Off-Topic Detection in Automated Speech Assessment Applications

Jian Cheng, Jianqiang Shen

Knowledge Technologies, Pearson
299 S. California Ave, Palo Alto, California 94306, USA
jian.cheng@pearson.com, jianqiang.shen@pearson.com

Abstract

Automated L2 speech assessment applications need some mechanism for validating the relevance of user responses before providing scores. In this paper, we discuss a method for off-topic detection in an automated speech assessment application: a high-stakes English test (PTE Academic). Different from traditional topic detection techniques that use characteristics of text alone, our method mainly focused on using the features derived from speech confidence scores. We also enhanced our off-topic detection model by incorporating other features derived from acoustic likelihood, language model likelihood, and garbage modeling. The final combination model significantly outperformed classification from any individual feature. When fixing the false rejection rate at 5% in our test set, we achieved a false acceptance rate of 9.8%. a very promising result.

Index Terms: off-topic detection, confidence, speech assessment

1. Introduction

With traditional automated speech tutoring applications, it is assumed that the test takers will respond in good-faith, but with automated L2 speech assessment applications, the possibility of cheating must be taken into account. A test taker may say something irrelevant in an attempt to take advantage of its automated scoring. Regardless of what was spoken to an automatic speech recognition (ASR) system, the utterance will be recognized as something that can be accepted by the language model and the acoustic model and may be scored erroneously. To provide reliable scores that reflect the test takers' true abilities and discourage test takers from "gaming the system", it is necessary to implement some mechanisms to validate responses before providing scores. Once detected, off-topic responses may be assigned low scores or sent to human raters. This process is important for face validity of a high-stakes assessment.

There has been a considerable research on text classification. A typical text classification system adopts a bag-of-words approach and treats each document as an unordered collection of words, disregarding word order [1]. The trained classifier makes predictions based on the frequency or appearance of words. Unlike traditional bag-of-words metrics, Latent Semantic Analysis (LSA) [2] projects documents onto a latent semantic subspace and makes predictions based on features in the semantic subspace. Similar to LSA, Probabilistic Latent Semantic Analysis (PLSA) [3] and Latent Dirichlet Allocation (LDA) [4] try to generate topic models by analyzing word document co-occurrences. Lane et al. [5] made use of classification confidence scores of multiple topics derived from above technologies on N-best recognition hypothesis and applied a linear discriminant verifier to detect out-of-domain utterances in the context of spoken dialog system. Higgins et al. [6] used the similarity of vocabulary items between new essays and sample on-topic essays (or prompts) to identify those that were off-topic.

The goal of our research was to reject off-topic utterances for automated speech assessment applications. The definition of off-topic utterances is any response with irrelevant material and/or an excessive amount of mispronounced or unintelligible words as judged by a human listener. The data used in this paper came from a high-stakes English test (Pearson Test of English Academic [7]) in a real assessment environment. The subjects were randomly selected from locations throughout the world.

Pearson Test of English Academic (PTE Academic) [7] delivers real-life measures of test takers' English language ability to universities, higher education institutions, government departments and professional associations requiring academic-level English. It uses automated speech scoring [8, 9] to measure the test takers' speaking skill (pronunciation, fluency, content). The test has several different sections. In "Read aloud", test takers are required to read aloud several passages that appear on the computer screen. In "Describe image", test takers see an image on the screen and are asked to describe the image in detail. In "Re-tell lecture", test takers listen to or watch a lecture and are required to re-tell the lecture in their own words. The recording time for each response in these three tasks was about 40 seconds. On average, each response consisted of 75 words. For this paper, we focused on off-topic detection in the "Describe image" and "Re-tell lecture" tasks. The results can also be applied to "Read aloud". To improve the recognition performance for non-native utterances, the language models used for these tasks were item-specific. In other words, every test item had a dedicated language model, which only covered relevant topics. It is different from the generalized language model used in [5]. Traditional text classification methods will lose their discriminative power in such case.

The method used here mainly focused on the features derived from speech confidence scores. We also explored features derived from acoustic likelihood, language model likelihood, and garbage modeling. Although there has been extensive work on utterance verification [10, 11, 12], it mainly focused on the verification of keywords, short responses, predefined lists of possible (correct and incorrect) responses [11], or predesigned sentences [12]. Since the focus of this paper is topic relevance, we did not attempt to determine whether or not the recognition results were correct. The goal was to detect responses that were off-topic regardless of the recognition results.

2. Off-Topic Measurements

For our off-topic detection model, we designed different measurements based on recognition confidence and language models.

2.1. Confidence Scores at the Word Level

The rationale for focusing on confidence scores is that the ASR engine will generate a significant number of recognition errors when processing off-topic utterances. We used three different methods to generate confidence score variants at the word level:

1. As the basis for normalizing every acoustic score in a given signal frame, we used the maximum acoustic score of a set of alternative hypotheses derived from active tokens. We summed up the acoustic score difference in log for every word and then divided by the number of frames of the word to calculate the average. We treated this average as the confidence score for the word. We used an exponent function to convert it back to the range of zero to one. We denoted this value as $mconf$.

2. For the second confidence score variant, we calculated the difference of the acoustic likelihood at the phoneme level between two different language models. For each utterance, we used its machine transcription and alignment from normal recognition, and derived acoustic likelihoods for the recognized words and phonemes. Then we performed constrained all phone recognition using the same set of acoustic models for the same utterance. In the original recognition, the phonemes were constrained by the original language model. When we performed recognition with the constrained all phones, only time intervals were fixed from the previous recognition. The best one was chosen from all phones. To decrease computation time, we selected from monophones in the second step since we assume that monophones are representative of all phones. We reasoned that the acoustic likelihood from normal recognition would be similar to the likelihood of a single monophone if the recognition result was indeed mapping onto an utterance with the relevant phones. Based on this idea, the confidence score we used was the difference between the best acoustic likelihood using constrained all phone recognition and normal recognition, averaged by the number of frames. We used a log-sigmoid function to convert the score back to the range of zero to one. We denoted this value as $aconf$.

3. We also developed a lattice-based confidence score using an approach similar to the one reported by Mangu et al. [13]. It took word lattices as input and turned them into confusion networks. The word-level posterior probabilities included in the confusion network served as the word confidence scores. We implemented a simplified algorithm to suit our data structure, which ran in less time than the original algorithm but that also built different confusion networks. The performance difference between the two was small. The final confidence score for a word was already in the range of zero to one. We denoted this value as $lconf$.

These three methods were considered as different kinds of approximations to the word level posterior probabilities after giving the acoustic observation X that is recognized as W:

$$P(W|X) = \frac{P(X|W)P(W)}{\sum_H P(X|H)P(H)}, \quad (1)$$

where H denotes a hypothesis for X. In a real-world ASR system, it is extremely difficult to estimate $\sum_H P(X|H)P(H)$ in a precise manner. So approximations are used, such as using maximum to approximate summation.

The computations for $mconf$ and $aconf$ were very simple since we were only concerned with the best path. Compared with speech decoding, the extra computation was trivial. The computation for $lconf$ was heavy because of the overhead needed to maintain a significant number of active tokens per state. In our implementation, it took almost the same amount of time as the regular best path speech decoding. If only the best path is needed for speech decoding, the computational load could become a factor when considering its application in a production system.

2.2. Prediction Features

The confidence scores computed in the previous subsection were at the word level. To detect off-topic responses, we needed to generate features at the response level. To do this, we used the mean, the standard deviation, and the maximum and minimum confidences. We denoted these values as $(x)conf^{\mu}$, $(x)conf^{\sigma}$, $(x)conf^{max}$, and $(x)conf^{min}$ respectively. In addition, we derived a few more features based on confidence scores. For each response, we checked how many words had confidence scores below a threshold and computed the percentage of these words over the total recognized words (denoted as $(x)conf_n^w$). Since the confidence scores computed in the previous subsection were duration normalized, to put more weight on long words, we also computed the percentage of phonemes that were in the low confidence words over the total recognized phonemes (denoted as $(x)conf_n^p$). For each type of confidence score, we tried several different thresholds. In our notation, underscore n stands for a specified threshold. These thresholds were chosen based on the training set to maximize the discriminative power of each individual feature.

Besides using confidence scores as features to detect the off-topic responses, we also tried some other features produced during the ASR process to see if they could help with the detection. We denoted the averaged language model likelihood as $lmloglike$, the speech duration as $duration$, word per minute as wpm, the number of unique grammatical verb forms (POS tags) in the recognized string normalized by the total number of verbal POS tags in the tag set as $gvar$. When we calculated these values, including previous confidence-based features, we did not include hesitations, mouth noises, or pauses.

2.3. Garbage Modeling

An out-of-vocabulary (OOV) word will be forcedly recognized as a word appearing in the item-specific language model even if their pronunciations are quite different. Our garbage modeling tries to capture OOV words with a garbage word. When a response is off-topic, most of its words will not be covered by the item-specific language model, which included about 520 words on average. Based on this assumption, we built another language model to account for OOV words [14, 15]. Our reasoning was that we could detect off-topic responses by inferring the presence of OOV words.

After we obtained the recognition result R by using the item-specific language model, we built a new language model that was a directed graph having a path from the first word to the last word according to the sequence in R. Then for each word we added an alternative path to a garbage model to allow for the substitution of the original word with a garbage word. Figure 1 shows the structure of the model. p was the probability of choosing the garbage word and was set to a low value (in our case, p=0.05). We tried two different simple methods to model garbage words. First, we tried a model in which the garbage word could be any arbitrary phoneme sequence. We denoted this model as $garbage_{all}$. Second, we tried a model in which the garbage word could be a sequence of any combination of mouth noises and/or hesitations. We denoted this model

as $garbage_{noise}$. We then rerecognized each response with this new language model and obtained the result $\hat{R}$. We computed the edit distance between $\hat{R}$ and R and derived the percentage difference, which might approximate how many OOV words exist in the response.

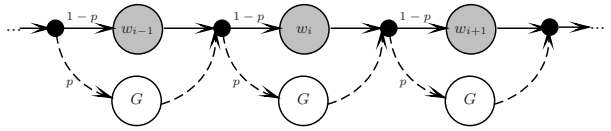

Figure 1: The language model based on the recognition result with the ability to absorb OOV words. G is the garbage word.

Not all the features mentioned in this section were used for the final model building process. We used the training set to select the best ones. A list of the selected best features are in Table 3. We listed few here for reference: $lconf^{\mu}$, $lconf_2^p$, $mconf_1^p$, $mconf^{\mu}$, $garbage_{noise}$, $garbage_{all}$, $aconf^{\mu}$, $lmloglike$, $aconf_1^w$, $gvar$, wpm, and $duration$. Although other papers [10, 11, 12] used duration-related features to compute confidence scores or perform utterance verification, the few duration-related features we tried did not improve the detection accuracy.

3. Experiments and Results

Experimental Data The recordings for "Describe image" and "Re-tell lecture" from PTE Academic were used. To collect more off-topic responses, we asked some speakers to take the PTE test and intentionally game the system, for example, by speaking in a language other than English, or using baby talk or nonsensical speech. We filtered the data by using only responses with a speech duration above 4 seconds, since utterances shorter than this are usually assigned low scores anyway.

We had human raters rate the responses according to the rating criteria listed in Table 1. Every response was rated by two raters. When there was some dispute (non-ideal agreement), a third super-rater gave the final judgment.

Scale	Criteria
0	Poor sound quality: This response has a quality of sound (recording noise, buzzing, mouth so close to the mic that the volume is too loud, etc.) that is so poor that a human would not be able to rate it.
1	Gibberish: The response is nonsense words, gibberish, gobbledygook, noise, or a foreign language.
2	Off-topic: There is a verbal response in intelligible English but it is mostly or completely off-topic (could be due to ignorance or miscomprehension about the task, or could be due to a malicious test of the integrity of the system test-taker)
3	Unintelligible English: The response sounds like a good-faith attempt to respond in English, but is so distorted, unintelligible, mispronounced, or disfluent that you are hard-pressed to understand it.
4	Scorable: The response is scorable (i.e. a human could give it a score with confidence, and would expect the system could also score it).

Table 1: The rubric used by human raters to classify the responses.

We built four data sets. Responses from the same subject were assigned to the same set. We selected those responses with scores 1 or 2 as off-topic responses and 4 as scorable responses. The goal was to detect the off-topic responses accurately. We randomly split the data set such that about 60% was used for training and the rest was used for testing. We called them the natural sets (N-Training and N-Test). To generate more data for training and also to evaluate the proposed methods from a different angle, we also built artificial data sets. First we selected real PTE test takers whose scores were very high. We treated their responses as on topic. We split these responses into two groups: for the first group, we kept the original audio files; for the second group, we replaced the original audio file with one from the same task but from a different test item. Our expectation was that all the responses in the second group should be rejected. Then, we split the groups into training and test sets (called A-Training and A-Test). These four data sets are listed in Table 2. We also denoted a combination from both training sets as C-Training, from both test sets as C-Test.

Set	#Off-Topic	#Scorable
N-Training	283	539
N-Test	193	364
A-Training	305	745
A-Test	288	351

Table 2: The number of responses in four different sets. N- and A- stand for the natural and artificial set, respectively.

Results We used Receiver Operating Characteristic (ROC) Curves to measure the performance. The X axis is False Rejection Rate (FRR) and the Y axis is False Acceptance Rate (FAR). FRR is the percentage of false rejected responses by machine over total scorable responses. FAR is the percentage of false accepted responses by machine over total off-topic responses. Scorable or off-topic responses were judged by humans as mentioned in the previous subsection and served as the gold standard. Three measurements were considered. The first one was the area under the curve (AUC). The less the area under the curve was, the better the performance was. The second one was the equal error rate (EER), the intersection of the ROC curve with the diagonal. The last one was the value of FAR when FRR was fixed. In our case, we wanted to see how low FAR could go when FRR was fixed at a low value, such as 5%.

To get a rough idea of each individual feature's discriminative power, we drew ROC curves based on the single feature. We then computed the three different measurements and listed the selected best individual features in Table 3. A few features that had strong correlations with the ones used in the final model but that did not significantly improve the performance in the training set were removed. From the table we can see that the confidence features were the most powerful. $lconf$ and $mconf$ were similar, with $lconf$ performing a little better. $aconf$ performed the worst among the three confidence scores. All the other features that were not derived from the confidence did not perform well individually, but still, we noticed that these features enhanced the final performance in the combination model due to their relative independence.

We built logistic regression classifier models using the training sets and then reported the performance on the test sets. We considered the computed value from this classifier model as a kind of confidence score for a response. The results are presented in Table 4. Performances using the natural test set and the artificial test set were similar, with performance slightly worse with the artificial set. But when we trained a model from the combined data set, the performance became worse. It could be caused by the more diverse properties in the combined training set. When we applied the trained model to the test set that did not correspond to the training set, we observed a significant decrease in performance. This finding suggests that it is important that the training materials have the same dynamic properties

Feature	AUC	EER	FAR
$lconf^{\mu}$	0.048	0.091	0.254
$lconf_2^p$	0.052	0.107	0.363
$lconf_1^p$	0.054	0.107	0.368
$mconf_1^p$	0.078	0.155	0.295
$mconf^{\mu}$	0.079	0.161	0.295
$lconf^{max}$	0.101	0.175	0.409
$mconf_2^p$	0.136	0.209	0.368
$lconf_2^w$	0.176	0.236	0.433
$lconf^{\sigma}$	0.177	0.223	0.731
$lconf^{min}$	0.190	0.272	0.710
$garbage_{noise}$	0.193	0.245	0.798
$mconf_2^w$	0.202	0.275	0.459
$garbage_{all}$	0.248	0.319	0.798
$aconf^{\mu}$	0.259	0.306	0.953
$lmloglike$	0.282	0.326	0.798
$aconf_1^w$	0.308	0.357	0.655
$gvar$	0.334	0.384	0.886
wpm	0.362	0.392	0.839
$duration$	0.410	0.441	0.865

Table 3: Performance of individual features when detecting off-topic responses from the natural test set. No machine learning methods were involved. For FAR values, FRR was fixed at 5%.

Modeled From	Test Set	AUC	EER	FAR
N-Training	N-Test	0.022	0.078	0.098
A-Training	A-Test	0.022	0.084	0.111
C-Training	C-Test	0.029	0.088	0.129
N-Training	A-Test	0.044	0.120	0.198
A-Training	N-Test	0.055	0.100	0.322

Table 4: Performance of the logistic regression classifier when detecting off-topic responses from the test set. For FAR values, FRR was fixed at 5%.

as future test sets in speech off-topic detection. ROC curves of different methods are shown in Figure 2. Compared with the best performance from the individual features, the combination model improved the performance significantly: when we fixed FRR at 5%, FAR decreased from 25.4% to 9.8% (more than a 61% improvement); AUC decreased from 0.048 to 0.022 (more than a 54% improvement).

Experimental results show that confidence scores have strong predictive power for off-topic detection and garbage modelings give discriminative information, too. Combining those good features can reliably detect off-topic responses.

4. Conclusions

Off-topic detection is important for face validity of automated speech assessments. Different from traditional topic detection in text documents, our method mainly focused on using the features derived from speech confidence scores, instead of recognized text. Three different methods for computing confidence scores were discussed. We also investigated measurements derived from other sources including acoustic likelihood, language model likelihood, and garbage modeling. By building a logistic regression model combining all the selected features together, we achieved a false acceptance rate of 9.8% when fixing the false rejection rate at 5% in our test set. These findings suggest that a combination model allows for the best off-topic detection.

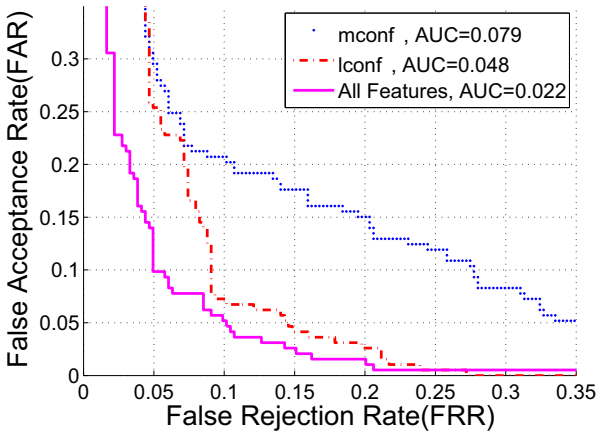

Figure 2: ROC curves of different methods. *All Features* stands for the logistic regression classifier using all selected features.

5. References

[1] T. Joachims, *Learning to Classify Text Using Support Vector Machines – Methods, Theory, and Algorithms*, Kluwer, 2002.

[2] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[3] T. Hofmann, "Probabilistic latent semantic indexing," in *SIGIR99*, pp. 50–57.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[5] L. R. Lane, T. Kawahara, T. ad Matsui, and S. Nakamura, "Out-of-domain detection based on confidence measures from multiple topic classification," in *ICASSP 2004*, pp. 757–760.

[6] D. Higgins, J. Burstein, and Y. Attali, "Identifying off-topic student essays without topic-specific training data," *Natural Language Engineering*, vol. 12, no. 2, pp. 145–159, 2006.

[7] Pearson, "Skills and scoring in PTE Academic," `http://www.pearsonpte.com/SiteCollectionDocuments/US_Skills_Scoring_PTEA_V3.pdf`, 2011.

[8] J. Bernstein and J. Cheng, "Logic and validation of a fully automatic spoken English test," in *The Path of Speech Technologies in Computer Assisted Language Learning*, V. M. Holland and F. P. Fisher, Eds., pp. 174–194. Routledge, New York, 2007.

[9] J. Bernstein, A. Van Moere, and J. Cheng, "Validating automated speaking tests," *Language Testing*, vol. 27:3, pp. 355–377, 2010.

[10] H. Jiang, "Confidence measures for speech recognition: A survey," *Speech Communication*, vol. 45, pp. 455–470, 2005.

[11] J. van Doremalen, H. Strik, and C. Cucchiarini, "Utterance verification in language learning applications," in *SLaTE 2009*.

[12] W. Lo, A. Harrison, and H. Meng, "Statistical phone duration modeling to filter for intact utterances in a computer-assisted pronunciation training system," in *ICASSP 2010*, pp. 5238–5241.

[13] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech and Language*, vol. 14, no. 4, pp. 373–400, 2000.

[14] H. Sakamoto and S. Matsunaga, "Detection of unknown words using garbage cluster models for continuous speech recognition," in *EUROSPEECH-1995*, pp. 2103–2106.

[15] T. Hazen and I. Bazzi, "A comparison and combination of methods for OOV word detection and word confidence scoring," in *ICASSP 2001*, pp. 397–400.