The University of Birmingham 2018 spoken CALL shared task systems

Qian, Mengjie; Wei, Xizi; Jancovic, Peter; Russell, Martin

DOI:
10.21437/Interspeech.2018-1372

License:
Other (please specify with Rights Statement)

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (Harvard):
Qian, M, Wei, X, Jancovic, P & Russell, M 2018, The University of Birmingham 2018 spoken CALL shared task systems. in *Proceedings of Interspeech 2018.* Interspeech, vol. 2018, ISCA, Hyderabad, India, pp. 2374-2378, Interspeech 2018, Hyderabad , India, 2/09/18. https://doi.org/10.21437/Interspeech.2018-1372

Interspeech 2018
2-6 September 2018, Hyderabad

The University of Birmingham 2018 Spoken CALL Shared Task Systems

Mengjie Qian, Xizi Wei, Peter Jančovič, Martin Russell

Department of Electronic, Electrical & Systems Engineering, The University of Birmingham, UK
{mxq486, xxw395, p.jancovic, m.j.russell}@bham.ac.uk

Abstract

This paper describes the systems developed by the University of Birmingham for the 2018 CALL Shared Task (ST) challenge. The task is to perform automatic assessment of grammatical and linguistic aspects of English spoken by German-speaking Swiss teenagers. Our developed systems consist of two components, automatic speech recognition (ASR) and text processing (TP). We explore several ways of building a DNN-HMM ASR system using out-of-domain AMI speech corpus plus a limited amount of ST data. In development experiments on the initial ST data, our final ASR system achieved the word-error-rate (WER) of 12.00%, compared to 14.89% for the official ST baseline DNN-HMM system. The WER of 9.28% was achieved on the test set data. For TP component, we first post-process the ASR output to deal with hesitations and then pass this to a template-based grammar, which we expanded from the provided baseline. We also developed a TP system based on machine learning methods, which enables to better accommodate variability of spoken language. We also fused outputs from several systems using a linear logistic regression. Our best system submitted to the challenge achieved F-measure of 0.914, D of 10.764 and D_{full} score of 5.691 on the final test set.
Index Terms: Spoken CALL Shared Task, speech recognition, text processing, DNN-HMM, rule-based grammar, word2vec

1. Introduction

Shared tasks have been a major factor in the development of many areas of speech and language technology. The first shared task (ST) for Computer Assisted Language Learning (CALL), referred to as "2017 SLaTE CALL Shared Task", was presented in 2017 [1, 2]. This was led by the University of Geneva with support from the University of Birmingham and Radboud University. Following the success of the first edition, the above consortium of universities along with University of Cambridge introduced this year the second edition of the ST [3]. The task is using recordings from the German speaking Swiss teenagers interacting with the CALL-SLT systems [4]. As part of the 2018 ST, a development set of 6698 recordings was released in October 2017. Each recording has a corresponding German prompt, transcription, ASR output from a baseline DNN-HMM recogniser, and human judgments for grammar and semantic correctness. The test set consists of 1000 recordings, each with a German prompt. It was released in February 2018 and research groups had 1 week to submit up to 3 judgment results made by their developed systems.

This paper describes the three systems that we submitted to the 2018 CALL Shared Task and also additional developments we performed after the challenge submission deadline. Each system consists of two components, automatic speech recognition (ASR) and text processing (TP). Our ASR system was developed using the Kaldi toolkit [5] and builds on the best ASR system developed in the 2017 CALL Shared Task challenge [6]. For ASR training, we used a portion of the AMI corpus of unscripted speech [7]. This plus 90% of ST-DEV was used for pre-training and training, followed by a final phase of training using only ST-DEV. The optimum amount of AMI training data (to balance with ST-DEV) and various parameters of the ASR system were determined empirically in development experiments on ST-DEV. For text processing we expanded the baseline grammar to include word sequence patterns from ST-DEV that were judged correct but were missing from the original grammar. In addition, we also developed a text processing system based on machine learning methods.

The rest of the paper is organised as follows. In section 2, we briefly introduce the spoken CALL Shared Task challenge. Sections 3 describes the ASR systems, section 4 and 5 introduce the rule-based and machine learning based text processing systems, respectively. Section 6 presents our final experiments and results and section 7 gives conclusions.

2. Spoken CALL Shared Task Challenge

The Shared Task challenge is based on data collected from a speech-enabled online tool CALL-SLT [8, 4], which has been under development at the University of Geneva since 2009. The system was designed to help young Swiss German teenagers to practise skills in English conversation.

The items of data are prompt-response pairs, where the prompt is a piece of German text and the response is an utterance spoken in English and recorded as an audio file.

The task of the challenge is to label pairs as "accept" or "reject", accepting responses which are grammatically and linguistically correct and rejecting those incorrect either in grammar or meaning according to the judgments of a panel of human listeners and machines [1, 2, 3].

The baseline system for the challenge consists of two components, speech-processing and text-processing. Participants of the challenge could work on one or both of the components. The baseline system for the speech-processing component consisted of a DNN-HMM ASR system which achieved best word-error-rate in 2017 ST challenge [6]. For the text-processing component, a baseline rule-based grammar was provided.

3. Automatic Speech Recognition

3.1. Training and Test Corpus

The training and test data of the 2018 Shared Task (ST2) were released in October 2017 and February 2018, respectively. In addition to this, we also used data released for the 2017 edition of Shared Task (ST1). The number of utterances and total length of each set are given in Table 1. Our ASR systems were developed using the ST12 data. The ST2_train set, after excluding sentences containing only silence, was split into 10 sub-sets and these were used to evaluate our best ASR system from the 2017 ST edition. The ST12_dev set was formed to

10.21437/Interspeech.2018-1372

contain two sub-sets with the best and with the worst WER plus the ST1_test. The remaining sub-sets of the ST2_train and all the data in ST1_train formed the ST12_train set.

Table 1: *Amount of the Shared Task 1 (ST1) and Shared Task 2 (ST2) data.*

Data abbreviation	Num. of utts	Num. of hours
ST1_train	5222	4.80
ST1_test	996	0.89
ST2_train	6698	5.99
ST2_test	1000	0.91
ST12_train	10521	9.50
ST12_dev	1948	1.86

As the amount of ST data is limited, we also used an out-of-domain speech data. We opted for the AMI corpus as it contains conversational speech of native and non-native speakers. The AMI corpus includes 100 hours audio recordings of meetings made with 3 different conditions: recordings of independent headset microphone (IHM), multiple distant microphone (MDM) and single distant microphone (SDM). We explored augmenting the ST training data with IHM and with SDM data. The use of IHM data provided better recognition performance and as such this was used in all of our developed ASR systems.

3.2. Baseline System

Our baseline ASR system is a hybrid deep neural network - hidden Markov model (DNN-HMM) built using Kaldi [5]. First, 39-dimensional Mel-frequency cepstral coefficients (MFCCs) with first- and second-order derivatives were used to train a GMM-HMM triphone model and produce the state level time alignments. Then linear discriminant analysis (LDA) was applied on 91-dimensional vector of MFCCs, containing 13-dimensional static MFCCs with context of 7 frames (i.e., ± 3 frames), to decorrelate and reduce to 40-dimensional features. After further decorrelating using maximum likelihood linear transform, we applied feature space maximum likelihood linear regression (fMLLR) to do speaker adaptation. These fMLLR features and the new alignments were used to train a DNN.

The 40-dimensional fMLLR features were spliced in time taking a context of 11 frames (i.e., ± 5 frames) and used as the input to a neural network with 6 hidden layers and 1024 neurons at each layer. The output was a softmax layer with 3700 units. We varied the amount of out-domain AMI data to obtain a robust model while keeping the characteristics of the in-domain ST data. After the DNN model was trained, we removed the output layer and fine-tuned the parameters only using the in-domain data in order to adapt the model. Two DNN models were trained using the procedure described above either with 20% (16.08 hours) or with 50% (40.64 hours) of the IHM data together with the ST12_train and the models were then fine-tuned using only ST12_train. The DNN models used to produce the transcriptions for ST2_test were trained using the same procedure as described above but ST12_dev data were also included into the training set.

We used a trigram language model (LM) trained on the reference transcriptions of the ST data using the SRILM toolkit [9]. The LM1 denotes model obtained based on the reference transcriptions of ST12_train and used during the ASR development. The LM2 was trained on both the ST12_train and the ST12_dev and used for the final experiments on ST2_test.

Results on ST12_dev and ST2_test are shown in Table 2. Varying the amount of AMI-IHM data to augment the ST data has a small effect on the recognition performance.

Table 2: *Recognition results (%WER) obtained by DNN-HMM system on the development and final test set, when using different amount of AMI-IHM data and language model.*

ASR model: DNN-HMM			Test data	
Data Aug	Train	LM	ST12_dev	ST2_test
IHM20	DEV	LM1	12.68	9.62
IHM50	DEV	LM1	12.64	9.78
IHM20	FINAL	LM2	-	9.84
IHM50	FINAL	LM2	-	10.01

3.3. Developed Systems

3.3.1. Long Short-Term Memory

Long short-term memory (LSTM) has often been shown to perform better than DNNs in large vocabulary speech recognition [10, 11, 12]. Our LSTM networks were trained based on the alignments we obtained from our best DNN-HMM system. We compared 13-dimensional MFCCs and 40-dimensional fMLLR features, both with context of 5 frames (i.e. ± 2 frames). To make use of the information from the future frame, we delayed the output HMM state label by 5 frames. The LSTM network has 1024 memory cells, a hidden layer with 1024 units, a recurrent projection layer with 256 units and a non recurrent projection layer with 256 units. Results, presented in Table 3, show that LSTMs trained on 20% of AMI-IHM perform better than 50% of AMI-IHM for the development model on ST12_dev and also the final model on ST12_test. When comparing the results of the two set of features, inconsistent performance improvements on fMLLR features can be seen for different models.

Table 3: *Recognition results (%WER) obtained by the LSTM model on the development and final test set, when using different amount of AMI-IHM data, features, and language model.*

ASR model: LSTM				Test data	
Data	Feat	Train	LM	ST12_dev	ST2_test
IHM20	MFCC	DEV	LM1	12.79	9.99
IHM20	fMLLR	DEV	LM1	12.11	10.21
IHM50	MFCC	DEV	LM1	12.82	8.65
IHM50	fMLLR	DEV	LM1	13.11	9.76
IHM20	MFCC	FINAL	LM2	-	8.82
IHM20	fMLLR	FINAL	LM2	-	9.60
IHM50	MFCC	FINAL	LM2	-	9.71
IHM50	fMLLR	FINAL	LM2	-	10.15

3.3.2. Sequence Discriminative Training

Systems in section 3.2 and 3.3.1 were trained to model the label posterior probability based on the cross-entropy criterion, which treats each frame independently. However, speech recognition is a sequence classification problem. Some sequence-discriminative training techniques are popularly used in speech recognition, including maximum mutual information (MMI), boosted MMI (BMMI), minimum phone error (MPE) and minimum Bayes risk (MBR) training criteria. We used the state-

level minimum Bayes risk (sMBR) [13] in our experiments. The DNN-HMM system from section 3.2 was used as the base for sequence training, which used 3 iterations and learning rate of 0.00001. For each iteration, the alignments and word lattices were generated by decoding the ST12_train data using the corresponding cross-entropy trained DNN.

Results are shown in Table 4. Comparing the results in Table 3 and Table 4, we can observe that during the development stage the sMBR trained DNNs slightly outperformed the LSTMs. The best WER is 12.00% by the sequence training model using 50% of IHM. For the final model, the best WER of 8.82% was achieved by LSTM trained using 50% of IHM data.

Table 4: *Recognition results (%WER) obtained by the sMBR-trained DNN on the development and final test set, when using different amount of AMI-IHM data and language model.*

ASR model: sMBR trained DNN			Test data	
Data Aug	Train	LM	ST12_dev	ST2_test
IHM20	DEV	LM1	12.28	9.08
IHM50	DEV	LM1	12.00	9.50
IHM20	FINAL	LM2	-	10.56
IHM50	FINAL	LM2	-	9.28

4. Rule-based Text Processing

4.1. Baseline System

The baseline text processing system, provided by the organisers of the challenge, is a rule-based system based on a reference grammar. This grammar includes a set of possible responses for each prompt. If an ASR transcription of a given utterance was in the response list, then this utterance would be labelled as "accept", otherwise, it would be labelled as "reject".

This grammar is generated following a few basic templates and some updating methods [14]. The baseline reference grammar from 2017 ST challenge, which was provided online, was expanded using both the ST1 and ST2 data. The updated grammar contains 557 prompt units and 63469 responses in total.

4.2. Developed Systems

4.2.1. Post-processing the ASR Output

As the CALL-SLT tool was designed to practice English conversation, it seems reasonable to disregard some hesitation, repetitions and modification of the spoken responses. As such, the ASR output was post-processed as described below.

Formulaic expressions: Words like "yes", "hello", "hi", "sorry" often occur in the beginning of the sentences. These words were removed because they are not useful to make judgments neither on grammar nor meaning.

Interjections: Some hesitation words (such as "um", 'ah", "hah") may appear anywhere within the sentences. These words were removed.

Repetitions: There are different kinds of repetitions appearing in the sentences which are caused by hesitations of person speaking. Those repetitions could be words or phrases, e.g., "I have three three tickets", "No I don't have a do not have a reservation". Duplicated words or phrases were deleted.

Half-words: Another case we processed is false-start. Persons may be uncertain about their answer, so they may not give the correct response for the first time, but we should also accept it if they make it correct for the second time. Half words, like "gal" in "I want tickets for the gal gallery", "a" in "I want a an orange juice", "post" in "I would like to pay by post postcard" were removed from the sentences.

4.2.2. Expanding Reference Grammar

We found that the set of responses for some prompts was not sufficiently well covered in the baseline text processing grammar. Thus, we employed the same procedure which we used in the 2017 ST challenge to expand the baseline reference grammar. This is described in detail in [6].

4.3. Fusing Multiple Text Processing Results

In developing our ASR systems, we have trained multiple acoustic models and we observed that the performance of models varies on different test sets. As such, we explored fusion of outputs obtained from multiple systems including DNN-HMM models with 20% or 50% of AMI-IHM, sequence training models and LSTM models each with 20% or 50% of AMI-IHM data. We employed fusion based on linear logistic regression using the FoCal toolkit [15], with further details on this given in [6].

5. Text Processing using Machine Learning

The rule-based text processing system may not be able to accommodate well the variability of spoken language and it provides only a binary decision about the ASR output transcription. To overcome these shortcomings, we explored the use of machine learning techniques for text processing. We took the approach of first computing a similarity measure between ASR output and the responses in the reference grammar and then building a two-class ("accept" and "reject") classifier.

A number of methods have been proposed to calculate a similarity between sentences, e.g., [16]. We employed a method that first converts the words in a sentence into a vector representation (referred to as 'word2vec') and then calculates a distance between two sentences based on these word vectors. As our training data is too small to train a word2vec model, we used Google's pre-trained model [17]. This model contains word vectors for a vocabulary of 3 million words and phrases which are trained on approximately 100 billion words from Google News dataset. The word vector dimension is 300. We then employed the Word Mover's Distance (WMD) [18] algorithm to calculate a sentence-level distance between the ASR output and each correct response from the reference grammar for the given prompt. The WMD algorithm finds the minimum distance that the word vectors of one document need to "travel" to reach the word vectors of another document. This was performed using the gensim software package [19].

The distances from N best matching responses were used to construct a feature vector that characterises the ASR output. The feature vector was filled with the average distance in a case the number of possible responses in the grammar was lower than N. We also explored transforming this N-dimensional sentence similarity feature vector to a lower-dimensional representation using the Principal Component Analysis.

We employed several different classifiers to obtain the decision about an ASR output transcription based on the sentence-level N-dimensional vector. These include: linear discriminant analysis (LDA), logistic regression, support vector machine (SVM), and neural network (NN). In the NN-based text processing, a network with one hidden layer with 16 neurons, tanh as the activation function and the limited-memory Broyden-

Fletcher-Goldfarb-Shanno (L-BFGS) training algorithm [20], which is more numerically stable than stochastic gradient descent (SGD), were used.

6. Experimental Results and Discussion

6.1. Scoring Metrics

The sentences are annotated by native speakers according to linguistic correctness and meaning. Comparing the system's judgments with the human language and meaning annotations, the result for each response falls into one of the following categories: i) Correct Accept (CA) – sentence that is labelled as correct both in language and meaning is accepted by the system; ii) False Reject (FR) – sentence that is correct linguistically and semantically is rejected; iii) Correct Reject (CR) – sentence that is incorrect either in language or in meaning is rejected; iv) False Accept (FA) – an incorrect sentence is accepted. The FAs are split into "Plain FAs" (PFAs) and "Gross FAs" (GFAs), corresponding to an FA of a response that is incorrect in language but has correct meaning and that is incorrect in both linguistic and semantic sense, respectively. In calculating the overall FA, the GFA are given k times heavier weight than PFA. The FA is calculated as $FA = PFA + k \times GFA$, with $k = 3$.

The challenge used originally the following metrics: F-measure, scoring accuracy (SA), and differential response (D) score. The F-measure is defined as $F = \frac{2PR}{(P+R)}$, where P and R denotes the precision and recall, respectively, being defined as $P = \frac{CA}{(CA+FA)}$ and $R = \frac{CA}{(CA+FR)}$. The SA is defined as $SA = \frac{(CA+CR)}{(CA+CR+FA+FR)}$. The D-score is defined as the ratio of the rejection rate on the incorrect responses to the rejection rate on the correct responses – this can be expressed as $D = \frac{CR(FR+CA)}{FR(CR+FA)}$. After the challenge deadline, D_a and D_{full} metrics were added. The D_a is defined similarly as D but with concern on acceptance rate, i.e., $D_a = \frac{CA(CR+FA)}{FA(FR+CA)}$. The D_{full} is the geometric average of D and D_a, i.e., $D_{full} = \sqrt{DD_a}$.

6.2. Results of Official Submissions

This section presents results on ST-TST data obtained by the three systems we submitted by the deadline of the challenge. All these systems employed the expanded rule-based text processing as described in section 4.

Submission 1 (system DDD on the official 2018 SLaTE CALL Shared Task results table [21]) consisted of our best single ASR system – sequence training model with 50% of IHM data. This submission achieved the F-measure of 0.915, D-score of 10.714, and D_{full} score of 5.778.

Submission 2 (EEE) was the result obtained by the final DNN-HMM model trained with 50% of IHM data. This submission achieved the F-measure of 0.904, D-score of 8.804, and D_{full} score of 4.958.

Submission 3 (FFF) was the result obtained by fusing the outputs of six separate systems using linear logistic regression. The individual systems were six variants of the ASR including DNN-HMM, sequence training model and LSTM model each with 20% or 50% of IHM data. This submission achieved the F-measure of 0.914, D of 10.764, and D_{full} score of 5.691.

6.3. Results using ML-based Text Processing System

Results obtained using different classifiers in our ML-based text processing system are presented in Table 5. The presented results are with N set to 10 (but they did not vary largely for different values of N). It can be seen that the NN-based system performed better than other classifiers. The achieved performance is considerably better in all evaluation measures (F-measure, D and D_{full} score) than our submitted systems using the rule-based text processing.

Table 5: *Results obtained by machine-learning text processing systems employing different classifiers (N was set to 10).*

Classifier	Evaluation measure		
	F-measure	D	D_{full}
LDA	0.88	9.767	4.136
logReg (PCA)	0.884	10.263	4.281
SVM (PCA)	0.891	10.939	4.616
NN	0.928	12.716	7.101

We have also explored the effect of varying the threshold for making the final judgement decision. This was 0.5 in all previous experiments. Results, as a function of the threshold value, obtained using the NN-based TP system are depicted in Figure 1. It can be seen that a higher D-score (12.900) can be achieved when the threshold is set to 0.250, while the D_{full} measure (5.340) is lower than 7.101. Besides the system is not stable when the threshold is around 0.25.

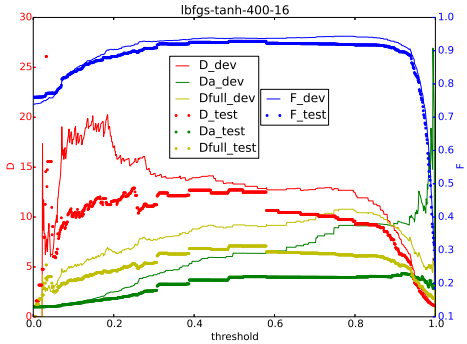

Figure 1: *Results in terms of various evaluation measures for the NN-based TP system when varying the decision threshold.*

7. Conclusions

This paper described the University of Birmingham's submissions to the 2018 CALL Shared Task challenge. Our systems comprised of an ASR and text processing (TP) component. Our initial focus was on ASR. We extended the DNN-HMM system which achieved the best performance in 2017 CALL Shared Task challenge by using LSTM and sequence training. Our best developed ASR system, developed with Kaldi using the AMI and Shared Task corpora, achieved WER of 12.00% and 9.89% on the ST-DEV and ST-TST, respectively. Our best submission to the challenge, employing an expanded version of the rule-based TP, obtained the F-measure of 0.914, D-score of 10.764 and D_{full} score of 5.691. After the challenge deadline, we developed a machine-learning (ML) -based TP. This used word2vec representation and Word Mover's Distance to obtain a similarity measure to reference grammar responses. In our future work, we plan to incorporate 'doc2vec' similarity measure instead of using 'word2doc' representation and use directly the D_{full} measure as the optimisation criteria in ML-based TP.

8. References

[1] C. Baur, J. Gerlach, E. Rayner, M. Russell, and H. Strik, "A shared task for spoken CALL?" in *Proc. Language Resources and Evaluation Conf. (LREC)*, 2016.

[2] C. Baur, C. Chua, J. Gerlach, M. Rayner, M. Russel, H. Strik, and X. Wei, "Overview of the 2017 spoken CALL shared task," in *Proc. of SLaTE Workshop, Stockholm, Sweden*, 2017.

[3] C. Baur, A. Caines, C. Chua, J. Gerlach, M. Qian, M. Rayner, M. Russel, H. Strik, and X. Wei, "Overview of the 2018 spoken CALL shared task," in *Proc. of Interspeech, Hyderabad, India (accepted)*, 2018.

[4] C. Baur, "The potential of interactive speech-enabled call in the swiss education system: A large-scale experiment on the basis of english CALL-SLT," Ph.D. dissertation, Université de Genève, 2015.

[5] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, and P. Schwarz, "The kaldi speech recognition toolkit," *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011.

[6] M. Qian, X. Wei, P. Jančovič, and M. Russell, "The University of Birmingham 2017 SLaTE CALL shared task systems," in *Proc. of SLaTE Workshop, Stockholm, Sweden*, 2017.

[7] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal *et al.*, "The AMI meeting corpus: A pre-announcement," in *Int. Workshop on Machine Learning for Multimodal Interaction*. Springer, 2005, pp. 28–39.

[8] E. Rayner, P. Bouillon, N. Tsourakis, J. Gerlach, Y. Nakao, and C. Baur, "A multilingual CALL game based on speech translation," in *Proc. Language Resources and Evaluation Conf. (LREC),* Valetta, Malta, 2010.

[9] A. Stolcke, J. Zheng, W. Wang, and V. Abrash, "Srilm at sixteen: Update and outlook," in *Proceedings of IEEE automatic speech recognition and understanding workshop*, vol. 5, 2011.

[10] F. B. H. Sak, A. Senior, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Interspeech*, 2014.

[11] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *CoRR*, vol. abs/1402.1128, 2014. [Online]. Available: http://arxiv.org/abs/1402.1128

[12] X. Li and X. Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," *CoRR*, vol. abs/1410.4281, 2014. [Online]. Available: http://arxiv.org/abs/1410.4281

[13] M. Gibson and T. Hain, "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition," in *Interspeech*, 2006.

[14] E. Rayner, C. Baur, C. Chua, and N. Tsourakis, "Supervised learning of response grammars in a spoken CALL system," in *Proc. Workshop on Speech and Language Technology in Education (SLaTE)*, 2015.

[15] N. Brümmer, "Focal multi-class: Toolkit for evaluation, fusion and calibration of multi-class recognition scorestutorial and user manual," *Software available at http://sites. google. com/site/nikobrummer/focalmulticlass*, 2007.

[16] P. Achananuparp, X. Hu, and X. Shen, "The evaluation of sentence similarity measures," in *International Conference on data warehousing and knowledge discovery*. Springer, 2008, pp. 305–316.

[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[18] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *Int. Conf. on Machine Learning*, 2015, pp. 957–966.

[19] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proc. of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

[20] R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer, "A stochastic quasi-newton method for large-scale optimization," *SIAM Journal on Optimization*, vol. 26, no. 2, pp. 1008–1031, 2016.

[21] "Spoken CALL Shared Task - Second Edition."