SLaTE 2019: 8th ISCA Workshop on Speech and Language Technology in Education
20-21 September 2019, Graz, Austria

The University of Birmingham 2019 Spoken CALL Shared Task Systems: Exploring the importance of word order in text processing

Mengjie Qian[1], *Peter Jančovič*[1], *Martin Russell*[2]

[1]Department of Electronic, Electrical & Systems Engineering, The University of Birmingham, UK
[2]School of Computer Science, The University of Birmingham, UK

{mxq486, p.jancovic, m.j.russell}@bham.ac.uk

Abstract

This paper describes the systems developed by the University of Birmingham for the 2019 Spoken CALL Shared Task (ST) challenge. The task is automatic assessment of grammatical and semantic aspects of English spoken by German-speaking Swiss teenagers. Our system has two main components: automatic speech recognition (ASR) and text processing (TP). We use the ASR system that we developed for 2018 ST challenge. This is a DNN-HMM system based on sequence training with the state-level minimal Bayes risk criteria. It achieved word-error-rates (WER) of 8.89% for the ST2 test set and 10.94% for the ST3 test set. This paper focuses on development of the TP component. In particular, we explore machine learning (ML) approaches which preserve different degrees of word order. The ST responses are represented as vectors using Word2Vec and Doc2Vec models and the similarities between ASR transcriptions and reference responses are calculated using Word Mover's Distance (WMD) and Dynamic Programming (DP). A baseline rule-based TP system obtained a D_{full} score of 5.639 and 5.476 for the ST2 and ST3 test set, respectively. The best ML-based TP, consisting of a Word2Vec model trained on the ST data, DP-based similarity calculation and a neural network, achieved D_{full} score of 7.379 and 5.740 for ST2 and ST3 test sets, respectively.

Index Terms: Spoken CALL Shared Task, speech recognition, text processing, word embedding, Word Mover's Distance, dynamic programming.

1. Introduction

Shared tasks have been a major factor in the development of many areas of speech and language technology. The Spoken CALL Shared Task (ST) is a series of open challenges jointly organised by the University of Geneva, the University of Birmingham, the Radboud University and the University of Cambridge. The task is to provide feedbacks to prompt-based responses spoken by English learners using the CALL-SLT systems [1, 2]. The first ST, which had 20 submission entries from 9 groups, was presented in the ISCA SLaTE 2017 workshop [3, 4] in Stockholm. The second ST [5] with improved training data and improved baseline recogniser resources, was carried out in 2018 and presented as a special session of Interspeech 2018 in Hyderabad. Following the success of the first two editions, the organisers introduced the third edition of the ST in 2019. Participating groups are allowed to use the data from the 2017 ST and 2018 ST, referred to as "ST1" and "ST2". There were 5222 and 966 recordings released as the development set and the test set in ST1, 6698 and 1000 recordings released in ST2 as development set and test set, respectively. Each recording has a corresponding German prompt, transcription, ASR output from a baseline DNN-HMM recogniser, and human judgments for grammar and semantic correctness. In 2019 ST, referred to as "ST3", a new test set consisting of 1000 recordings, each with a German prompt, were released one week before the submission.

This paper describes the systems that we developed for the 2019 CALL Shared Task. Each system consists of two components, automatic speech recognition (ASR) and text processing (TP). We used the ASR system that we developed using the Kaldi toolkit [6] for 2018 CALL Shared Task [7]. For text processing we improved several aspects of our machine learning based TP system from ST2 [7] from several aspects. We compared the different effects of word embeddings extracted from various Word2Vec or Doc2Vec models [8, 9], explored the importance of word order in sentence similarity calculation, and discussed the impact of ASR transcriptions to the TP performance.

The rest of the paper is organised as follows. In section 2, we briefly introduce the Spoken CALL Shared Task challenge and the metric used to evaluate the systems. Section 3 describes the baseline system, section 4 introduces the improved machine learning based text processing systems. Section 5 presents our experiments and results and section 6 gives conclusions.

2. Spoken CALL Shared Task Challenge

2.1. Introduction to the Shared Task

The Shared Task challenge is based on data collected from a speech-enabled online tool CALL-SLT [2, 10], 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 semantically correct and rejecting those incorrect either in grammar or meaning according to the judgments of a panel of human listeners and machines [3, 4, 5].

The baseline system for the challenge consists of two components, speech-processing and text-processing. Participants in 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 [11]. For the text-processing component, a baseline rule-based grammar was provided.

2.2. Scoring Metric

The sentences are annotated by native speakers according to linguistic correctness and meaning. Comparing the system's judg-

10.21437/SLaTE.2019-3

ments 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 the 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 $k = 3$.

Originally the challenge used the following metrics: D score, D_A score and a D_{full} score. 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 $D = \frac{CR(FR+CA)}{FR(CR+FA)}$. The D_a is defined similarly as D but focuses more 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., $D_{full} = \sqrt{DD_a}$, this is the metric used to rank the submissions in the 2019 challenge.

3. Baseline System

The baseline system consists of an automatic speech recognition (ASR) component and a text processing (TP) component, both of them are the same as our 2018 system [7]. The acoustic model of the speech recogniser used the sequence discriminative training with the state-level minimum Bayes risk criteria [12, 13, 14] trained on the 2017 [4] and 2018 [5] Shared Task data (ST1 and ST2) and the AMI corpus [15]. The language model (LM) is a tri-gram model trained on the ST1 and ST2 training transcriptions using the SRILM toolkit [16]. This ASR model obtained a word-error-rate (WER) of 8.89% for the ST2test and 10.94% for the ST3test.

	ST2test		ST3test	
	(a) orig	(b) post	(a) orig	(b) post
INCORRECT	250	250	260	260
CorrectReject	194	191	210	209
GrossFalseAccept	8	10	13	14
PlainFalseAccept	48	49	37	37
RejectionRate	0.729	0.707	0.734	0.726
CORRECT	750	750	740	740
CorrectAccept	688	697	669	680
FalseReject	62	53	71	60
RejectionRate	0.083	0.071	0.096	0.081
D	8.822	10.01	7.653	8.950
D_a	3.389	3.176	3.402	3.350
D_{full}	5.468	5.639	5.103	5.476

Table 1: *Results obtained by rule-based TP system for ST2test and ST3test set with the original and post-processed ASR transcriptions.*

The baseline TP system is a rule-based system with a reference grammar [7], which includes a set of possible responses for each prompt. The ASR transcription of a given utterance will be labelled as "accept" if it's in the grammar, or "reject" if it is not in the grammar. We used this system to classify the ST2test and ST3test set with their ASR transcriptions. The same post-processing as we did in [7] has been applied on the original ASR outputs. The text processing results on the original and post-processed transcriptions are shown in Table 1. Post processing helped to increase the D_{full} score, it brings more "accept"s for both ST2test and ST3test. To be specific, the number of "CorrectAccept" increases considerably, while the number of "GrossFalseAccept" and "PlainFalseAccept" only have a minor increase.

To explore the influence of the ASR system, we applied the baseline TP on both the first best and second best ASR transcription. For the ST2test, the 1^{st}-best and 2^{nd}-best ASR transcriptions have a WER of 8.89% and 9.04%, respectively. The best two ASR transcriptions for ST3test have a WER of 10.94% and 10.96%, respective. The D, D_a and D_{full} scores for ST2test and ST3test are shown in Table 2. The second best ASR output of ST2test decreases the D_{full} score by 4.4% relatively. However, the second best ASR output of ST3test increases the D_{full} score by 2.2% relatively. The WER of the best two ASR outputs for ST3test are very close, while the 1^{st} best has more deletion errors and the 2^{nd} best has more insertion and substitution errors. This might imply that deletion errors have more effect on the text processing performance.

Score	ST2test		ST3test	
	1^{st}-best	2^{nd}-best	1^{st}-best	2^{nd}-best
D	10.010	9.619	8.950	8.950
D_a	3.176	3.019	3.350	3.499
D_{full}	5.639	5.389	5.476	5.596

Table 2: *Results obtained by rule-based text processing system for ST2test and ST3test set with first and second best ASR transcriptions.*

4. Text Processing using Machine Learning

4.1. System Structure

The baseline TP system is making decisions based on a 1-best match, the quality of the judgements highly depends on the coverage of the reference grammar. To make use of the multiple responses in the reference grammar and make decision based on more information, we developed a machine learning based TP system, the structure of the system is shown in Figure 1. It first converts the ASR transcriptions and the reference responses into vector representations with a Word2Vec or Doc2Vec model [8, 9]. Then these vectors are used to calculate the sentence similarities between the ASR transcription and the reference responses. The similarities are taken as feature representations for the utterances given their specific prompts. As the number of reference responses are different for each prompt, the number of similarities varies among the prompts. We selected top K similarities (lowest K distances) as the features. A 2-class classification can be applied to these features with any conventional classification approach, e.g., Logistic Regression (LR), Support Vector Machine (SVM), Nearest Neighbor and Neural Networks. In our experiments, we use a neural network as the classifier.

4.2. Vector Representation

Word embedding is a vector representation of document vocabularies. It is capable of capturing the context of a word in a doc-

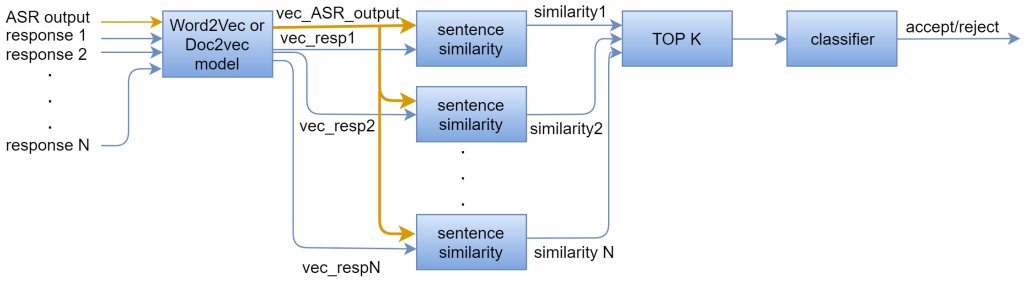

Figure 1: *Structure of the machine learning based text processing system for ST.*

ument, semantic and syntactic similarity, the relation with other words, etc. Word2Vec [8] is one popular technique to learn word embeddings using a shallow neural network. It can be obtained using a skip-gram or continuous bag-of-words (CBOW) algorithm. The pre-trained Google News[1] Word2Vec model was used in our 2018 system [7]. It contains 300-dimensional word vectors for a vocabulary of 3 million words and phrases which are trained on approximately 100 billion words from Google News dataset. In the new system, we compared the GoogleNews model with a Word2Vec model trained on the ST data. This ST Word2Vec model includes a vocabulary of 1366 words, each has a vector of 100 dimensions. It is trained using the CBOW algorithm with the responses in the reference grammar, transcriptions of ST1 and ST2 training set, and recognition transcription for the ST2 test set.

Furthermore, we compared the above two Word2Vec models with a Doc2Vec model. The goal of Doc2Vec is to create numerical representation of a document. There are two algorithms to train the model, distributed memory (DM) and distributed bag-of-words (DBOW), as proposed in [9]. Our Doc2Vec model is trained with ST1 and ST2 training set using DBOW algorithm with 100-dimensional word vectors.

4.3. Sentence Similarity

Word Mover's Distance [17, 18, 19], a distance function between two documents (sentences), measures the minimum traveling distance from the embedded words of one sentence to another one without considering the word orders. We used WMD in the 2018 system [7], but we would like to explore whether the order is important in sentence similarity calculation, especially for short sentences like the ST data. Hence, we compared WMD with a dynamic programming (DP) distance [20], which takes into consideration of the word orders when calculating the sentence similarity. Each word in the ASR and reference transcriptions is represented as a word vector and the distance between each ASR and reference word vector is calculated with the Euclidean distance. DP is used to find the alignment between the ASR and reference transcriptions that minimizes the accumulated distance. We explored different ways of normalising the resulting DP distance and these are detailed in Section 5.

5. Experiments and Results

5.1. Comparing Word Embeddings

A GoogleNews model, the ST Word2Vec and ST Doc2Vec models were used to obtain the word embeddings for the ST data. We fixed the similarity algorithm (WMD) and the number of features (K=10) to explore the influence of different word embeddings on the system. For each vector model, neural networks with different numbers of layers and numbers of neurons per layer were trained and a threshold is used to decide whether to accept or reject the utterance. Apart from using 0.5 as the threshold, we also tried optimizing the threshold. For each vector model, the D_{full} scores for ST2test and ST3test with the best threshold or with 0.5 as the threshold are shown in Table 3, the threshold and the neural network structure are tuned based on ST2test.

Model	optimize th		th=0.5	
	ST2test	ST3test	ST2test	ST3test
GoogleNews	6.073	4.925	5.562	5.236
ST_word2vec	6.894	5.221	5.697	5.461
ST_doc2vec	5.705	5.740	5.639	5.526

Table 3: *D_{full} score for ST2test and ST3test with different vector models and different threshold.*

It seems that vector models trained with ST data outperform the GoogleNews model. This might be related to the fact that the ST data has a very small vocabulary size and its context scenarios are limited. The word embeddings for the ST vocabulary have been trained well in a small Word2Vec or Doc2Vec model, while a huge generic model trained on general external data may not represent the ST vocabulary very well.

5.2. Comparing Distance Algorithms

We explored the importance of word orders in distance calculation by comparing the WMD distances and the distances obtained with Dynamic Programming (DP). The results for ST2test and ST3test are shown in Table 4. Apart from the accumulated DP distances (dp0), we tested three other DP distances (dp1 $\sim$ dp3) obtained using different normalisation. The DP distance for a long sentence may be bigger on average than that for a short sentence, as the DP distance is heavily influenced by the length of the path chosen by the algorithm, and a longer sentence usually needs a longer path to move to the end point on the distance lattice than a shorter one. Hence, we divided the DP distance by the length of the chosen path. This is the dp1 distance in the result Table 4. For each utterance, comparing the ASR transcription of the utterance with the reference responses results in a set of DP distances. The number of the distances and the variance of the distances depend on the number and the variance of the responses for this prompt. The variance normalised distance (dp2) equals to dp0 divided by the standard deviation of the set of the distances for the given prompt. The

[1] https://code.google.com/archive/p/word2vec/

Model	Distance	st2Test			st3Test		
		D	D_A	D_{full}	D	D_A	D_{full}
GoogleNews	wmd	8.974	4.110	6.073	7.655	3.168	4.925
	dp0	12.199	3.727	6.743	9.114	3.098	5.314
	dp1	10.367	4.096	6.516	7.768	3.173	4.965
	dp2	11.979	4.011	6.932	8.909	3.177	5.320
	dp3	11.004	3.926	6.573	8.293	3.231	5.176
ST_word2vec	wmd	10.478	4.536	6.894	8.244	3.307	5.221
	dp0	11.571	4.706	**7.379**	9.239	3.341	5.556
	dp1	11.485	4.569	7.244	9.239	3.341	5.556
	dp2	10.957	4.396	6.940	8.643	3.321	5.358
	dp3	11.691	4.231	7.033	9.082	3.336	5.504
ST_doc2vec	wmd	10.349	3.146	5.705	9.764	3.375	**5.740**

Table 4: *ST2TEST-Results for ST2test and ST3test obtained by the machine learning based system with different word embeddings and distance algorithms.*

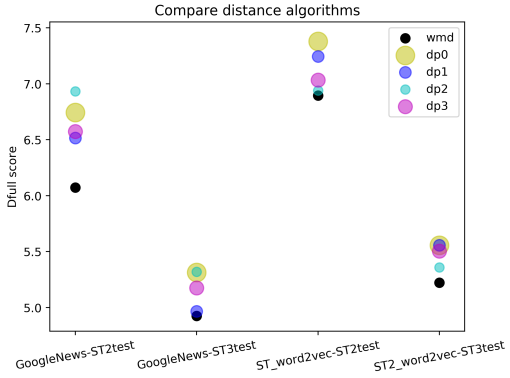

Figure 2: *Comparison of different distance algorithms.*

dp3 distance in Table 4 is the DP distance with both length normalisation and variance normalisation.

A 10-dimensional distance feature from each vector model has been input to a neural network, the hidden layers of the neural network and the threshold for the output layer have been tuned based on ST2test. The results in Table 4 are from the best network for each setup. The best D_{full} score is 7.379 for ST2test and 5.740 for ST3test, both are obtained with the DP distances extracted from the ST Word2Vec model. The comparison between the distance algorithms with different dataset is shown in Figure 2. For different data, the best distance algorithm is always the DP distance, although it's not always the same DP distance. All different DP distances outperform WMD for both ST2test and ST3test.

5.3. Comparing ASR transcriptions

It has been shown to be beneficial to use multiple ASR hypothesises as an n-best list or lattice in many spoken language processing tasks. In the 2018 Shared Task [5], the system developed by Liulisho [21] shows that using 2-best ASR hypothesis outperforms 1-best ASR hypothesis. They compute the edit-distance between each hypothesis and each sample response from the reference grammar, then use the ASR hypothesis with the smallest edit-distance as the input of the text classifier. In our system, we leverage the 2-best ASR transcription in a different way. In the experiments discussed above, the input of our text classifier is a 10-dimensional distance feature calculated between the best ASR hypothesis and the reference responses. We obtained a 5-dimensional feature with each of the 2-best ASR hypothesis, then 10-dimensional concatenated features are the input of our text classifier.

ASR	ST2test	ST3test
1-best	7.379	5.740
2-best	7.089	5.727

Table 5: *The best D_{full} score for ST2test and ST3test obtained by using the best and two best ASR transcriptions.*

Similar systems have been developed with these 2-best hypothesis features as we did for the 1-best hypothesis features. All the results with different word embedding models and different sentence similarity algorithms show that the 2-best system is worse than 1-best system. The best D_{full} scores for ST2test and ST3test from 1-best and 2-best systems are shown in Table 5. This suggests that the advantage of using the n-best hypotheses, in terms of mitigating the effects of ASR errors, can also be achieved using vector representations of words and similarity calculations.

6. Conclusions

In this paper, we described the University of Birmingham systems for the 2019 Spoken CALL Shared Task (ST) challenge. Our systems comprised an ASR and text processing (TP) component. We extended our work from the 2018 CALL ST and focused on the TP component. We explored the influence of different word embeddings from a Word2Vec or Doc2Vec model to represent the text. Small Word2Vec or Doc2Vec models trained with the ST data outperform a big generally trained model. Order insensitive Word Mover's Distance and order sensitive Dynamic Programming (DP) are used to calculate the distance between the ASR transcriptions and the responses in the reference grammar. DP distance has a better performance in most of the experiments, showing that the order of the words in sentence similarity calculation does have some importance in the Shared Task. The employment of n-best ASR transcriptions, with n set to 2, did not provide any advantage. In our future work, we plan to improve the language model in the ASR and embed the prompt information into the text classification.

7. References

[1] M. Rayner, N. Tsourakis, C. Baur, P. Bouillon, and J. Gerlach, "Call-slt: A spoken call system based on grammar and speech recognition," *LiLT (Linguistic Issues in Language Technology)*, vol. 10, 2012.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] M. Qian, X. Wei, P. Jančovič, and M. Russell, "The University of Birmingham 2018 spoken CALL shared task systems," in *Proc. of Interspeech, Hyderabad, India*, 2018.

[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[9] 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.

[10] 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.

[11] 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.

[12] J. Kaiser, B. Horvat, and Z. Kacic, "A novel loss function for the overall risk criterion based discriminative training of hmm models," in *Sixth International Conference on Spoken Language Processing*, 2000.

[13] M. Gibson and T. Hain, "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition," in *Ninth international conference on spoken language processing*, 2006.

[14] D. Povey and B. Kingsbury, "Evaluation of proposed modifications to mpe for large scale discriminative training," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 4. IEEE, 2007, pp. IV–321.

[15] 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.

[16] 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.

[17] O. Pele and M. Werman, "A linear time histogram metric for improved sift matching," in *European conference on computer vision*. Springer, 2008, pp. 495–508.

[18] O. Pele and M. Werman, "Fast and robust earth mover's distances," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 460–467.

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

[20] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.

[21] H. Nguyen, L. Chen, R. Prieto, C. Wang, and Y. Liu, "Liulishuos system for the spoken call shared task 2018," *Proc. Interspeech 2018*, pp. 2364–2368, 2018.