



Liulishuo's System for the Spoken CALL Shared Task 2018

Huy Nguyen, Lei Chen, Ramon Prieto, Chuan Wang, and Yang Liu

Liulishuo
www.liulishuo.com/en

{huy.nguyen, lei.chen, ramon.prieto, chuan.wang, yang.liu}@liulishuo.com

Abstract

The Spoken CALL (Computer-Assisted Language Learning) 2018 shared task requires systems to automatically accept or reject each single-sentence spoken response depending on whether the response is correct given a prompt. Spoken responses are first recognized into texts and then classified as 'accept' or 'reject' based on their language and meaning. This paper describes our system for the shared task. We focused on improving speech recognition performance, developing a rich set of features to capture the linguistic and semantic meaning of the responses, and optimizing classification results for various factors (training set, n-best hypotheses of speech recognition, decision threshold, model ensemble). Our system achieves the best performance among the participating teams.

1. Introduction

As defined in [1], CALL is "the research for and study of applications of the computer in language teaching and learning." Since the first implementations of CALL in 1960's, the tremendously fast development of technologies has been transforming CALL from simple stimuli and responses by "computer tutors" to leveraging the Internet, multimedia, and Artificial Intelligence [2, 3].

In speaking practice, CALL systems utilizing automatic speech recognition (ASR) technology offers new abilities to process learners' responses for error detection and automated feedback-generation [4, 5, 6]. As an initiative to further develop related technologies, a shared task for the spoken CALL research was presented in 2016 and participating systems were reported in the ISCA SLATE 2017 workshop [7]. The task is to provide feedbacks to prompt-based spoken responses by English learners using the CALL-SLT system [6]. Participating systems need to accept responses with correct meaning and language usage, and reject others. Following the success of the first shared task with 20 submissions from 9 groups, the second edition with new resources and updated training data was announced in October 2017 and the test data was released in February 2018 [8]. Similar to the previous edition, the task organizers provide the audio data, ASR outputs, and reference response grammar. There are two tasks: the *text task* where the ASR outputs for the spoken responses are provided by the organizers, and the *speech task* where participants can use their own recognizers to process audio responses. This paper describes our system developed for both the text and the speech tasks.

The key components of our system, common to the text and the speech tasks, are the feature extraction and classification modules. We derived over 50 features including those borrowed from prior studies as well as newly proposed ones in this study, which represent the linguistic quality and semantic correctness of the responses. The classifier was an ensemble model, with final decision threshold optimized based on the development

set. In addition, we augmented the reference response grammar to increase its coverage. For the speech task, we made the following extra efforts. First, we improved ASR performance by cleaning training data and applying language model (LM) adaptation. Second, since recognition errors negatively impact the subsequent classifier, we leveraged multiple ASR outputs (n-best lists). The shared task results showed that our models achieved competitive performance, ranking the first in both text and speech tasks.

2. Spoken CALL shared task

The data used in the shared task are prompt-response pairs collected from an English course running CALL-SLT developed for German-speaking Swiss teenagers [8]. Prompts in the course are written texts in German associated with animation video clips each showing an English native speaker asking a question. Each response is labeled as "correct" or "incorrect" for its linguistic correctness (language) and its meaning respectively. A response is *accepted* when it is correct in both language and meaning given the prompt. Otherwise, it is *rejected*. It is possible that a response is correct only in one aspect. The following shows an example of question, prompt in German (with English translation), and accepted student response:

Question: *How many nights would you like to stay at our hotel?*

Prompt: *Frag: Zimmer fr 3 Nchte. (Ask: room for 3 nights)*

Response: *I would like to stay for three nights.*

Table 1 shows the information of the data from the 2017 and 2018 tasks. In #reject column, numbers in parentheses are ungrammatical responses with correct meaning. Different from the first edition, the second edition of the shared task has additional annotations [8]. In particular, the organizers used the top four systems from the first edition to predict the responses and split the data into three sets, i.e., A, B, and C. For set A, at least three systems agreed with each other and at least one annotator supported the systems' prediction. Set B includes responses for which three annotators agreed and at least one system predicted the same labels. Set C consists of the remaining responses. While the second edition provides larger training data than the first one, participating teams have choices to use all or part of training data.

For performance evaluation, the organizers propose the D metric that takes into account different types of *false accepts* when evaluating the prediction output [8]. In particular, between the two types of mis-predictions, i.e., accepting incorrect responses with wrong meaning vs. accepting responses with right meaning but incorrect grammar, a higher penalty weight, $k=3$, is applied to the former type, which is a more serious error.

3. Our system

We tackle the Spoken CALL challenge by solving a text classification problem that gives each student response a label *accept*

Table 1: Numbers of accepts/rejects in different datasets.

Data set	#accept	#reject	Total
2017 Training	3,880	1,342 (802)	5,222
2017 Test	716	279 (159)	995
2018 Training A	4,225	1,302 (543)	5,526
2018 Training B	90	783 (382)	873
2018 Training C	103	196 (165)	299
2018 Test	750	250 (139)	1,000

or *reject* depending on whether the response is linguistically and meaningfully correct. We develop a text classifier based on an augmented grammar resource, a rich set of features, and an ensemble learning model. In the text task, we use the ASR outputs provided by the shared task’s organizers, while in the speech task, we use the ASR outputs by our own ASR system trained on the shared task data.

3.1. Prediction features

Given a text form of the response, we extract 54 features for text classification. First, we implement the syntactic and semantic features proposed in [9] that achieved good performance in the first edition of this challenge. Next, we propose new features enabled by topic models and grammatical error detection. The features used in our models are described below, with the number of features shown for each group:

Language model scores (14): The authors in [9] implemented different language models (LMs) based on words and their part-of-speech (POS) tags from correct responses, incorrect responses, and sample responses from the grammar resource. We supplement this feature group with LMs obtained based on production rules, e.g., $S = NP + VP$, and dependency rules, e.g., *nsubj(like-3, i-1)*, extracted from the syntactic parsing results using Stanford CoreNLP [10]. We expect the LMs trained on syntactic parses help capture the linguistic structure, and thus can differentiate grammatical sentences from ungrammatical ones. As in [9], we also use two LMs trained on the words and POS tags extracted from ungrammatical sentences in the two corpora used for grammatical error correction [11, 12]. Overall there are 14 features in this group.

Prompt-specific LM scores (2): This feature group uses word-based and POS-based LMs trained on responses for each prompt [9].

N-gram matching (5): Given a response, we count the total numbers of matching raw and lemmatized unigrams, bigrams, and trigrams with each sample response from the reference grammar, and then the numbers are averaged by the number of sample responses [9]. In addition, we also calculate numbers of unigrams, bigrams, and trigrams in the response that are not found in the reference grammar.

Word embedding (6): As proposed in [9], sample responses of each prompt are used to train different word embeddings using both a skip-gram and continuous bag-of-words training algorithms, with embedding dimensions of 30 and 50. We add embedding with dimension of 15 to obtain total six word embedding models per prompt. Given a response, the maximum similarity between it and the sample responses is calculated for each embedding model.

Topic models (1): We use a Latent Dirichlet Allocation implementation [13] to learn a topic model from the sample responses in the reference grammar, and then calculate the mini-

mum similarity between the topic distributions of the response and sample responses. We expect the topic model and word embedding features can capture the semantic similarity between the response and the sample responses in the grammar file.

Prompt-based (7): Because a response must satisfy the content of the provided prompt, we count the number of prompt words missing from the response, and the average missing prompt words normalized by the response’s length. In addition, we include 5 features counting the numbers of missing prompt words for the following five POS tag sets: nouns, verbs, modal verbs, determiners, and prepositions, since we observe that these are usually more important for linguistic and meaning correctness.

Length (6): We calculate numbers and ratios of tokens and unrecognized tokens (marked as “**”) for each response, length ratio of the response with respect to the average length of sample responses, and whether the response has any words or not.

Parse score (1): This is the ratio of the parse score of the response (returned by Stanford CoreNLP [10]) and the average parse score of sample responses.

Grammatical and spelling errors (12): We apply our in-house grammar error counting tool that is based on neural network models [14] and extract features measuring 11 types of typical L2 learners’ grammar errors, e.g., article, verb form, subject verb agreement. In addition, we count number of spelling errors as described in [9].

3.2. Machine learning models

Several types of machine learning models have been used in the first edition of this challenge, for example, Support Vector Machine (SVM), K-Nearest Neighbor models, and Feed-Forward Neural networks [9, 15, 16].

Our experiments on the data from the 2017 text task show that tuning a model’s parameters for higher conventional metrics, e.g., accuracy or area under curve (AUC), does not always translate to better D scores. Therefore, rather than using one model and tuning its parameters, we follow a widely-accepted practice in many machine learning tasks to use an ensemble of diverse models. In particular, we utilize four types of models as base classifiers, including Logistic Regression (LR), Random Forest (RF), SVM classifier (SVC), and XGBoost Gradient Boosting Tree model [17]. Implementations are available in the Scikit-learn toolkit [18]. On top of these base classifiers, we use a VotingClassifier to do a soft weighting to aggregate the base classifiers’ probability outputs

Given the 54 input features, we perform a model-based recursive feature extraction (RFE) operation to select a subset of features (K). A Random Forest model is used inside the RFE operation. Also, K value is automatically determined by using the CV-RFE that runs a separate cross-validation on the training set to find the optimal K . When converting our model’s final probability output to the required *accept/reject* decisions, we use a threshold t and reject the input response if its prediction probability is lower than t . A lower threshold means that more correct utterances will be accepted and thus yielding a lower correct utterance rejection rate (CRej). Since CRej is the denominator in the D metric [8], a well-controlled lower CRej value tends to result in higher D score.

3.3. Augmenting grammar resource and cleaning the data

Prior work from the last challenge has shown that text classification performance can be improved by cleaning input data and augmenting the reference grammar in a bootstrapped way

Table 2: WER on the 2017 test set. Δ column shows relative (%) WER reduction compared to the baseline.

#	ASR model	WER	Δ
1	Qian et al. [19]	15.6	
2	Baseline	15.7	
3	(2) + 2018 data	14.3	8.92
4	(3) + silent/indistinct removal	13.8	12.10
5	(4) + prompt-specific LM	13.4	14.65

[19]. We, however, observed that expanding the grammar with correct responses from training data made our classifier overfit the training data, possibly due to the dominance of LM-related features. Therefore, we perform some simple pre-processing to increase the coverage of the reference grammar without relying on the responses from the training data too much.¹

First, for each sample response in the grammar, we replace a verb’s contraction form with its complete-word form, and vice versa, and added to the grammar if the new response did not exist before. For example, “I am” generates “I’m”, and “they’re” has replacement “they are”. Second, following [19], we remove greeting/feeling words at the beginning of the responses, e.g., “thank you”, “yes”, “sorry”. Third, for the prompts with regular expressions (e.g., “at (three (p m | pm) | three o’clock)”), we instantiate the expressions for complete sets of prompts, which are used to extract prompt-based features. Our augmented grammar consists of over 56K responses, nearly five time larger than the original grammar.

For feeding our text classifier with higher quality inputs, we also clean ASR outputs as suggested in [19]: removing filler sounds (e.g., “uhm”, “mmm”), repeated words (“i want i want”), greeting/feeling words, and tokens marked unrecognized (after counting). We expect that the removed parts in a response should not impact human judgment, but make the cleaned text easier to compare against the grammar.

3.4. Improving speech task

Besides using the above enhancements that work for both text and speech tasks, we improve the speech task by increasing ASR accuracy and exploiting n-best hypotheses. To the best of our knowledge, this is the first time that potential benefit of using n-best hypotheses is investigated for this challenge.

Our ASR system uses Kaldi [20] and is based on the winning ASR system in the 2017 shared task and [19], which is provided as the baseline in the 2018 shared task. We improve ASR performance from three aspects: (1) we add 2018 data for model training; (2) we clean the training data by filtering out undesirable utterances, i.e., silent or those with indistinct words; (3) during ASR decoding for each utterance, we use a weighted interpolation of the generic LM trained on the entire training data set ($\lambda_{generic} = 0.8$) and the prompt-specific LM ($\lambda_{prompt} = 0.2$) that is trained only on the responses of the corresponding prompt. Table 2 shows the word error rate (WER) of our system on the 2017 test set. The first row presents the ASR performance reported in [19]. The baseline is our reproduction of their system. The following rows show the improvements brought by these efforts.

Since recognition errors impact the subsequent text classification module, we explore using ASR n-best list as inputs

¹We use the first version provided in 2017 rather than the expanded grammar in 2018 shared by [19].

Table 3: D scores on 2017 test set using different models ($t = 0.5$). Models are trained using 2017 training data.

Model	text	speech
LR	4.365	6.636
SVC	4.359	8.329
RF	3.339	6.768
XGB	4.115	8.005
Ensemble (E)	4.376	8.895

for classification. In many spoken language processing tasks, using more ASR hypotheses in the form of n-best lists or lattices has been shown to be beneficial. On the 2017 test set, we observe that the oracle WER using 2-best hypotheses is 9.4%, which is significantly better than the 1-best, shown in Table 2. We expect that using more ASR candidates may systematically address some problems due to incorrectly recognized words (if any) in ASR outputs (rather than errors caused by the students). ASR N-best hypotheses were extracted from lattices using Kaldi’s tools. When using n-best lists, we first compute the edit-distance between each hypothesis and each sample response from the reference grammar. The ASR hypothesis with the smallest edit-distance is then used as the input of the text classifier. Then the following feature extraction steps are the same as when using just the 1-best ASR outputs.

On the 2017 test data, we find that when using 1-best ASR output for training and top-2 ASR hypotheses in testing, we achieve the highest performance. However, using top-2 ASR hypotheses to build training data performs worse. This may be explained as an issue of over-fitting when the training data is too similar to the reference grammar, or our current LM features extraction, which has very limited negative examples.

4. Experiment results

In this section, we first describe experiments to optimize our machine learning models. Given the lessons learned, we performed the text and speech tasks with the optimized text classifier and the improved ASR system.

4.1. Model development

In order to evaluate the base classifiers, we use the training and test sets provided in the 2017 Spoken CALL text and speech tasks [7]. We use the speech recognition outputs provided by the task organizers in the text task, and our ASR system for the speech task. Table 3 shows the D values of different machine learning models with decision threshold $t = 0.5$. The results confirm that the ensemble model obtains the highest D values in both tasks. Therefore, we use the ensemble model in all of our experiments.

Different from 2017, the training data in 2018 is split into three sets. Based on how the *accept/reject* labels are generated, we expect that the human generated labels in set B and C are noisier than A. Therefore we compare the effect of using different data sets for model training. Table 4 shows the results using the ensemble model, with a threshold of 0.5. We can see that the combination of 2017 training set and 2018 set A yields the best D score. Therefore, we use this combination as the training data in the following experiments.

Next, we adjust threshold t to optimize D values. Using 2017 test data for development, results are shown in Table 5.

Table 4: *D* scores on 2017 test set using different training sets. Ensemble model is used, with threshold $t = 0.5$.

Training set	text	speech
2017	4.376	8.895
2017 + 2018 A	4.718	9.791
2017 + 2018 A, B, C	3.813	8.749

Table 5: *D* scores with different decision thresholds.

Decision threshold	text	speech (1-best)	speech (2-best)
$t = 0.5$	4.718	9.791	10.641
$t = 0.4$	6.033	12.126	15.181
$t = 0.35$	6.818	13.872	20.832

For the speech task, we show results using 1-best and 2-best ASR hypotheses to create test data respectively. As we can see, decreasing the decision threshold generally helps to obtain higher *D* scores. When using 2-best ASR hypotheses to select responses in test data, we observe consistent performance gain compared to using 1-best. The best *D* score is obtained when using $t = 0.35$, which is significantly higher than the top result of the 2017 challenge [19]. In our experiments, we also observe that when t was too small, the number of rejections for incorrect responses (IREj) was very small, which may violate the task’s requirements (IREj > 0.25). Therefore, we keep 0.35 as the minimum t in our experiments.

Among 54 original features, over 40 were selected by CV-RFE to be used by our ensemble model. Top features reveal the dominance of language-model and similarity-score features. Besides, the appearance of prompt-based and length features supports our belief in their necessity for prompts with few numbers of responses.

4.2. Spoken CALL 2018 results

Given the best systems obtained from the above model development process, our submissions use models trained by the ensemble mechanism, and using the 2017 training and 2018 set A data. For the speech task, we submitted results using 1-best and 2-best hypotheses.

The text task uses the ASR output provided by the organizers. Our model performance for the text task is shown in Table 6, for a few different metrics. The first column presents our submission codes [8]. Experiments with no submission codes are conducted after the test results were released. We submitted results using different thresholds, since we noticed that optimizing the threshold for better *D* score is not always the best for other metrics, such as macro F-1 scores. Our submission LLL with the lowest threshold t achieves the highest *D* value, and ranks the best among all the entries of both text and speech tasks. Our submission KKK, using a higher threshold, yields a lower *D* score but higher F-1.

For the speech task, we run our ASR model on both 2017 and 2018 data and use the output to retrain the text classifier. The top-half of Table 7 presents our results for speech task when using 1-best ASR output, and the lower part corresponds to using 2-best lists. Similar to the 2017 test data, we can see that using 2-best ASR hypotheses improves performance. However, some observations are worth discussing. First of all, our submission HHH ($t = 0.4$) returns $D = 13.492$, which is the top

Table 6: Performance in the 2018 text task. *D* scores of our submissions are highlighted.

Code	Model	F-1	D	IREj
KKK	Ensemble, $t = 0.5$	0.791	11.965	0.399
–	Ensemble, $t = 0.4$	0.766	15.369	0.328
LLL	Ensemble, $t = 0.35$	0.755	19.088	0.305

Table 7: Performance in the 2018 speech task. *D* scores of our submissions are highlighted.

Code	Model	F-1	D	IREj
–	$t=0.5$	0.816	11.330	0.453
–	$t=0.4$	0.787	11.792	0.393
III	$t=0.35$	0.772	10.909	0.364
–	$t=0.5$, 2-best	0.803	14.573	0.401
HHH	$t=0.4$, 2-best	0.764	13.492	0.342
–	$t=0.35$, 2-best	0.753	13.237	0.318

among submissions to the speech task, but a higher $D = 14.57$ can be obtained when $t = 0.5$.² This pattern is different from what we found on the 2017 test data. Secondly, using the transcripts provided after the shared task evaluation, we calculated our ASR system’s WER on 2018 test set, which is 8.9%, 10% relatively lower than the WER from the provided ASR results (10.8%). Even with this better ASR performance, we obtained a lower *D* score for the speech task compared to the text task, but the F-1 scores were higher. We performed some analyses and found that this was mostly caused by the effect when changing the decision thresholds. In 2017 data, lowering t helps obtain higher *D* score, but not F-1, for both text and speech conditions. However, that does not hold in 2018 speech task data. In fact, Table 7 shows that $t = 0.35$ returns the lowest F-1 and *D* values in the speech task. This suggests that the optimal value of decision threshold may not transfer well among tasks and/or data, which needs more future studies.

5. Conclusions

In this paper, we presented our system for the 2018 Spoken CALL challenge and reported our submissions to both the text and speech tasks of the challenge. We first improved the text classification by using a large number of syntactic and semantic features, and an ensemble model consisting of diverse classifiers. For the speech task, we further enhanced ASR models for higher recognition accuracy, and proposed to utilize n-best ASR hypotheses to reduce the impact of recognition errors. We achieve the top *D* scores for both text and speech tasks among all the participating teams. While the results in the development data show the benefit using our improved ASR model and n-best lists, our submissions to the speech task do not yield higher *D* scores than our submissions to the text task. This raises a question of model parameter transfer across tasks and data, and requires us to conduct further analyses in the future. Furthermore, we will investigate other features that can better represent n-best ASR hypotheses rather than only using the one that best matches with the reference grammar.

²We did not submit the result with $t = 0.35$ since we observed IREj on the 2017 test set was close to the lower bound 0.25.

6. References

- [1] M. Levy, *Computer-assisted language learning: Context and conceptualization*. Oxford University Press, 1997.
- [2] M. Virvou and V. Tsigra, “Web Passive Voice Tutor: an intelligent computer assisted language learning system over the WWW,” in *Proceedings IEEE International Conference on Advanced Learning Technologies*, 2001, pp. 131–134.
- [3] T. Heift and D. Nicholson, “Theoretical and Practical Considerations for Web-Based Intelligent Language Tutoring Systems,” in *Intelligent Tutoring Systems*, G. Gauthier, C. Frasson, and K. VanLehn, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 354–362.
- [4] B. Penning de Vries, S. Bodnar, C. Cucchiari, H. Strik, and R. v. Hout, “Spoken grammar practice in an ASR-based CALL system,” in *Speech and Language Technology in Education (SLaTE)*, Grenoble, France, 2013, pp. 60–65.
- [5] C. Cucchiari, S. Bodnar, B. Penning de Vries, R. V. Hout, and H. Strik, “ASR-based CALL Systems and Learner Speech Data: New Resources and Opportunities for Research and Development in Second Language Learning.” Reykjavik, Iceland: European Language Resources Association (ELRA), May 2014.
- [6] E. Rayner, N. Tsourakis, C. Baur, P. Bouillon, and J. Gerlach, “CALL-SLT: A Spoken CALL System Based on Grammar and Speech Recognition,” *Linguistic Issues in Language Technology*, vol. 10, no. 2, 2014. [Online]. Available: <https://archive-ouverte.unige.ch/unige:42119>
- [7] C. Baur, C. Chua, J. Gerlach, M. Rayner, M. Russell, H. Strik, and X. Wei, “Overview of the 2017 Spoken CALL Shared Task,” in *Proc. 7th ISCA Workshop on Speech and Language Technology in Education*, 2017, pp. 71–78. [Online]. Available: <http://dx.doi.org/10.21437/SLaTE.2017-13>
- [8] C. Baur, A. Caines, C. Chua, J. Gerlach, M. Qian, M. Rayner, M. Russell, H. Strik, and X. Wei, “Overview of the 2018 Spoken CALL Shared Task,” in *Interspeech 2018*, India, Sep. 2018.
- [9] A. Magooda and D. Litman, “Syntactic and semantic features for human like judgement in spoken CALL,” in *Proc. 7th ISCA Workshop on Speech and Language Technology in Education*, 2017, pp. 109–114. [Online]. Available: <http://dx.doi.org/10.21437/SLaTE.2017-19>
- [10] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, “The Stanford CoreNLP Natural Language Processing Toolkit,” in *Association for Computational Linguistics (ACL) System Demonstrations*, 2014, pp. 55–60. [Online]. Available: <http://www.aclweb.org/anthology/P/P14/P14-5010>
- [11] R. Dale and A. Kilgarriff, “Helping Our Own: The HOO 2011 Pilot Shared Task,” in *Proceedings of the Generation Challenges Session at the 13th European Workshop on Natural Language Generation*. Nancy, France: Association for Computational Linguistics, Sep. 2011, pp. 242–249. [Online]. Available: <http://www.aclweb.org/anthology/W11-2838>
- [12] H. T. Ng, S. M. Wu, T. Briscoe, C. Hadiwinoto, R. H. Susanto, and C. Bryant, “The CoNLL-2014 Shared Task on Grammatical Error Correction,” in *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 1–14. [Online]. Available: <http://www.aclweb.org/anthology/W14-1701>
- [13] X.-H. Phan and C.-T. Nguyen, “GibbsLDA++: A C/C++ implementation of latent Dirichlet allocation (LDA),” Technical report, Tech. Rep., 2007.
- [14] C. Wang, R. Li, and H. Lin, “Deep Context Model for Grammatical Error Correction,” in *Proc. 7th ISCA Workshop on Speech and Language Technology in Education*, Aug. 2017, pp. 167–171.
- [15] K. Evanini, M. Mulholland, E. Tsuprun, and Y. Qian, “Using an Automated Content Scoring Engine for Spoken CALL Responses: The ETS submission for the Spoken CALL Challenge,” in *Proc. 7th ISCA Workshop on Speech and Language Technology in Education*, 2017, pp. 97–102. [Online]. Available: <http://dx.doi.org/10.21437/SLaTE.2017-17>
- [16] Y. R. Oh, H.-B. Jeon, H. J. Song, B. O. Kang, Y.-K. Lee, J.-G. Park, and Y.-K. Lee, “Deep-Learning Based Automatic Spontaneous Speech Assessment in a Data-Driven Approach for the 2017 SLaTE CALL Shared Challenge,” 2017, pp. 103–108. [Online]. Available: <http://dx.doi.org/10.21437/SLaTE.2017-18>
- [17] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16. New York, NY, USA: ACM, 2016, pp. 785–794. [Online]. Available: <http://doi.acm.org/10.1145/2939672.2939785>
- [18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [19] M. Qian, X. Wei, P. Janovi, and M. Russell, “The University of Birmingham 2017 SLaTE CALL Shared Task Systems,” in *Proc. 7th ISCA Workshop on Speech and Language Technology in Education*, 2017, pp. 91–96. [Online]. Available: <http://dx.doi.org/10.21437/SLaTE.2017-16>
- [20] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, and others, “The Kaldi speech recognition toolkit,” in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.