Proceedings Paper:

Errattahi, R., El Hannani, A., Ouahmane, H. et al. (1 more author) (2016) Automatic Speech Recognition Errors Detection Using Supervised Learning Techniques. In: 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA). 13th International Conference of Computer Systems and Applications (AICCSA), Nov 29 – Dec 02, 2016, Agadir, Morocco. IEEE .

https://doi.org/10.1109/AICCSA.2016.7945669

Automatic Speech Recognition Errors Detection Using Supervised Learning Techniques

Rahhal Errattahi*, Asmaa El Hannani*, Hassan Ouahmane*, and Thomas Hain†
*Laboratory of Information Technologies, National School of Applied Sciences
University of Chouaib Doukkali, El Jadida - Morocco
Email: {errattahi.r, elhannani.a, ouahmane.h}@ucd.ac.ma
†Speech and Hearing Group, Department of Computer Science
University of Sheffield, UK
Email: t.hain@sheffield.ac.uk

Abstract—**Over the last years, many advances have been made in the field of Automatic Speech Recognition (ASR). However, the persistent presence of ASR errors is limiting the widespread adoption of speech technology in real life applications. This motivates the attempts to find alternative techniques to automatically detect and correct ASR errors, which can be very effective and especially when the user does not have access to tune the features, the models or the decoder of the ASR system or when the transcription serves as input to downstream systems like machine translation, information retrieval, and question answering. In this paper, we present an ASR errors detection system targeted towards substitution and insertion errors. The proposed system is based on supervised learning techniques and uses input features deducted only from the ASR output words and hence should be usable with any ASR system. Applying this system on TV program transcription data leads to identify 40.30% of the recognition errors generated by the ASR system.**

1. Introduction

Automatic Speech recognition (ASR) is the process by which a machine converts a speech signal into a sequence of words either for text-based communication purposes or for device control. The importance of such technology reach when using the traditional input devices (as keyboard, mouse, and touchscreen) is not possible such, for example, when our hands are busy or with limited mobility, when we are using the phone, we are in the dark, or we are moving around etc.

ASR has improved over the last decade to the point of commercial applications by providing transcription with an acceptable level of performance, especially with the introduction of deep learning technologies (as Deep Neural Networks DNN) for acoustic modeling, which allows integration into many applications such as dictation, meeting and lectures transcription, speech translation, voice-search, phone-based services and others. Those systems are, in general, effective when the conditions are well controlled, especially when the speech is recorded under clean condition.

The Large Vocabulary Continuous Speech Recognition (LVCSR) task poses a particular challenge to ASR technology developers and still one of the most challenging tasks in the field, due to a number of factors, including poor articulation, speaking rate and high degree of acoustic variability caused by noise, side-speech, accents, sloppy pronunciation, hesitation, repetition, interruptions and channel mismatch, and/or distortions. Performances of LVCSR systems vary from domain to other. In some domains, as read continuous speech where generally the speech was recorded under clean conditions, results are satisfactory with an error rate under 5%. While in other domain that contain more speech variations, as video speech, telephone conversations or distant conversational speech (meeting), results are not acceptable presenting an error rate near to 50% in certain conditions [1], [2].

Even though many algorithms and technologies have been proposed by the scientific community for all steps of LVCSR over the last decade, including pre-processing, feature extraction, acoustic modeling, language modeling, decoding and result post-processing. The problem of LVCSR is far from being solved, where more research efforts are still needed to reduce the height presence of errors in their outputs. To deal with this key problem and to enhance the performance of imperfect ASR systems, the automatic detection and correction of the transcription errors can, in some cases, be the only choice. Particularly, when tuning the ASR system itself is not possible (e.g. the system is purchased as a black-box) or when the manual correction is not convenient or even impossible as in the case where the transcription is not the final goal of the system (e.g. machine translation, information retrieval and question answering systems).

In this paper, we present an ASR errors detection system trained to differentiate between a correct word and an erroneous word in the transcript of an ASR system. This system is based on supervised learning techniques and used input features deducted only from the ASR output and hence should be usable with any ASR system. The focus of this work was on substitution and insertion errors. The proposed system was evaluated on the transcription of a dataset from Multi-Genre Broadcast Media using Sheffield ASR system [3].

The remainder of the paper is organized as follows. In Section 2 we give a brief overview of previous works in the field. Section 3 describes our proposed errors detection system, including feature extraction. Section 4 describes experiments that were conducted on, including a detailed description of the data set used either for training and testing our system, and the ASR system used in transcription. In Section 5 we present a detailed discussion of the results in comparison with other existing results. Finally, in Section 6 we give some concluding remarks and future directions of this work.

2. Related works

There are two categories of research, that address the subject of errors detection in ASR systems, in the literature: The first one focused on features generated from the ASR decoder, such as confidence scores [4], linguistic information, confusion networks etc. The second one used additional features generated from hypothesized word sequence, such as n-grams, parts of speech, syntactic features, semantic features etc.

2.1. Decoder based features

For the decoder based features, Zhou et al. [5] addressed the issue of errors detection in ASR, especially in Dictation Speech Recognition (DSR), by using data-mining techniques. Their study consists of using three different data-mining classifiers, including Nave Bayes (NB), Neural Networks (NN), and Support Vector Machines (SVM) for detecting errors in DSR. The three models were trained to identify errors using features extracted from DSR output, including confidence scores and linguistic information. Results of this study have shown that those systems could identify until 50% of ASR errors.

Another study [6] proposed the use of additional features extracted from the confusion networks and estimated a correctness probability using logistic regression based on those features. The proposed system achieved a Classification Error Rate (CER) of 12.3% on a French broadcast news corpus.

Pellegrini et al. [7] investigated the use of a Markov Chains (MC) classifier with two states: error state and correct state, to model errors using a set of 15 features common in errors detection. The resulting system was tested on American English broadcast news speech NIST corpus and it has achieved 16.7% CER.

2.2. Non-decoder based features

In [8], a word correctness prediction system was constructed for use in a dialog domain. In this work, the authors investigated the use of 11 features: (i) 9 decoder based features in four categories: acoustic features, language model features, word lattice features, and N-Best list features, and (ii) 2 non-decoder based features: a parsing-mode feature indicating if a word is parsed by the grammar and a slot-backoff-mode feature using a bigram language model for the slots. They have shown that both decoder based features and non-decoder based features independently contribute to the correctness prediction accuracy and that Support Vector Machines (SVM) appear to be an affective classifier for this task compared to the Decision Tree and Neural Nets classifiers, with 18.2% CER in a travel-planning domain. This finding has been confirmed in [9], where simlar SVMs were constructed for phoneme correctness prediction in ASR.

In [10] Pellegrini et al. suggested using, in addition to the traditional decoder based features, non-decoder based features extracted from other sources: a binary word match feature that presents a binary comparison between two different ASR systems, a bigram hit feature measuring the number of hits found by querying a very popular Web search engine, and a topic feature to identify if a word is out of the global topic of the hypothesized sentence. The introduction of these non-decoder based features led to significant improvements, from 13.87% to 12.16% CER with a maximum entropy model, and from 14.01% to 12.39% CER with linear-chain conditional random fields, comparing to a baseline using only decoder-based features.

Chen et al. [11] proposed a system for errors detection in conversational spoken languages translation. In addition to traditional features obtained from ASR outputs, this system used additional features provided as the feedback of Statistical Machine Translation (SMT), including SMT confidence estimates and posteriors from named entity detection. Furthermore, this system used an automated word boundary detector based on acoustic-prosodic features to verify the existence of ASR-hypothesized word boundaries. This system provided 2.8% absolute improvement in errors detection over the error detector that used features traditionally employed in the field (e.g. ASR confidence score, LM perplexity, confusion network density and phonetic acoustic model score deviation).

3. Proposed Method

In this work, we propose supervised learning based techniques for ASR errors detection. As a first attempt we automate the detection process of the transcription errors with the intention of automating the correction process as well in the future. The work consists of training a classifier to differentiate between a correct word and an erroneous word in the transcript of an ASR system.

There are three types of errors that occur in ASR. First, substitution; where a word in the reference word sequence is transcribed as a different word. Second, deletion; where a word in the reference is completely missed in the automatic transcription. And finally, insertion; where a word appears in the automatic transcription that has no correspondent in the reference word sequence. In this paper we only address

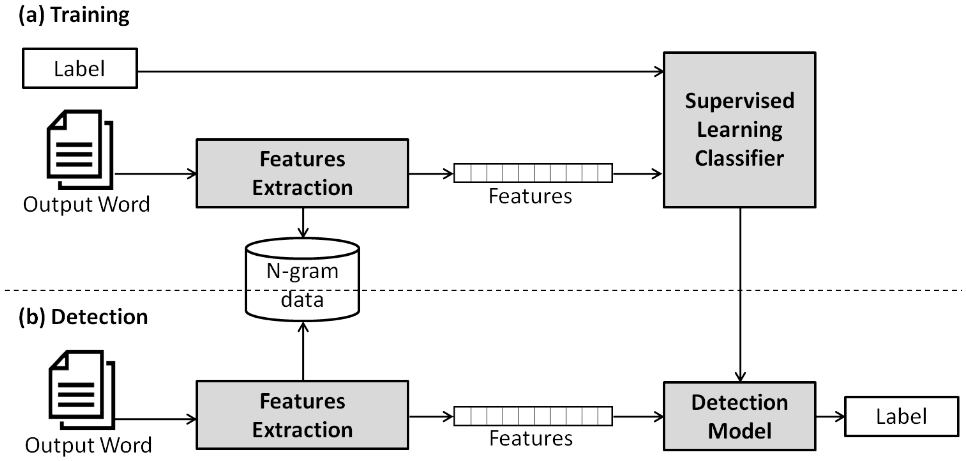

Figure 1. General process of the error detection system.

the substitution and insertion errors because in the case of deletion it is difficult to decide if a non-existent word is correct or not.

Figure 1 illustrates the general process of the proposed system. Our approach consists of two steps: a training phase and a prediction phase. During training, a feature extractor is used to convert each word from the ASR transcription to a feature set. This feature set, which capture the basic information about each word that should be used to classify it, will be discussed in the next section. Both, feature sets and labels (correct, substitution and insertion) are presented to the machine learning classifier to generate a classification model. During prediction, the same feature extractor is used to convert recognized words to feature sets. These feature sets are then fed into the model, which generates predicted labels: correct or error.

3.1. Features

In order to identify the errors in the ASR output, we extract a set of features that capture the basic information about each word. Those features should potentially indicate if a given word is correct or not. Given that the majority of recognition systems are used as a black-box and that the user does not have access to the internals of the system, we attempt to develop our system using a minimum of information from the ASR decoder.

The majority of ASR systems generate, in addition to the output words, a score (probabilistic value between 0 and 1) called confidence score to indicate the trustworthiness of any word recognition made by the ASR decoder. Otherwise, a confidence score can be computed for every word to indicate how likely it is correctly recognized or for an utterance to indicate how much we can trust the results for the utterance as a unit. Many researches have confirmed that the confidence score can be used to predict the correctness of a given word [12], such that, correct words tend to have higher confidence score than errors. Thus the confidence score of a recognized word (w_i) in its context could help to make a decision if the given word is correct or not. The three selected features based on confidence score are:

CS: Confidence score of the recognized word;
LCS: Confidence score of the preceding (Left) word (w_{i-1});
RCS: Confidence score of the following (Right) word (w_{i+1});

In addition to the confidence score based features, we can use the bigram language model to verify the context of an output word. For this we calculate the probability $P(w_i|w_{i-1})$ that the recognized word w_i comes with its preceding word in the transcription w_{i-1}, and the probability $P(w_{i+1}|w_i)$ that w_i is followed by the next word in the transcription w_{i+1}:

LBG: Bigram language model of (w_{i-1}, w_i);
RBG: Bigram language model of (w_i, w_{i+1});

Given that:

$$P(w_i|w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})} \quad (1)$$

According to Fong et al. [13], if a word is a substitution, then it is expected that it won't fit well in the context of the sentence. And when we discard this word and it was not meaningful in the context, then the frequency of the bag-of-words with the word discarded should be greater than that of the original bag. This can be represented using a sentence oddity as follows:

$$SO = \frac{\textit{frequency of the bag-of-words with the word discarded}}{\textit{frequency of the entire bag-of-words}} \quad (2)$$

The concept of sentence oddity was introduced first for the problem of detecting substitution word in intercepted communication. This problem is called obfuscation where words that might raise attention are replaced by other innocent words that are in general not meaningful in the context of the sentence. In this paper, we adopt the definition of the sentence oddity [13] for the problem of ASR errors detection, starting from the similar proposition: ”If a word is a substitution or insertion, then it is expected that it won’t fit well in the context of the sentence”. Thus, instead of calculating the frequency of bag-of-words we propose to use the sentence probability using the maximum likelihood estimation. So, for a given $sentence = w_1, w_2, ..., w_n$, we redefine the Sentence Oddity (SO) as:

$$SO = \frac{P(\textit{sentence with the word discarded})}{P(sentence)} \tag{3}$$

With:

$$P(sentence) = P(w_1, w_2, ..., w_n) = \prod_{k=1}^{n} P(w_k|w_{k-1}) \tag{4}$$

We extract in total 6 features (CS, LCS, RCS, LBG, RBG, SO) for each word in the ASR output and then use them to train a classifier to predict whether a recognized word is correct or not.

3.2. Classifiers

We investigated the ASR errors detection using three supervised learning classifiers: Classification and Regression Tree (CART) [14], Neural Network (NN) [15] and Bayesian Network (BN) [16]. The CART classifier uses a standard information gain criterion. The NN classifier is a single backward propagation network with one hidden layer. The number of neurons in the hidden layer was chosen empirically in the range of 0.5 to 1 times the total size of the input and output layers. The BN classifier uses the simple estimator function to estimate the conditional probability and the k2 algorithm to heuristically search for the most probable beliefnetwork structure. All experiments were performed using the Weka Machine Learning Software [17].

4. Experimental setup

4.1. Data and ASR System

To train and evaluate the effectiveness of our proposed approach, we conducted experiments on a recent and very challenging dataset from the Multi-Genre Broadcast (MGB) Challenge [18]. The MGB data is a large broad and multi-genre, spanning the whole range of TV output. The Automatic transcriptions were produced by the Sheffield system described in [3]. This system was built using two different types of systems. The first ones are Hybrid DNN-HMM systems, where the DNNs consisted of 6 hidden layers of 2,048 neurons, and an output layer of 6,478 triphone state targets. The second are Bottleneck DNN-GMM-HMM systems, where (i) the DNNs consisted of 4 hidden layers of 1,745 neurons plus 26-neuron bottleneck layer, and an output layer of 8,000 triphone state targets, and (ii) the GMM-HMM models were trained using 16 Gaussian components per state, and around 8k distinct triphone states. The Sheffield ASR system has a performance that varies significantly from news shows, with a 13.2% WER, to comedy shows, with a 40.9% WER and a 27.5% WER as the global result, which reflects the complexity of the chosen task.

We first performed a preprocessing on the MGB data to remove utterances with hesitations. Then using the alignment provided by the scoring script we connected each hypothesis word with its equivalent reference word, and we added a label to the word; correct, substitution, insertion or deletion. Given that we decided to work only on substitution and insertion errors, we performed an additional preprocessing step to eliminate deletion errors from the samples. The resulted dataset was divided into two parts. The first split is about 70%, which is considered as the training set. It contains 13912 utterances from 30 speakers, with a total of 102570 words. The remained data (30%), will be used to evaluate the system. The test set contains 7788 utterances from 16 speakers, with a total of 51562 words. The distribution of the words in the training and test sets is summarised in Table 1. If we analyse the training set distribution we can see that the correct words represent about 83% of the samples in this set, which means that we have unbalanced classes in our training set. In this case the predictive model will ignore the error class. Therefore we performed a sampling on our training data by duplicating instances from the under-represented class (over-sampling). For the n-gram based features, we used the google book n-gram data [19].

TABLE 1. Words distribution in the training and test sets.

Class	Training set	Test set
Correct	85054	40803
Substitution	13844	8532
Insertion	3672	2221

4.2. Evaluation Metrics

In order to measure the performance of our predictive model and to compare our results with the literature, we used four popular classification evaluation metrics: Accuracy (ACC), Precision (PRE), Recall (REC) and F-measure (FM). Accuracy is the most intuitive performance measure for the classification tasks. It is simply the percent of correctly classified words by the predictive model and is calculated as follows:

$$ACC = \frac{\textit{\# of correctly classified words}}{\textit{total \# of words}} \tag{5}$$

The precision measures the number of words correctly classified as belonging to a given class, divided by the total number of elements labelled as belonging to this class. So for the correct class, it is the percentage of words detected

TABLE 3. Classification performance (% PRE, % REC and % FM) based on different features combinations and using three supervised classifiers, on the Test set.

Classifier	Features	Correct			Error		
		PRE	PEC	FM	PRE	REC	FM
NB	CS,LCS,RCS	84.00	87.80	85.86	44.00	36.50	39.90
	CS,LCS,RCS,LBG,RBG	85.80	80.70	83.17	40.30	49.40	44.39
	CS,LCS,RCS,LBG,RBG,SO	85.80	80.60	83.12	40.20	49.50	44.37
NN	CS,LCS,RCS	79.10	100.00	88.33	0	0	-
	CS,LCS,RCS,LBG,RBG	80.10	99.10	88.59	64.90	6.60	11.98
	CS,LCS,RCS,LBG,RBG,SO	80.80	97.80	88.49	58.80	11.70	19.52
CART	CS,LCS,RCS	80.60	97.90	88.41	56.40	10.50	17.70
	CS,LCS,RCS,LBG,RBG	81.50	97.10	88.62	59.70	16.20	25.48
	CS,LCS,RCS,LBG,RBG,SO	81.50	97.00	88.58	59.50	16.60	25.96

as correct that are, indeed, correct words. And for the error class, it is the percentage of words detected as error that are, indeed, erroneous words.

The recall is defined as the total number of words correctly classified as belonging to a given class, divided by the total number of elements that actually belong this class. In our context, for the correct class, it is the proportion of actual correct words that are correctly classified as correct. And for the error class, is the proportion of actual ASR errors that are correctly classified as errors.

The last performance measure is the F-measure, which is a quality score for the classification results and is obtained by combining the indices of recall and precision. The formula for FM score is:

$$FM = 2 * \frac{PRE * REC}{PRE + REC} \tag{6}$$

TABLE 2. Detection Accuracy in %, based on different features combinations on the Test set.

Features	Classifier		
	NB	NN	CART
CS,LCS,RCS	77.05	79.13	79.63
CS,LCS,RCS,LBG,RBG	74.18	79.77	80.23
CS,LCS,RCS,LBG,RBG,SO	74.08	79.86	80.24

5. Results & Discussion

Table 2 presents the performance of our ASR errors detector based on different features combinations and using three supervised classifiers on our test set. Firstly, it is shown that all the three models achieve encouraging results in term of classification accuracy. In particular, both NN and CART outperform NB, with accuracy around 80%. On the other hand, it can be seen by comparing the same results but using different features combinations that using only confidence score (CS, LCS, RCS) based features gives high accuracies across the three classifiers, which confirms the effectiveness of confidence score for ASR errors detection. Adding N-gram based features (LBG, RBG) and Sentence Oddity (SO) enhanced slightly the classification performance.

In Table 3, we present extra details about the classification performances based on the different features combinations and using the three classifiers by presenting the PRE, REC and FM of each class (correct versus error). The results show that adding both n-gram based features(LBG, RBG) and SO improves the detection performance overall the three measures. For example, the PRE and the REC of the NN classifier increased from 0% to 58.80% and from 0% to 11.70%, respectively, after combining the confidence score based features with n-gram based features and SO. In addition n-gram based features and SO are generally effective in detecting errors because they are susceptible to have low values when an error occurs. In contrast, they are less useful in detecting correct words. They increase PRE but their introduction decrease slightly both REC and FM measures.

The comparison of the three classifiers in Table 3, shows that the NN and CART classifiers present better results on correct words detection on all the three measures in comparison with the NB. For example, the FM of the CART classifier reached 88.62%, while the best value achieved by the NB was 85.86%. However, on the erroneous words detection, the NB presents the best results with an FM of 44.39% (40.30% PRE and 49.40% REC) compared to 19.52% for the NN and 25.96% for the CART.

Our system can be compared with the work of Pellegrini et al. [10], where the authors performed a similar approach for errors detection using data-mining techniques. In this paper, they performed experiments on two different dictation sets. In their case, the FM for error words detection ranged from 55.3% to 62.5% on the first dataset and from 30.19% to 41.3% on the second dataset. In the other side, the best result reached by our system, was by using NB where the FM is ranged from 39.90% to 44.39%. Comparing these two findings, we can confirm that our model is very competitive, given that we outperform one of the results achieved in [10]. We should also notice that in our experiments we used a very challenging task which is TV programs transcription where speech is more spontaneous, whereas in [10], Pellegrini et al. performed their experiments on dictation tasks where speech is recorded in a quite lab environment using reading text. Also, they used a very large feature set with 37 features; most of them are depending on the ASR decoder, such as alternative hypothesis. In comparison, in our work we used only 6 features where solely confidence score is depending on the ASR decoder, bearing in mind that the majority of ASR systems provide the confidence score of the hypothesis word as output.

6. Conclusion

We have presented an automatic system for the ASR errors detection task. The system was tested using three supervised classifiers, with a set of 6 features obtained from the ASR output. Evaluating this system using TV programs transcriptions data from the MGB challenge gives encouraging results, with an FM of 83.17% for correct words detection and 44.39% for erroneous words detection. Our experimental results confirmed the utility of confidence score features in the task of ASR errors detection and also proved the positive influence of both n-gram and SO features on the system performance. The proposed system is based on features extracted exclusively from the ASR output and hence should be usable with any ASR system.

In future works, we will continue developing our system along several axes. Firstly, we will investigate the possibility of adding supplementary features, such as lexical and semantic features, with further refinements of our training model, while providing additional training data from different tasks and using different ASR decoders. We also intend to consider the deletion errors and investigate the usefulness of web based information in order to replace the n-gram dictionary. And finally, our ultimate aim is to develop a post-editing ASR errors correction system to correct the erroneous segments detected in the automatic transcription.

Acknowledgments

This work is supported by the CNRST Morocco under Contract No. 5UCD2015.

References

[1] P. Swietojanski, A. Ghoshal, and S. Renals, "Convolutional neural networks for distant speech recognition," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1120–1124, 2014.

[2] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, "Application of pretrained deep neural networks to large vocabulary speech recognition," in *the proceedings of interspeech*, 2012.

[3] O. Saz, M. Doulaty, S. Deena, R. Milner, R. W. Ng, M. Hasan, Y. Liu, and T. Hain, "The 2015 sheffield system for transcription of multigenre broadcast media," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015.

[4] H. Jiang, "Confidence measures for speech recognition: A survey," *Speech communication*, vol. 45, no. 4, pp. 455–470, 2005.

[5] L. Zhou, Y. Shi, J. Feng, and A. Sears, "Data mining for detecting errors in dictation speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 681–688, 2005.

[6] A. Allauzen, "Error detection in confusion network." in *the proceedings of Interspeech*, 2007, pp. 1749–1752.

[7] T. Pellegrini and I. Trancoso, "Error detection in broadcast news asr using markov chains," in *Human Language Technology. Challenges for Computer Science and Linguistics*. Springer, 2009, pp. 59–69.

[8] R. Zhang and A. I. Rudnicky, "Word level confidence annotation using combinations of features," in *the proceedings of the European Conference on Speech Communication and Technology (Eurospeech)*, 2001, pp. 2105–2108.

[9] M. Gibson and T. Hain, "Application of svm-based correctness predictions to unsupervised discriminative speaker adaptation," in *the proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012, pp. 4341–4344.

[10] P. Thomas and T. Isabel, "Improving asr error detection with non-decoder based features," in *the proceedings of interspeech*, 2010, pp. 1950–1953.

[11] W. Chen, S. Ananthakrishnan, R. Kumar, R. Prasad, and P. Natarajan, "Asr error detection in a conversational spoken language translation system," in *the proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2013, pp. 7418–7422.

[12] L. Gillick, Y. Ito, and J. Young, "A probabilistic approach to confidence estimation and evaluation," in *the proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2. IEEE, 1997, pp. 879–882.

[13] S. Fong, D. Skillicorn, and D. Roussinov, "Detecting word substitution in adversarial communication," in *in In the proceedings of 6th SIAM Conference on Data Mining. Bethesda, Maryland*, 2006.

[14] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[15] D. E. Rumelhart, J. L. McClelland, P. R. Group *et al.*, "Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1-2," *Cambridge, MA*, 1986.

[16] I. J. Good, I. Hacking, R. Jeffrey, and H. Törnebohm, "The estimation of probabilities: An essay on modern bayesian methods," 1966.

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[18] P. Bell, M. Gales, T. Hain, J. Kilgour, P. Lanchantin, X. Liu, A. McParland, S. Renals, O. Saz, M. Webster *et al.*, "The mgb challenge: Evaluating multi–genre broadcast media transcription," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015.

[19] Y. Lin, J.-B. Michel, E. L. Aiden, J. Orwant, W. Brockman, and S. Petrov, "Syntactic annotations for the google books ngram corpus," in *the Proceedings of the ACL 2012 system demonstrations*. Association for Computational Linguistics, 2012, pp. 169–174.