text-to-speech system," in *Proceedings of Interspeech*, 2015.

[19] E. Aharoni, A. Polnarov, T. Lavee, D. Hershcovich, R. Levy, R. Rinott, D. Gutfreund, and N. Slonim, "A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics," in *ArgMining@ACL*, 2014, pp. 64–68.

[20] R. Levy, Y. Bilu, D. Hershcovich, E. Aharoni, and N. Slonim, "Context dependent claim detection," in *The 25th International Conference on Computational Linguistics, COLING*, 2014.

[21] R. Rinott, L. Dankin, C. A. Perez, M. M. Khapra, E. Aharoni, and N. Slonim, "Show me your evidence - an automatic method for context dependent evidence detection," in *EMNLP*, 2015, pp. 440–450.

[22] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement 20 (1)*, pp. 37–46, 1960.

[23] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, 2011.

[24] C. Z. F. Ribeiro, D. Florłncio and M. Seltzer, "crowdmos: An approach for crowdsourcing mean opinion score studies," in *Proceedings of ICASSP, IEEE*, 2011.

[25] G. M. Y, G. Varun, D. A. M, and H. G. E, "Who said what: Modeling individual labelers improves 2017.