A HYBRID PHONOTACTIC LANGUAGE IDENTIFICATION SYSTEM WITH AN SVM BACK-END FOR SIMULTANEOUS LECTURE TRANSLATION

Michael Heck, Sebastian Stüker, and Alex Waibel

Institute for Anthropomatics
Karlsruhe Institute of Technology
Karlsruhe, Germany
michael.heck@student.kit.edu, sebastian.stueker@kit.edu, alexander.waibel@kit.edu

ABSTRACT

In this paper we describe our work in constructing a language identification system for use in our simultaneous lecture translation system. We first built PPR and PPRLM baseline systems that produce score-fusing language cue feature vectors for language discrimination and utilize an SVM back-end classifier for the actual language identification. On our bi-lingual lecture tasks the PPRLM system clearly outperforms the PPR system in various segment length conditions, however at the cost of slower run-time. By using lexical information in the form of keyword spotting, and additional language models we show ways to improve the performance of both baseline systems. In order to combine the faster run-time of the PPR system with the better performance of the PPRLM system we finally built a hybrid of both approaches that clearly outperforms the PPR system while not adding any additional computing time. This hybrid system is therefore our choice for the use in the lecture translation system due to its faster run-time and good performance.

***Index Terms*—** language identification, support vector machines, speech translation, lecture translation

1. INTRODUCTION

Automatic language identification (LID) is the task of automatically identifying the language used by an unknown speaker for voicing an utterance [1]. Among all established LID approaches, phone—or more precisely phonotactics—based techniques are the most popular ones, being able to do robust identification on sufficiently short input sequences even for real-time demands [1, 2]. Phonotactic approaches utilize phone decoders and operate on the streams of symbol sequences produced by them. Usually, a language decoder scores these sequences, given a set of potential target languages. Based on these scores, a back-end classifier then performs the actual language classification.

In this paper we describe our work in implementing an LID system for use in our simultaneous lecture translation system [3]. By augmenting the translation system with an LID system we plan to ease its use by eliminating the need for manually selecting the input language, and potentially being able to detect longer lasting switches in the the speakers' language during a lecture.

We first implement two baseline LID systems—a *parallel phone recognition* (PPR) system and a *PPR followed by language modeling* (PPRLM) system[4]—and evaluate their performance on the lecture translation task, consisting of a bilingual classification task between English and German lecture snippets, in 5 different snippet length conditions: 30s, 20s, 10s, 5s and 3s. Both systems make use of an SVM classifier back-end. We further show improvements by modifications to these two baseline systems which consist of the use of a rudimentary key-word spotting technique for the PPR system, and the use of additional, slightly varied language models for the PPRLM system.

With respect to the requirements for use in our lecture translation system we then propose a hybrid system of both approaches that combines the faster run-time of the PPR approach with the higher classification accuracy of the PPRLM approach.

2. RELATED WORK

Similar to [5] we also decided to use an SVM back-end classifier for a PPRLM system, conducting a fusion of all language model scores, which results in a error reduction of 29%, compared to a baseline system without back-end classifiers. Similar to [6], which unlike us used a GMM back-end classifier, we also make use of "score vectors" in a bi-lingual PPRLM framework, comprising several features for language classification. Additional to PPRLM language model scores, a "differential acoustic score" calculated from the phone recognizer generated scores was used in [6] and gives a 2% error reduction. Unlike in other work, e.g. [7], we do not use phone tokenizers that share a common set of acoustic models. The choice of an SVM back-end classifier is also motivated by experiments from [8] that demonstrated the higher performance of an SVM over a GMM and an ANN back-end classifier. Similar to [9], for each language we take into account the N most frequent words for modeling additional linguistic information.

3. PHONE RECOGNIZERS

The phone recognizers for our LID systems were realized with the help of the Janus Recognition Toolkit (JRTK) which features the IBIS single pass decoder [10]. We used the acoustic models of the German and English LVCSR systems that we had trained for the 2010 Quaero Speech-to-Text Evaluation [11]. We re-trained them without vocal tract length normalization and left out the discriminative training step. In the acoustic model phonemes are modelled by context-dependent three-state left-to-right HMMs without skip state, that have been clustered into generalized quinphones with the help of a CART decision tree. The German phone set consists of 45 phonemes, the English of 52 phonemes, including noise and silence phonemes. Each model uses Gaussian Mixture Models with diagonal covariance matrices and up to 128 components for calculating the emission probability with 16.000 distributions over 6.000 codebooks. The model was trained on broadcast news data, European

Parliament sessions and manually transcribed podcasts and similar sources collected from the World Wide Web—275h of training data for the German model and 309h for the English model.

In order to obtain phoneme recognizers from the LVCSR systems we replaced word based dictionary of each decoder by a dictionary comprising the respective phone set.

4. BASELINE SYSTEMS

4.1. Parallel Phone Recognition (PPR)

In the PPR approach, phonotactic knowledge is incorporated directly into the decoding process of a test utterance by means of a language model. Phonotactic, as well as acoustic-phonetic knowledge likewise contribute to each final recognition score. Phonotactic information is modelled by phone-based n-gram language models. In order to generate the training corpora for these n-gram models we converted the German and English acoustic model training corpus into phoneme sequences with the help of the forced alignments used for acoustic model training.

We then used the SRILM toolkit [12] to estimate phonetic trigram language models. As experiments in [13] showed that Witten-Bell discounting works best given an LID system utilizing phonotactic constraints in a PPRLM architecture, we also used it for estimating our language models. The phoneme recognizers obtained in this way achieved phoneme accuracies of 57% for German, and 53% for English.

The actual language classification is then performed by an SVM classifier back-end, that we trained with the LIBSVM toolkit [14]. The SVM operates on language cue feature vectors. A feature vector $\boldsymbol{v}_i = \{score^i_{DE}, score^i_{EN}\}$ for a test message i contains the scores of each phone recognizer. We decided to use the C-Support Vector Classification (C-SVC) formulation, as it is the original SVM fomulation [15], and fits our requirements. According to test runs on a small set of randomly selected data, the linear kernel type function $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T \boldsymbol{x}_j$ seemed to work best for us. During the classifier training, the linear penalty parameter C is determined automatically via a grid-search.

A full test run is performed by Viterbi decoding a test utterance by each phone recognizer, with the influence of the respective n-gram language model. Each decoder generates a log-likelihood recognition score. The scores are normalized by the utterance frame count and combined to a feature vector for SVM classification.

4.2. Parallel Phone Recognition followed by Language Modeling (PPRLM)

The PPRLM system's phone recognizers are in principal the same as for the PPR approach. The difference is, that the decoders no longer incorporate language models during the decoding process, rendering the decoding a mere tokenization. Instead, phonotactics are now modelled by language model back-ends. For each phone recognizer for the different target language, back-end language models for all target languages are trained. For our bi-lingual classification scenario, this results in two times two phone set dependent and language-specific phone-based n-gram language models (an English and a German language model in the phoneme set of the German phone recognizer, and both language models also in the phoneme set of the English phone recognizer).

In order to generate the training corpus for the English language model in the German phoneme set, and the corpus for the German language model in the English phoneme set, we tokenized the English acoustic model training data with the German phoneme recognizer, and the English training data with the German phoneme recognizer.

The SVM back-end classifier is trained exactly the same way as for the PPR framework. Each training message is decoded by the bank of phone tokenizers, resulting in two separate phone set dependent streams. Both streams are analyzed by the respective n-gram models: Let $\mathcal{T} = \{DE, EN\}$ be the set of target languages, $W_r = \{w_1^r, w_2^r, \ldots, w_m^r\}$ with $r \in \mathcal{T}$ be the phone sequence produced by the decoder front-end using the phone set of language r, and be $l \in \mathcal{T}$, then

$$\mathcal{L}(M|l) = \frac{1}{f} \cdot \sum_{r\in\mathcal{T}} \sum_{i=2}^{m} \log P(w_i|w_{i-1}, \lambda_l^r) \qquad (1)$$

is the joint language score for a given test message M, where $f = |\mathcal{T}|$ is the amount of phone tokenizers. This score is calculated with the help of the language models described above. Both scores of a particular target language l are averaged in the log domain, as both decoders are seen as working independently of each other.

Further, the joint language score is normalized in length by the number of frames in the message. The scores are stacked into a feature vector for SVM classification. A language cue feature vector $\boldsymbol{v}_i = \{\mathcal{L}(M|DE)^i, \mathcal{L}(M|EN)^i\}$ for every test message i is comprised by the joint language scores for each target language.

5. EXPERIMENTAL RESULTS

5.1. Test Data Base

Our test database consists of recorded lectures which have been manually transcribed. The German audio material mainly consists of lectures, that where given at the Karlsruhe Institute of Technology (KIT) and the Carnegie Mellon University (CMU), but also recordings of the state parliament of Baden-Württemberg, and speeches and various talks of ceremonial acts at KIT. The English data is comprised of lectures of the same type as for the German set and of TED Talks downloaded from the TED website[1]. The total amount of recorded data is 51 hours for German and 12 hours for English.

All recordings are available in 16 kHz and 16 bit quality, most of them were done with close-talk microphones. We generated five sets of audio segments with 30s, 20s, 10s, 5s and 3s average duration. The segments that we selected for these five sets had to meet the following constraints: Max. 20% of silence within the segment and max. 20% of foreign words per segment, according to the transcription data, and max. 20% variance in segment length, given the average.

The test data is split into a development set for training the SVM back-end classifiers and evaluation sets for every test condition, in a way that all sets are recording disjunct, and mostly speaker disjunct. However, one speaker had to appear in both the German and English sets, as well as both the development and evaluation set, since it provided a vast amount of the lecture data in the sets. Data of this speaker covers approximately 50% of the development set and, on average, 37% of each test set. However, the recording disjointedness is still maintained.

[1] http://www.ted.com

Training Set	Test Sets					
30s	30s	20s	10s	5s	3s	
1856	1715	2436	3959	5224	5207	DE
523	387	532	917	1325	1453	EN

Table 1. Fragmentation of the database.

5.2. Baseline Results

Table 2 shows the results for the baseline PPR and PPRLM systems. The PPRLM system outperforms PPR, regardless of the segment length category. However, with decreasing test messages length, the loss in performance for PPR system is relatively smaller than for the PPRLM system. This might be due to the fact that phone recognizer scores are be more robust to segment length, compared to the phonotactic scores delivered by language model back-ends, which need a sufficient amount of statistically relevant data for proper probability estimation.

System	30s	20s	10s	5s	3s
PPR baseline	91.1	90.9	89.4	87.7	86.5
+ Keyword spotting	96.9	96.6	93.8	90.5	87.0
PPRLM baseline	99.7	99.7	98.8	96.0	92.0
+ cleaned LMs	99.8	99.8	99.0	96.3	92.4
PPRLM & PPR hybrid	97.7	97.7	96.4	93.2	91.2

Table 2. Identification accuracy (in %) of the tested systems.

5.3. Keyword Spotting for PPR

In a next step we improved the PPR baseline system by incorporating rudimentary language-specific lexical knowledge. The assumption is, that a test message W_l in language l would generate more phoneme sequences resembling common words of language l, than a test message in a language $l' \neq l$. We generated a list of the 100 most common words per language, using existent word frequency lists, which are computed upon corpora of transcribed TV and movie recordings [16]. The corpora comprise 25 mio words for German and 29 mio words for English. Pronounciations and pronounciation alternatives were extracted from large dictionary files which have also been used for the 2010 Quaero evaluation system [11]. The lists were cleaned manually, e.g. by deleting entries consisting of single phones only.

We tested on two variants of applying these lists. The first is the use of generic n-gram language models, computed upon the word lists, where the keywords have a very high count (same for all words), and the uni-grams slip in with counts of 1. The models were applied as back-ends for the phone recognizers, which resulted in a performance gain for all but the shortest test segment category. A more straight-forward approach yielded more promising results: A phone sequence parser checks a recognizer hypothesis for sequences listed in the keyword pronounciation dictionary. The dictionary search runs sorted by phone sequence length, i.e. a phone already associated to a keyword cannot be associated again to a second keyword. Phones assumed to belong to a keyword are marked as such. Best language discrimination results were obtained by computing a keyword-phones to hypothesis length ratio. The identification accuracy significantly increased for longer test messages, whereas the performance of the short-time tests remained almost the same. The combination of both, the generic language model back-ends and the latter expansion, yielded an overall best performance.

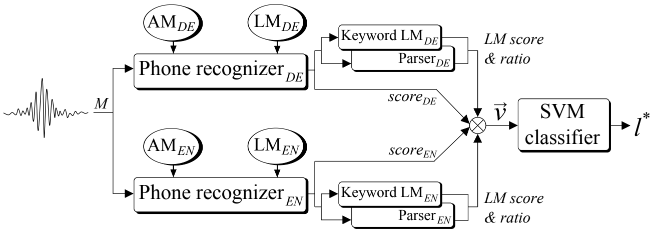

Fig. 1. Schematic of the improved PPR system.

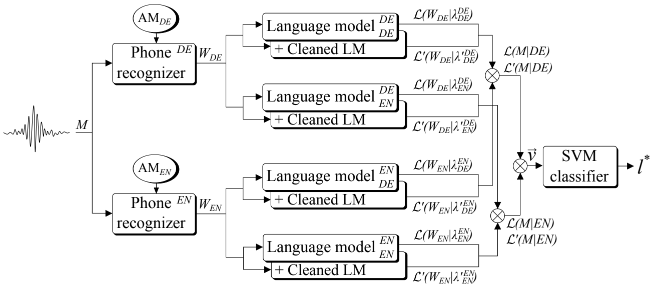
AM_{DE}
Phone recognizer ^{DE}
W_{DE}
Language model $^{DE}_{DE}$
+ Cleaned LM
$\mathcal{L}(W_{DE}|\lambda^{DE}_{DE})$
$\mathcal{L}'(W_{DE}|\lambda'^{DE}_{DE})$
$\mathcal{L}(M|DE)$
$\mathcal{L}'(M|DE)$
Language model $^{DE}_{EN}$
+ Cleaned LM
$\mathcal{L}(W_{DE}|\lambda^{DE}_{EN})$
$\mathcal{L}'(W_{DE}|\lambda'^{DE}_{EN})$
M
$\vec{v}$
SVM classifier
l^*
AM_{EN}
Phone recognizer ^{EN}
W_{EN}
Language model $^{EN}_{DE}$
+ Cleaned LM
$\mathcal{L}(W_{EN}|\lambda^{EN}_{DE})$
$\mathcal{L}'(W_{EN}|\lambda'^{EN}_{DE})$
$\mathcal{L}(M|EN)$
$\mathcal{L}'(M|EN)$
Language model $^{EN}_{EN}$
+ Cleaned LM
$\mathcal{L}(W_{EN}|\lambda^{EN}_{EN})$
$\mathcal{L}'(W_{EN}|\lambda'^{EN}_{EN})$

Fig. 2. Schematic of the improved PPRLM system.

5.4. Supplementary Language Models for PPRLM

During system implementation, we experimented with several variations of back-end language models. One attempt was to clean the corpora of all non-phone tokens, such as noise tokens, silence and filler tags, before language model generation. This variant led to comparable results compared to the initial system. This is not surprising, as the phone streams are cleaned of non-phone tokens, before they are processed by the back-end language models. However, using both sets of language models, resulting in two separate language scores, slightly improved the identification accuracy. The modified PPRLM framework achieves the best overall performance of our tested LID architectures. It yields an identification accuracy of 99.8% on 30s average test messages, and 92.4% on the average 3s category.

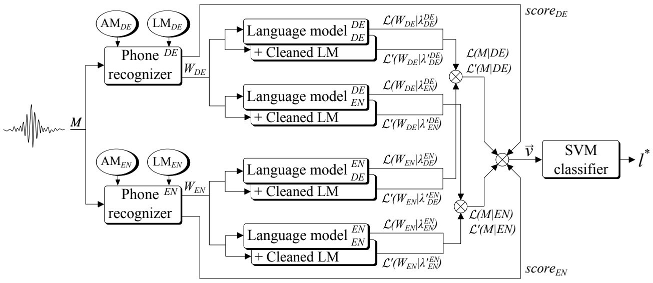
$score_{DE}$
AM_{DE}
LM_{DE}
Phone recognizer ^{DE}
W_{DE}
Language model $^{DE}_{DE}$
+ Cleaned LM
$\mathcal{L}(W_{DE}|\lambda^{DE}_{DE})$
$\mathcal{L}'(W_{DE}|\lambda'^{DE}_{DE})$
$\mathcal{L}(M|DE)$
$\mathcal{L}'(M|DE)$
Language model $^{DE}_{EN}$
+ Cleaned LM
$\mathcal{L}(W_{DE}|\lambda^{DE}_{EN})$
$\mathcal{L}'(W_{DE}|\lambda'^{DE}_{EN})$
M
$\vec{v}$
SVM classifier
l^*
AM_{EN}
LM_{EN}
Phone recognizer ^{EN}
W_{EN}
Language model $^{EN}_{DE}$
+ Cleaned LM
$\mathcal{L}(W_{EN}|\lambda^{EN}_{DE})$
$\mathcal{L}'(W_{EN}|\lambda'^{EN}_{DE})$
$\mathcal{L}(M|EN)$
$\mathcal{L}'(M|EN)$
Language model $^{EN}_{EN}$
+ Cleaned LM
$\mathcal{L}(W_{EN}|\lambda^{EN}_{EN})$
$\mathcal{L}'(W_{EN}|\lambda'^{EN}_{EN})$
$score_{EN}$

Fig. 3. Schematic of the proposed hybrid system.

5.5. PPRLM & PPR Hybrid

In order to combine the fast run-time of the PPR system, which is due to the use of a language model during the phone recognition, with the better classification accuracy of the PPRLM system, we constructed a hybrid system of both approaches. For that we used the phone recognizers of the PPR framework, followed by the unmodified PPRLM language model back-ends: Each phone recognizer generates a best hypothesis by means of acoustic and phono-

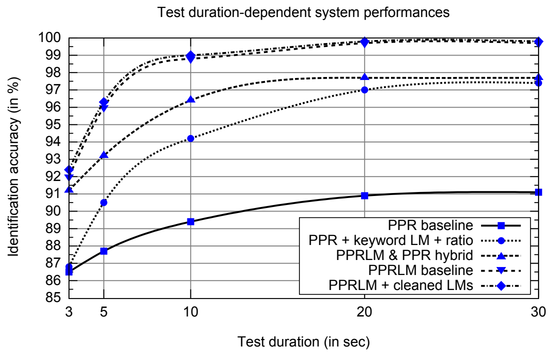

Fig. 4. System performances dependent on the test duration.

tactic scores. The phone streams are scored by the respective phone-dependent language models. A language cue feature vector for SVM processing contains both, the averaged language-specific scores and the phone recognizer scores.

Compared to the baseline PPR system, a significant performance boost is observable for all test conditions. However, the final PPRLM system remains unmatched. Considering the performance in terms of run-time, the hybrid system beats our best PPRLM framework, given the fact, that the phone recognition, which is faster in the PPR framework, consumes most of the computation time, whereas the language score computation, as well as the SVM classification run much faster. Real time factor analyses revealed, that the phone recognizers of the PPRLM system, which do not use language models during decoding, need about 20%-40% more computation time than the recognizers from the PPR framework.

So, with the hybrid system, it is now possible to have an LID system that runs as fast as a PPR system, but gives significantly better performance, as the PPRLM-like language model back-ends seem to compensate for the weaknesses of the pure PPR approach on short-duration testing. Hence, our proposed PPRLM & PPR hybrid outperforms the enhanced PPR system with a relative increase of 5% in accuracy on the 3s test condition.

6. CONCLUSION

In this paper we have described our LID systems for use in our simultaneous lecture translation system. We have started with two baseline systems—a PPR and a PPRLM system—of which the PPRLM system performed better on our lecture, bi-lingual test set. We were able to significantly boost the performance of the PPR system by applying keyword spotting techniques, while still not matching the performance of the PPRLM system, however. We were able to slightly improve the PPRLM performance by doubling the amount of back-end language models, using cleaned data for training a second model set. In order to combine the faster run-time of the PPR system, which is 30% faster than the PPRLM system, with the better performance of the PPRLM system, we designed a hybrid of both approaches. The hybrid system significantly outperforms the PPR approach, even on the shortest segments in our test, without adding additional run-time on top of it, thus making it a good candidate for use in our simultaneous lecture translation systems as it combines fast run-time with good performance.

7. ACKNOWLEDGEMENTS

‘Research Group 3-01’ received financial support by the ‘Concept for the Future’ of Karlsruhe Institute of Technology within the framework of the German Excellence Initiative.

8. REFERENCES

[1] Martine Adda-Decker, “Language identification,” in *Spoken Language Processing*, pp. 279–320. ISTE, 2010.

[2] Jirí Navrátil, “Automatic language identification,” in *Multilingual Speech Processing*, pp. 233–272. Academic Press, Burlington, 2006.

[3] Christian Fügen, *A System for Simultaneous Translation of Lectures and Speeches*, Ph.D. thesis, Universität Karlsruhe, 2009.

[4] M.A. Zissman, “Comparison of four approaches to automatic language identification of telephone speech,” *Speech and Audio Processing, IEEE Transactions on*, vol. 4, no. 1, pp. 31, 1996.

[5] Yan Deng and Jia Liu, “Automatic language identification using support vector machines and phonetic n-gram,” in *Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on*, 2008, pp. 71 –74.

[6] R. Córdoba, L. F. D’haro, R. San-segundo, J. Macías-guarasa, F. Fernández, and J. C. Plaza, “A multiple-gaussian classifier for language identification using acoustic information and PPRLM scores,” 2006.

[7] Rong Tong, Bin Ma, Haizhou Li, Engsiong Chng, and Kong-Aik Lee, “Target-aware language models for spoken language recognition,” in *INTERSPEECH*. 2009, pp. 200–203, ISCA.

[8] Hongbin Suo, Ming Li, Tiantan Liu, Ping Lu, and YongHong Yan, “The design of backend classifiers in pprlm system for language identification,” in *Natural Computation, 2007. ICNC 2007.*, 2007, vol. 1, pp. 678 –682.

[9] Driss Matrouf, Martine Adda-Decker, Lori Lamel, and Jean-Luc Gauvain, “Language identification incorporating lexical information,” in *ICSLP*. 1998, ISCA.

[10] Hagen Soltau, Florian Metze, Christian Fügen, and Alex Waibel, “A one-pass decoder based on polymorphic linguistic context assignment,” 2001.

[11] Sebastian Stüker, Kevin Kilgour, and Jan Niehues, “Quaero speech-to-text and text translation evaluation systems,” in *High Performance Computing in Science and Engineering ’10*, pp. 529–542. Springer Berlin Heidelberg, 2011.

[12] Andreas Stolcke, “SRILM-an extensible language modeling toolkit,” in *In Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, 2002, pp. 901–904.

[13] L. Wang, E. Ambikairajah, and E. H. C. Choi, “Multi-lingual phoneme recognition and language identification using phonotactic information,” in *ICPR*, 2006, pp. IV: 245–248.

[14] Chih-Chung Chang and Chih-Jen Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[15] Corinna Cortes and Vladimir Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, pp. 273–297, 1995.

[16] Wiktionary, “Wiktionary:frequency lists - wiktionary,” 2011, http://en.wiktionary.org/w/index.php?title=Wiktionary:Frequency_lists&oldid=13554525.