Virtual Example for Phonotactic Language Recognition

Rong Tong, Bin Ma and Haizhou Li

Institute for Infocomm Research, Singapore
{tongrong,mabin,hli}@i2r.a-star.edu.sg

Abstract

One challenge in spoken language recognition is the availability of training data. In this paper, we propose a virtual example construction method to derive artificial training examples from the existing training data. Using the proposed method, both target virtual examples and non-target virtual examples can be derived from the available training samples. An iterative virtual example selection method is proposed to select those virtual examples that may provide extra discriminative information for language separation. By incorporating virtual examples in language classifier training, the language recognition performances are improved for both closed-set and open-set tasks. Specifically, for LRE 2009 evaluation data of three durations: 30-seconds, 10-seconds and 3-seconds, the language recognition performance improved by 3.67%, 11.98%, 6.42% respectively in closed-set conditions, and 10.14%, 10.55%, 5.75% respectively in open-set conditions.

Index Terms: automatic language recognition, virtual example, phonotactic feature

1. Introduction

Spoken language recognition (SLR) is a task to determine what language is spoken in a given speech utterance. Generally it can be categorized into two categories, closed-set language recognition and open-set language recognition. In a closed-set language recognition, the test utterances are extracted from a pool of languages that are known to the system. In an open-set language recognition, some of the test utterances are extracted from languages that are unknown to the system, those languages are referred as out-of-set (OOS) languages. Open-set language recognition is more challenging than closed-set task, as the system has no prior knowledge of OOS languages.

The data driven approaches are widely adopted in the automatic language recognition studies. In practice, the SLR is often solved by evaluating unknown test utterances on a set of language classifiers, which are trained from the target language samples. The amount of available training data is crucial to the accuracy of the language classifier. With adequate training data, the variations of the language characteristics can be observed hence the language classifier is more generalized.

Many studies are focused on exploiting a broad utilization of the existing training data. For closed-set language recognition task, deep learning techniques [1, 2] and i-vectors [3] are applied in learning the discriminative characteristics of different languages. A fusion system for low resource language recognition is presented in [4], in which the acoustic and phonotactic feature based sub-system compensates for each other in language recognition. Various dimension reduction methods are applied in language recognition to alleviate the problem of limited training data [5, 6, 7]. A method of using a different negative pool for different target languages is proposed for language pair recognition [8]. For open-set language recognition task, multilayer perceptron networks are trained from the available target language data, an unknown language sample is rejected if its output value is lower than a predefined threshold [9]. Different backend methods are studied in [10] for better language recognition performance. In [11, 12], a target independent Gaussian back end model is used to represent the out-of-set languages, this model is trained from the training data of all target languages.

Some works have studied in acquiring more training data for language recognition, especially for open-set language recognition. In [13, 14], the additional OOS development data sets are used for open-set language modelling and score calibration. For closed-set task, [15] proposed a co-training algorithm to adapt the language classifier, which incorporates a small amount of un-labelled data in compensating the channel mismatch in training and testing data.

In this work, we propose a method to augment the language training set by creating artificial data samples for phonotactic feature based language recognition. These these artificial samples are referred as virtual examples. The concept of virtual example has been explored in handwriting recognition by transforming the reference images [16]. Various methods of creating new training samples have been studied in related research areas. In speech recognition, new training samples are created by applying a random linear warping on the spectrograms of the existing speech utterances [17, 18]; in text classification, virtual examples are derived by making small changes on the available text samples [19, 20].

We extend the idea of virtual example to phonotactic language recognition. We augment the original training set by deriving both target and non-target virtual examples from the available training data. The advantages of incorporating virtual examples are: 1) No additional training data is required; 2) It is possible to derive more virtual examples for those languages that have small training sets, which may help with the data imbalance problem; 3) Some new phonotactic patterns of the target language may included in virtual examples; 4) The out-of-set languages can be explicitly modelled by using non-target virtual examples.

The proposed virtual example extraction method is validated on LRE 2009 language recognition evaluation set. Our experimental results show that the performance of language recognition is improved by incorporating the virtual examples in both closed-set and open-set test conditions.

2. Phonotactic feature and virtual example

Phonotactic features are used to capture the relatively higher level language characteristics, for example syllable structure, consonant clusters, and lexical constraint in a language.

The phonotactic feature can be represented by the n-gram

statistics. Suppose we have a phone recognizer, a speech utterance in any target language is converted into a sequence of phone *n*-gram statistics. The *n*-gram statistics can be derived from the best decoding results or from a lattice [21]. Although the same phone set is shared across all the target languages, their phonotactic statistics can differ considerably from one language to another. Hence the phonotactic statistics are used as features in automatic language recognition. A high dimensional phonotactic feature vector can be derived from the *n*-gram statistics:

$$X = [x_1\ x_2\ x_3\ x_4\ ...\ x_k\ ...] \quad (1)$$

Where x_k denotes a normalized *n*-gram statistics in the given speech segment.

In our implementation [22], a one-vs-others support vector machine (SVM) classifier is trained for each target language. Each classifier is trained with the phonotactic feature vectors of this language as the positive set and vectors from all the other languages as the negative set. An SVM classifier learns a binary decision over the feature vector X,

$$f(X) = \sum\nolimits_i y_i \alpha_i K(X_i, X) + b \quad (2)$$

Where α_i is the weight of the i-th support vector, y_i is the class label of the support vector, b is a threshold, and $K(.)$ is a kernel function and X_i is the i-th support vector.

2.1. Virtual examples

In perception tests, a listener can make good judgment on the language being spoken, even if he is not able to capture every word or sound in the speech [23]. Intuitively, we assume that the small alteration on the *n*-gram statistics vector may not affect the overall language characteristics of the utterance. We propose to create an artificial example by adding or deleting some *n*-gram elements from the original training sample (denoted as *reference utterance*). Such an altered *n*-gram statistics vector is denoted as a *target virtual example* (TVE). An TVE may still keep the phonotactic pattern of the target language, yet it introduces some phonotactic variations of the language.

Contrary to the target virtual example extraction, we take *n*-gram elements from different languages and combine them into a new *n*-gram statistics vector, denote it as a *non-target virtual example* (NVE). Such an NVE may represent a phonotactic pattern that is different from any of the existing target language, it could be used to represent non-target languages.

As defined in Equation 2, the support vectors are a subset of training samples that specify the classifier decision function. They can be seen as the most discriminating samples for language separation. Hence, in the proposed virtual example construction process, the support vectors of the existing language classifier are used as reference utterances.

The virtual examples derived by the random alteration are not guaranteed to be useful in language separation. We propose an iterative virtual example selection method that only keeps those virtual examples which are potentially has contribution to the language classification. The selection is based on the following principle: if a candidate virtual example can be correctly classified by the existing SVM classifier, it will not bringing new information to the classifier training, this virtual example will be discarded.

3. Virtual example construction

3.1. Target virtual example (TVE)

In this work, two target virtual example construction methods are proposed: *target virtual example by deletion* (TVE-D) and *target virtual example by addition* (TVE-A). In TVE-D method, a virtual example is derived by removing some *n*-gram elements randomly from a reference utterance, while in TVE-A method, a virtual example is derived by adding extra *n*-gram elements to a reference utterance, the extra *n*-gram elements are extracted from other reference utterances of the same language.

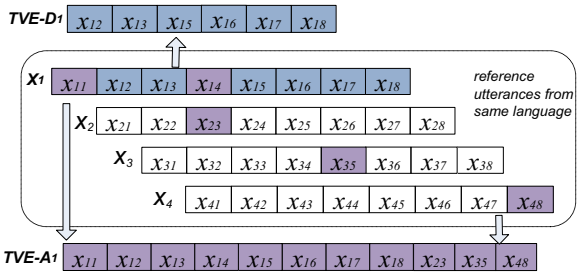

Figure 1: *Target virtual example construction (TVE)*

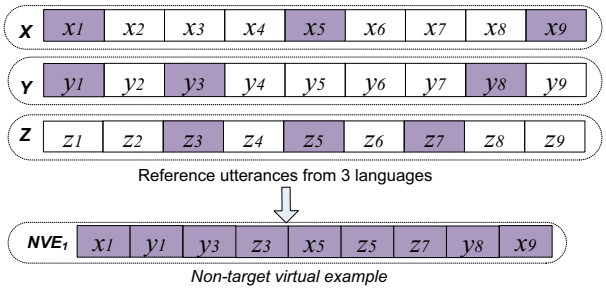

Figure 2: *Non-target virtual example construction (NVE)*

Figure 1 illustrates the proposed target virtual example extraction process. X_1, X_2, X_3, X_4 are reference utterances of the same language, they are support vectors obtained as Equation 2. Each of them is a *n*-gram statistics vector as described in Equation 1. TVE-D_1 is a target virtual example constructed by removing the elements: x_{11}, x_{14} from the reference utterance X_1. TVE-A_1 is a target virtual example constructed by augmenting the reference utterance X_1 with: x_{23}, x_{35}, x_{48}.

For both of the TVE-A and TVE-D methods, the following two parameters are used:

- n is the desired number of virtual examples
- t is a threshold which controls the ratio of elements to be added to or deleted from the reference utterance
- k is used to define the number reference utterances

Note that k is only used in TVE-A method. For instance, in Figure 1, TVE-D_1 is generated with n=1, t=0.25, as 1 virtual example is derived from reference utterance X_1, by removing $1/4$ (0.25) of the *n*-gram elements from reference sample. TVE-A_1 is generated with n=1, t=0.125 and k=3, as 1 virtual example is derived from reference utterance X_1, by adding $1/8$ (0.125) of the *n*-gram elements from 3 other reference utterances.

3.2. Non-target virtual example (NVE)

Figure 2 illustrates the non-target virtual examples (NVE) construction process. X, Y and Z represent the *n*-gram statistics vectors of three reference utterances, each from a different target language. A non-target virtual example NVE_1 is a union of 3 sets of *n*-gram elements: $\{x_1, x_5, x_9\}$ $\{y_1, y_3, y_8\}$ $\{z_3, z_5, z_7\}$.

In the non-target virtual examples construction process, three parameters are used:

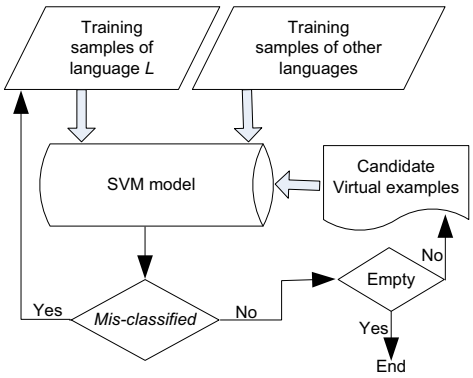

Figure 3: *Virtual example selection*

- n is the desired number of non-target virtual examples
- t is a threshold controlling the ratio of n-gram elements to be selected from each reference utterance
- k is the number of reference utterances required

For example, the virtual example NVE_1 in Figure 2 is derived with n=1, t=1/3 and k=3.

3.3. Virtual example selection

The virtual example selection follows two criteria: if a candidate virtual example can be correctly classified by the original model, the virtual example is not likely to provide extra information for classifier training. If a virtual example is misclassified, the original model is not able to separate this example, adding this example may help improve the classification capability.

The virtual example selection process is illustrated in Figure 3. For a target language L, an SVM model is trained using one-vs-others criterion with the original set of training samples. The candidate virtual examples are evaluated on this model. If a virtual example is mis-classified, we consider the virtual example has potential contribution to classification. It is added into the training set to train a new SVM model. The rest of the candidate virtual examples are evaluated on the new model until there is no more candidate virtual example to be selected, or no more virtual example is being mis-classified.

For each target language, a new language classifier is trained on the union of the original training samples and the virtual examples selected.

4. Experiment setup

We use the PPR-VSM [22] architecture in all the experiments. The BUT Hungarian phone recognizer [24] is used as the front-end tokenizer to convert the speech segments into phonotactic feature vectors which contain up to 3-gram statistics derived from the lattice. In the lattice generation, the Hungarian phone recognizer first tokenizes a speech utterance into posteriors, the HTK tool [25] is used to convert estimated posteriors into a lattice. The SRI lattice tool kit [26] is then used to derive n-gram counts from the lattice. In all the experiments, SVM-light [27] is used for SVM classifier training.

The language recognition system is trained on the LDC CallFriend, OHSU 2005 and LRE 2007 and LRE 2009 development data sets released by the LDC [28]. The training speech is first segmented into small segments by an energy based voice activity detection process. To match different durations of the evaluation data, multiple 30 seconds, 10 seconds, 3 seconds training samples are formed by combining consecutive small segments without overlap, they are separated into 8:2 ratio for classifier training and backend development respectively.

The proposed virtual example construction method is evaluated on LRE 2009 (LRE09) evaluation set. The evaluation set for LRE09 consists of samples from 23 target languages and 16 out-of-set languages. The evaluation set is grouped into 3 sets based on their length: 30 seconds, 10 seconds and 3 seconds. Table 1 shows the details of the target and out-of-set languages. The information of the out-of-set languages is not used in the model training process.

All the performances will be reported in an average cost detection function C_{avg}:

$$C_{\text{avg}} = \frac{1}{N_L} \sum_{L_T} \begin{cases} C_{\text{miss}} P_{\text{target}} P_{\text{miss}}(L_T) \\ + \sum_{L_N} C_{\text{FA}} P_{\text{non-target}} P_{\text{FA}}(L_T, L_N) \\ + C_{\text{FA}} P_{\text{out-of-set}} P_{\text{FA}}(L_T, L_O) \end{cases} \quad (3)$$

where N_L is the number of languages in the (closed-set) test, L_O is the Out-of-Set language, $C_{\text{miss}} = C_{\text{FA}} = 1$ and $P_{\text{target}} = 0.5$,

$$P_{\text{out-of-set}} = \begin{cases} 0.0 & \text{closed-set} \\ 0.2 & \text{open-set} \end{cases} \quad (4)$$

$$P_{\text{non-target}} = (1 - P_{\text{target}} - P_{\text{out-of-set}})/(N_L - 1) \quad (5)$$

In the evaluation, the average detection cost will be computed separately for each of the three duration categories, and for the closed-set and open-set conditions.

Table 1: *Languages in LRE09 evaluation set*

Target languages	Amharic, Bosnian, Cantonese, Creole (Haitian), Croatian, Dari, English (American), English (Indian), Farsi, French, Georgian, Hausa, Hindi, Korean, Mandarin, Pashto, Portuguese, Russian, Spanish, Turkish, Ukrainian, Urdu, Vietnamese
Out-of-set languages	Arabic, Azerbaijani, Belorussian, Bengali, Bulgarian, Italian, Japanese, Punjabi, Romanian, Shanghai-Wu, Southern-min, Swahili, Tagalog, Thai, Tibetan, Uzbek

5. Experiment results

5.1. Target virtual examples (TVE)

Figure 4 shows the language recognition performance on LRE09 closed-set by incorporating two types of target virtual examples (section 3). In TVE-D method, 2 (n=2) virtual examples are derived from each of the support vectors, the n-gram alteration parameter t varies from 0.1 to 0.4 (e.g. t=0.1 means 1/10 of the n-gram elements are removed from the reference utterances). In TVE-A method, the parameters are set to n=2, k=4 and t varies from 0.025 to 0.1. To make it comparable with the TVE-D method, they are labelled with the amount of alteration made, i.e., TVE-A 0.1 means $1/10$ (t=0.025, k=4) of n-gram elements are added.

The results in Figure 4 reveals that the language recognition performance is improved by incorporating target virtual examples for all the three test durations. There is no significant performance difference by using different alteration parameters in both TVE-D and TVE-A methods. Nevertheless, the TVE-D

Table 2: *Individual system and fusion results for LRE09 evaluation set*

	Closed-set (C_{avg}*100)			Open-set (C_{avg}*100)		
	30s	10s	3s	30s	10s	3s
Baseline	2.99	7.26	20.41	4.93	9.29	21.75
TVE-D 0.1	**2.84**	**6.36**	**18.83**	**4.53**	**8.36**	**21.25**
TVE-A 0.1	**2.94**	**6.53**	**19.45**	**4.83**	**8.41**	**21.32**
NVE 0.1	3.08	**6.85**	20.56	**4.80**	**8.99**	**21.09**
Fusion	**2.88** (3.67%)	**6.39** (11.98%)	**19.10** (6.42%)	**4.43** (10.14%)	**8.31** (10.55%)	**20.50** (5.75%)

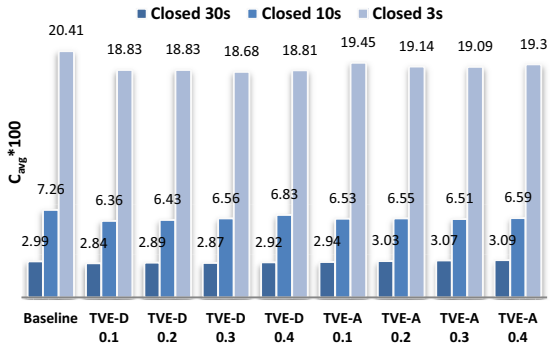

Figure 4: *Language recognition results of target virtual examples on LRE09 closed-set*

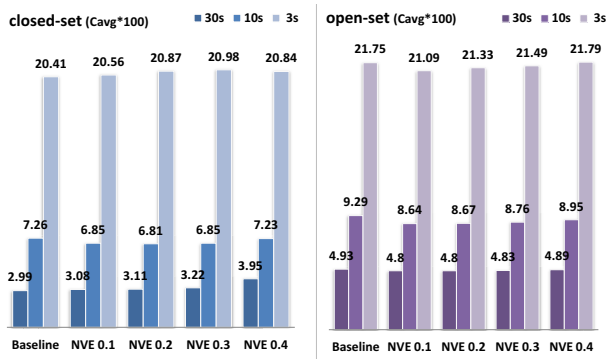

Figure 5: *Language recognition results of non-target virtual examples on LRE09 closed-set and open-set*

method gives slightly better improvements than TVE-A. One possible explanation is that the an TVE-D virtual example is derived by removing a small portion of elements from the reference utterance, the phonotactic pattern can be reserved, while in the TVE-A, extra n-gram elements from other reference utterances are added, the new phonotactic patterns may not be consistent with the reference utterance.

Another observation from Figure 4 is that larger performance gains obtained on shorter tests (10s and 3s) than long ones (30s). We believe this is due to the fact that there is a lack of discriminative information in the short test segments, virtual examples provided complementary information which is not conveyed in the original utterances.

5.2. Non-target virtual examples (NVE)

Figure 5 reports the language recognition performance by incorporating non-target virtual examples. In the experiments, n=2000, t varies from 0.1 to 0.4 and k is set proportional to t to ensure a non-target virtual example is comparable in length with the reference utterances (e.g. if t=0.1, k=10). The language recognition performance on the most of the open-set results are improved (except NVE 0.4 on 3s), while the NVE only helps some of the 10s tests in the closed-set conditions. This is reasonable as the NVEs are derived from reference utterances of different languages, the phonotactic patterns that are not belong to the target language may be included by NVE, which causes the performance drop of some closed-set tests.

Figure 5 also shows the virtual examples derived by a smaller selection parameter t gives better performance than those with large ones. When a larger t is used, a large proportion of the n-gram statistics from a same target language is preserved in the NVE. The is not desired as an NVE is labelled as negative samples in the language classifier training.

5.3. Incorporating both TVE and NVE

Table 2 compares the language recognition results of the baseline system and the proposed virtual example construction methods. The virtual examples are derived with comparable alteration parameters. Both close-set and open-set condition results are reported.

It is worth to note that the performances of the open-set tasks are improved by incorporating target virtual examples (TVE). It is attributed to the more generalized target language classifier obtained by incorporating TVE, the new classifier makes more accurate prediction on the target language, hence the false alarm rate is lowered.

The last row in Table 2 shows the results of a fusion system, the relative improvements to the baseline are shown in brackets. By incorporating both the target and non-target virtual examples, the language recognition performances are further improved across all the test conditions.

6. Conclusions and future works

This paper proposes a method to derive both target and non-target virtual examples for phonotactic language recognition. Our experiment results show that the target virtual examples improve the language recognition performance, especially on short duration test sets. The non-target virtual examples have small improvements to the open-set language recognition. Combining both of them improves the language recognition in all test conditions.

The proposed virtual example construction methods provide an alternative solution for the problem of lacking enough training data for language recognition.

In the future works, we would like to improve the non-target virtual example construction method for open-set condition. We also want to extend virtual example construction method in acoustic feature based language recognition.

7. References

[1] D. Yu, S. Wang, Z. Karam, and L. Deng, "Language recognition using deep-structured conditional random fields," in *ICASSP*, 2010, pp. 5030–5033.

[2] I. Lopez-Moreno1, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic language identification using deep neural networks," in *ICASSP*, 2014, pp. 5374–5378.

[3] M. Soufifar, S. Cumani, L. Burget, and J. H. Cernocky, "Discriminative classifiers for phonotactic language recognition with ivectors," in *ICASSP*, 2012, pp. 4853–4856.

[4] L. F. D'Haro, R. Cordoba, M. A. Caraballo, and J. M. Pardo, "Low-resource language recognition using a fusion of phoneme posteriorgram counts, acoustic and glottal-based i-vectors," in *ICASSP*, 2013, pp. 6852 – 6856.

[5] F. S. Richardson and W. M. Campbell, "Nap for high level language identification," in *ICASSP*, 2011, pp. 4392–4395.

[6] R. Tong, B. Ma, H. Li, and E. S. Chng, "Selecting phonotactic features for language recognition," in *Interspeech*, 2010, pp. 737–740.

[7] F. S. Richardson and W. M. Campbell, "Language recognition with discriminative keyword selection," in *ICASSP*, 2008, pp. 4145 – 4148.

[8] W. Liu, W.-Q. Zhang, and J. Liu, "Selection of negative pool for pr-svm in language recognition," in *Audio, Language and Image Processing (ICALIP)*, 2012, pp. 961–965.

[9] H. Kwan and K. Hirose, "Unknown language rejection in language identification system," in *ICSLP 96*, 1996, pp. 1776–1779.

[10] M. Penagarikano, A. Varona, M. Diez, L. J. Rodriguez-Fuentes, and G. Bordel, "Study of different backends in a state-of-the-art language recognition system," in *Interspeech*, 2012, pp. 2049–2052.

[11] M. F. Benzeghiba, J.-L. Gauvain, and L. Lamel, "Gaussian backend design for open-set language detection," in *ICASSP*, 2009, pp. 4349–4352.

[12] M. McLaren, A. Lawson, Y. Lei, and N. Scheffer, "Adaptive gaussian backend for robust language identification," in *Interspeech*, 2013, pp. 84–88.

[13] P. A. Torres-Carrasquillo, E. Singer, T. Gleason, A. McCree, D. A. Reynolds, F. Richardson, and D. Sturim, "The mitll nist lre 2009 language recognition system," in *ICASSP*, 2010, pp. 4994–4997.

[14] H. Li, K. A. Lee, and B. Ma, "Spoken language recognition: From fundamentals to practice," *Proceedings of the IEEE*, vol. 101, pp. 1136–1159, 2013.

[15] S. Ganapathy, M. Omar, and J. Pelecanos, "Unsupervised channel adaptation for language identification using co-training," in *ICASSP*, 2013, pp. 6857–6861.

[16] B. Scholkopf, C. Burges, and V. Vapnik, "Incorporating invariances in support vector learning machines," in *ICANN96*, 1996, pp. 47–52.

[17] N. Jaitly and G. E. Hinton, "Vocal tract length perturbation (vtlp) improves speech recognition," in *International Conference on Machine Learning (ICML)*, 2013.

[18] N. Kanda, R. Takeda, and Y. Obuchi, "Elastic spectral distortion for low resource speech recognition with deep neural networks," in *ASRU*, 2013, pp. 309–314.

[19] P. Niyogi, F. Girosi, and T. Poggio, "Incorporating prior information in machine learning by creating virtual examples," *Proceedings of the IEEE*, vol. 86, pp. 2196 – 2209, 1998.

[20] M. Sassano, "Virtual examples for text classification with support vector machines," in *EMNLP*, 2003, pp. 208–215.

[21] J. L. Gauvain, A. Messaoudi, and H. Schwenk, "Language recognition using phone lattices," in *Interspeech*, 2004, pp. 208–215.

[22] R. Tong, B. Ma, H. Li, and E. S. Chng, "A target-oriented phonotactic front-end for spoken language recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 17, pp. 1335–1347, 2009.

[23] Y. K. Muthusamy, N. Jain, and R. A. Cole, "Perceptual benchmarks for automatic language identification," in *ICASSP*, 1994, pp. 333–336.

[24] P. Matejka, P. Schwarz, J. Cernocky, and P. Chytil, "Phonotactic language identification using high quality phoneme recognition," in *Interspeech*, 2005, pp. 2237–2240.

[25] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, "The htk book version 3.0." *Cambridge University*.

[26] A. Stolcke, "Srilm - an extensible language modeling toolkit," in *ICSLP*, 2002.

[27] T. Joachims, "Learning to classify text using support vector machines," *Dissertation, Kluwer*, 2002.

[28] "Nist lre-2009 evaluation plan, 2009," http://www.itl.nist.gov/iad/mig/tests/lre/2009/LRE09_EvalPlan_v6.pdf.