Language identification of individual words with Joint Sequence Models

Oluwapelumi Giwa and Marelie H. Davel

Multilingual Speech Technologies, North-West University, Vanderbijlpark, South Africa.

oluwapelumi.giwa@gmail.com, marelie.davel@gmail.com

10.21437/Interspeech.2014-344

Abstract

Within a multilingual automatic speech recognition (ASR) system, knowledge of the language of origin of unknown words can improve pronunciation modelling accuracy. This is of particular importance for ASR systems required to deal with code-switched speech or proper names of foreign origin. For words that occur in the language model, but do not occur in the pronunciation lexicon, text-based language identification (T-LID) of a *single word in isolation* may be required. This is a challenging task, especially for short words. We motivate for the importance of accurate T-LID in speech processing systems and introduce a novel way of applying Joint Sequence Models to the T-LID task. We obtain competitive results on a real-world 4-language task: for our best JSM system, an F-measure of 97.2% is obtained, compared to a F-measure of 95.2% obtained with a state-of-the-art Support Vector Machine (SVM).

Index Terms: text-based language identification, joint sequence models, multilingual speech recognition

1. Introduction

Words, phrases and names are often used across language boundaries in multilingual settings. Especially for minority languages, such *code-switching* with a dominant language can become an intrinsic part of the language itself [1]. Automatic speech recognition (ASR) systems are required to deal with various types of words of foreign origin. For example, automated call routing systems or voice-driven navigation systems process proper names and foreign words: these tend to have pronunciations that are difficult to predict [2]. Knowing the language of origin of such words, can improve modelling accuracy [3, 4]. As these categories of words (proper names, foreign words) can be important content terms in an utterance [5], there is a need to handle them carefully.

In order to be able to model code-switched words through language-specific pronunciation or acoustic models, it becomes necessary to be able to identify the language of origin of words *in isolation*, that is, a single word from one language may be embedded in a matrix sentence of a second language. For minority languages, reliable word lists of substantial size can be surprisingly difficult to obtain, and text-based language identification (T-LID) systems must be able to generalise from fairly small corpora to be useful in practical systems. Much of the research in the T-LID field have been performed on running text (see [6] for an overview), but several studies have focused on identifying the language of origin of short text samples. Good results have been obtained using conventional statistical methods, such as n-gram based SVMs (Support Vector Machines) [7, 8] or n-gram based Naïve Bayes (NB) classification [7].

When words are considered in isolation, the task becomes more difficult, with shorter words (3 or 4 characters) retaining very little language-discriminative information. Studies on isolated word T-LID applied SVMs [8, 7], compression techniques [9] and Naïve Bayes classification [7]. SVMs still perform well in this task domain [7], but can be time-consuming to train and optimize, especially when large training corpora are used and multiple language classification is required.

At the same time, grapheme-to-phoneme (G2P) conversion techniques – aimed at predicting the pronunciation of a word from its orthographic form – have matured significantly during the past decade. Specifically, Joint Sequence Models (JSMs) have become a well-utilised method for pronunciation prediction [10]. (Also see [10, 11, 12] for reviews of general G2P techniques). We investigate the applicability of JSMs to the T-LID task, and analyse the comparative performance that can be obtained when applying JSMs, rather than the better-known SVM classifiers (which we use as our baseline classifier).

Specifically, we consider a four-language South African task (Afrikaans, English, isiZulu and Sesotho) and a data set for which semi-comparable baselines are available. In practice, the language classification is required for a directory enquiries application, building on the work described in [13]. In this paper, we describe how JSMs can be applied to the T-LID task, and demonstrate factors that influence identification accuracy.

2. Joint-Sequence Models

Joint Sequence Models (JSMs) [10] are based on the concept of 'graphones'. Each graphone consists of a sequence of graphemes linked to a sequence of phonemes modelled as a single unit. Both the graphone inventory and m-th order conditional probabilities (of one graphone following the preceding $m-1$ graphones) are estimated from training data. During pronunciation prediction, all possible co-segmentations (ways in which the word-pronunciation pair can be constructed from available graphones) are considered, and the pronunciation with the highest probability, when summing over all relevant co-segmentations, is produced.

Before we describe our approach to using JSMs for LID (Section 2.3), we first review the JSM learning model (Section 2.1) and the main parameter choices that influence model performance (Section 2.2). For more detail on JSMs, see [10].

2.1. Learning model

A baseline graphone set is deduced from the training set using an automatic alignment process, and these co-segmentations optimized using maximum likelihood (ML) estimation. Specifically, the expectation maximisation (EM) algorithm is used to improve ML estimates of the training data in an iterative fashion. The joint probability, $p(g, \varphi)$, for the most probable joint-segmentation for a particular sequence of graphemes can be

modelled by summing over all relevant co-segmentations:

$$p(g, \varphi) = \sum_{q \in S(g,\varphi)} p(q_1, ..., q_K) \quad (1)$$

where φ represents the phoneme sequence, g represents the grapheme sequence, q represents the graphone sequence, K is the length of the graphone sequence and S represents the set of all joint segmentations. The probability distributions of eq. 1 can be approximated:

$$p(q_1^K) \approx \prod_{i=1}^{K+1} p(q_i \mid q_{i-1}, ..., q_{i-M-1}) \quad (2)$$

where K is the length of the graphone sequence and M denotes the order of the multigram model employed.

2.2. Parameter definition

The following main algorithmic choices influence performance:

- *Model initialisation* can either use (1) a flat probability distribution, where all multigrams are initialised to an identical initial probability, or (2) initialise based on counts, where initialisation is based on the number of occurrences of a graphone. Both these methods are subjected to an important parameter: the graphone length constraint (L), which must be manually set. (See below.)
- Higher *m-gram model orders* produce better accuracy and gradually reach an asymptotic level.
- *Graphone length*: Graphone length (l_{min}, l_{max}, r_{min}, r_{max}) can be restricted during training. These values indicate the minimum and maximum number of graphemes (l for left-hand side) and phonemes (r for right-hand side) respectively allowed per graphone unit. These parameters have a significant effect on the size of the graphone inventory produced.
- *Estimating discount*: The standard JSM implementation uses Modified Kneser-Ney to better handle unseen data and prevent overfitting. Discount parameters are estimated on held-out data, across all model orders, one model order at a time. After parameter estimation, a fold-back strategy can be used to return held-out data back to the training set.

2.3. Using JSMS for LID

In this section, we explain our approach to using JSMs for LID. The T-LID task is recast as a 'pronunciation learning' task, with each phone replaced by a language identifier, repeated for the length of the word. For example, the English word 'queen' or Sesotho word 'dumela' will be represented in the lexicon as:

```
#queen#     E E E E E E E
#dumela#    S S S S S S S S
```

where *E* represents English, *S* represents Sesotho and # represents a word boundary marker (as found useful for Naive Bayes classification [7]). In effect, we are therefore modelling grapheme chunks rather than graphone chunks.

Given an input word, the task is to find the most likely source language. Once a string of 'phones' (in our case, language identifiers) has been predicted, one of two simple voting schemes is used to select the final language of origin:

1. Majority voting: The language identifier observed most is selected. When voting is tied, the source language is considered to be 'Unknown'.
2. Log probability voting: For each language, the conditional probabilities of eq. 2 are multiplied (for that specific language), and the language with the highest probability is selected. In practice, log probabilities across graphone sequences are summed in order to reduce computational complexity.

3. Experimental set-up

We analyse the applicability of JSMs to T-LID within the context of a 4-language task. An SVM baseline is obtained to evaluate the JSM results against. The effect of using different JSM parameters is analysed; we pay specific attention to the interplay between training corpus size, word length and classification accuracy.

3.1. Data sets

The specific T-LID task involves 4 South African languages (English, Afrikaans, isiZulu and Sesotho). Training data was obtained from the *NCHLT-inlang* dictionaries [14]. Only the word lists were used, consisting of 15,000 unique words per language. During development, these were estimated to be frequent words based on available corpus counts. While the correctness of the original *NCHLT-inlang* word lists was verified using automated spell checkers and language practitioners [14], our analysis showed that the published lists still contained errors. A second round of (higher precision, lower recall) spell-checking was performed for the three languages for which this was possible, namely: Afrikaans, isiZulu and English. This resulted in an 'original' (from *NCHLT-inlang*) and 'spell-checked' (further refined here) list. In addition, a third set ('no_bilingual') was created containing no known bilingual words (all words that occurred across training corpora, were removed).

3.2. Partitioning of the data set

During experiments, training set sizes are limited artificially to investigate the effect of training corpus size: the required number of training samples are selected randomly per language. Training corpora are constructed by combining an equal number of words per language. All results are obtained through 4-fold cross-validation. Where parameter optimisation is required, a small (10%) development set is used, and folded back post training. Specified training set sizes include the development set, and indicate the number of training samples *per language*. For each data set from Section 3.1 (individually), test sets are kept constant across all experiments.

3.3. Evaluation metrics

We use standard precision, recall and the combined F-measure to evaluate performance per language. That is, if h represents correctly classified 'hits', f_r represents 'false rejects' and f_a represents 'false accepts', then we calculate precision $p = h/(h + f_a)$, recall $r = h/(h + f_r)$ and the F-measure $(2 * p * r)/(p + r)$ per language. To estimate a confidence interval, we calculate the standard error of the different measures across n folds ($\sigma/\sqrt{(n)}$) with σ the standard deviation of the specific measure considered.

4. Experiments and results

4.1. SVM baseline

In order to obtain a baseline result, we use the SVM implementation that produced the best T-LID results in [7]. Each training sample is represented by an N-dimensional vector, with N the number of unique n-grams observed. Each feature constitutes an n-gram index and n-gram value, where the value indicates the number of times that the specific n-gram is observed in the training sample, and index represents the n-gram position in N-dimensional space; the language identifier per sample acts as class label. As in [7], training data is scaled in the range of [0,1], a Radial Basis Function (RBF) kernel is used and n-gram lengths of 3 provide optimal results. While these n-grams seem small, this is a result of the test words, many of which are themselves quite short.

The RBF kernel requires that two hyper-parameters be optimised: the soft margin parameter (C) and kernel width ($gamma$). As motivated in [15], we set $gamma$ to the reciprocal of the sample dimensionality (with sample dimensionality ranging from approximately 4,500 to 6,500 in this case) and perform a grid search for C (in the range 10^{-4} to 10^{4}) using a development set. Results are shown in Table 1; all results are 4-fold cross validated. As expected, results on the spell-checked data is better than on the original set, and removing bilingual words makes the task a bit easier. Also, larger training sets improve accuracy, with additional gains expected for data sets larger than 12K.

Table 1: *F-measure achieved with SVM baseline for different training data sets. The confidence interval is based on estimated standard error.*

	$original$	$spell\_checked$	$no\_bilingual$
2K	90.85 ±0.21	91.10 ±0.48	91.51 ±0.49
4K	92.94 ±0.24	93.05 ±0.10	93.14 ±0.14
6K	93.50 ±0.14	93.62 ±0.20	93.81 ±0.21
8K	94.21 ±0.14	94.10 ±0.23	94.33 ±0.18
10K	94.59 ±0.11	94.82 ±0.08	94.85 ±0.07
12K	94.73 ±0.22	95.05 ±0.12	95.16 ±0.11

4.2. Initial JSM implementation

The training data is constructed and JSM models trained as described in Section 2.3. Models are initialised using counts, contexts are not constrained, and full EM is used during training. Discount parameters are optimized using a 10% hold-out set, and folded back later. Model order is increased until asymptotic performance is obtained (for all experiments at a model order of $M = 8$).

Fig. 1 displays the results obtained for the same data sets analysed in Section 4.1, when classified using the initial JSM implementation. Similar trends are observed as with SVM classification: performance increases without yet reaching an asymptote; the 'original' task is the most difficult, the 'no_bilingual' one the easiest. However, for all data sizes, the JSM implementation outperforms the SVM implementation: at 2K, an absolute increase of 3.32% is observed; at 12K, an absolute increase of 2.03%. This translates to a relative error reduction of approximately 40% over the different data sizes evaluated. Fig. 2 shows precision, recall and F-measure separately for the JSM system, achieved on the *no_bilingual* data set: the only data set we use from this point onwards. As the test sets are balanced across languages, little additional information is provided by considering precision and recall separately, and only the F-measure is considered from here onwards.

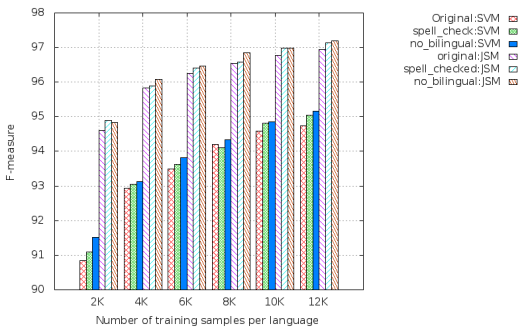

Figure 1: Comparing identification accuracy of (context-unconstrained) JSM and SVM across different data sizes, when trained and evaluated on 3 different data sets: $original$, $spell\_checked$ and $no\_bilingual$.

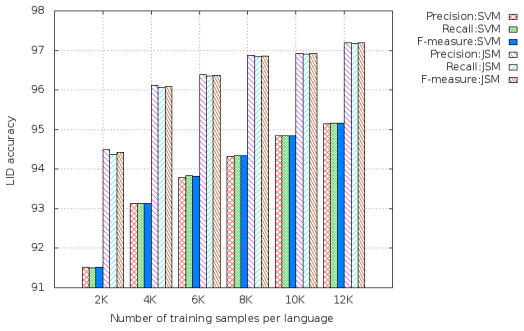

Figure 2: Precision, recall and F-measure of (context-unconstrained) JSM and SVM systems for different data sizes when trained and evaluated on the $no\_bilingual$ data set.

4.3. Effect of graphone length constraints

In [10], Bisani and Ney found that allowing graphones to have arbitrary length produced good G2P performance at low order models, but poorer performance at higher order models. Their best results were obtained when restricting graphemes (not graphones) to be singular (0 or 1) and allowing model order to increase unrestricted until asymptotic performance was reached. We therefore consider whether any benefits can be obtained from restricting graphone length.

Given the way the T-LID task has been framed (see Section 2.3), restricting either component of the graphone has the same effect as restricting both. That is, when restricting graphone length to x, we are in effect restricting both the left- and right-hand side to x. We use the (l_{min}, l_{max}, r_{min}, r_{max}) notation introduced in Section 3 and evaluate results when restricting graphone length to 1 or 2, respectively.

For the task at hand, we observe that, when varying l_{max} and r_{max}, results are comparable irrespective of whether l_{min} and r_{min} are set to 0 or 1. (This is not expected from pronunciation modelling tasks where a null grapheme or phoneme is required, but is specific to the way the LID task has been framed.) As using a minimum length value of 0 trains much slower than using a value of 1, we use 1 from here onwards.

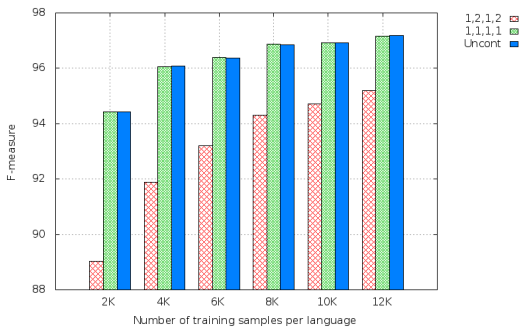

Figure 3: LID accuracy with different context constraints at different training data sizes.

Fig. 3 shows LID accuracy (using F-measure) across different data sizes for different graphone lengths. As can be seen, there is almost no difference between singular graphones (1,1,1,1) and the 'unconstrained' system. In fact, analysis of the graphone inventory produced shows that unconstrained graphone formation automatically produces singular graphones for this task. Restricting graphones to be of length 2, decreases accuracy. (Even though the algorithm is not prevented from forming singular graphones when restricting length of chunks to 2, sub-optimal chunk formation occurs early on in the training process.) In the remainder of the analysis, we allow unconstrained graphone formation.

4.4. Log probability voting

For the majority of words, tie resolution is not required. We are interested in understanding the extent to which log probability voting is required at all. For our smallest (2K) and largest (12K) training sets, we evaluate the number of errors per word length that do involve ambiguous labels (labels that can possibly be resolved using either majority or log probability voting). It is only a minority of errors (12.8% of errors at 2K, 4.2% of errors at 12K) that involve ambiguity: all other errors are tagged with a single (incorrect) language label. The number of errors that do contain ambiguous labels when training with the 12K set, is shown in Fig. 4. When using log probability voting, we observe a slight increase in performance for almost all test sample lengths (with only one decrease in performance, observed at length 7). The same trend is observed at 2K: while not seen at every word length individually, a small improvement in accuracy is observed overall.

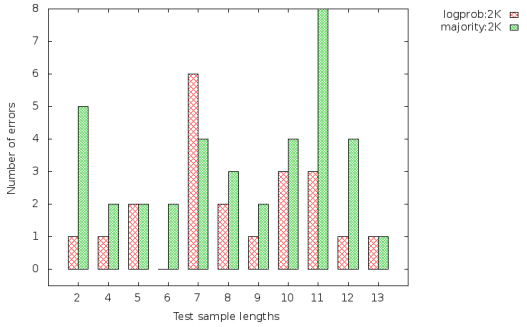

Figure 4: Analysis of errors using two different tie resolution strategies on largest training data set.

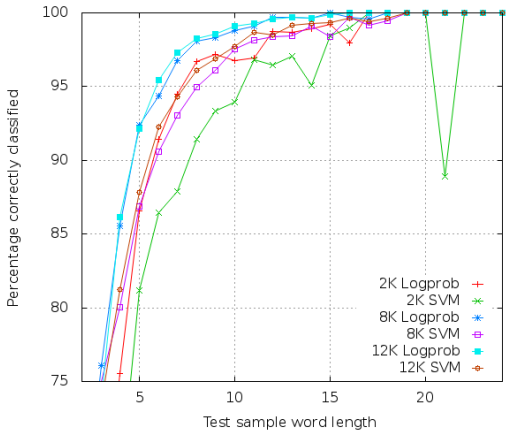

Figure 5: Comparative classification accuracy of words of different lengths for SVM baseline and JSM (with log probability voting) at 2K, 8K and 12K training set sizes.

4.5. Effect of training corpus size and word length

In this section, we compare the new method (unconstrained JSMs with log probability voting) with the baseline SVM classifier, given different training set sizes and test word lengths. Results are shown in Fig. 5. As expected, we observe that performance increases with both longer word lengths and additional training data. At longer word lengths (> 23 characters) both techniques achieve perfect classification accuracy. However, when considering shorter words, there is a clear advantage to using the JSM-based technique, across all training set sizes. It is interesting to note that the errors made by the two techniques show limited overlap: if required, additional wins may be possible by combining the two approaches.

5. Conclusion

For pronunciation modelling, accurate word-based T-LID is often required. We find that by recasting the T-LID task as a pronunciation modelling task, we can apply Joint Sequence Models with limited modification to the training process, and obtain competitive accuracies this way, especially on short test words. We evaluate two strategies for tie resolution, and find small gains in using the conditional log probabilities, rather than a simple majority voting scheme. An additional advantage of the new technique is that it is fast and simple to train, and does not require the extensive optimization that is typically part of the SVM training process.

For the specific task studied (a 4-language classification task for a South African directory enquiry system) we find that the proposed JSM-based technique reduces the SVM error with approximately 40% across a range of training data sizes evaluated. For the largest training set (48K words, 12K per language) an accuracy of 97.2% is obtained, which compares well with the best SVM classifier evaluated (at 95.2%).

6. Acknowledgements

This work was supported by the South African Department of Arts and Culture (DAC) and the National Research Foundation (NRF). Any opinion, findings and conclusions or recommendations expressed in this material are those of the author(s) and therefore neither DAC nor the NRF accepts any liability in regard thereto.

7. References

[1] T. I. Modipa, M. H. Davel, and F. de Wet, "Implications of Sepedi/English code switching for ASR systems," in *Proc. Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)*, Johannesburg, South Africa, 2013, pp. 64–69.

[2] B. Réveil, J.-P. Martens, and H. van den Heuvel, "Improving proper name recognition by adding automatically learned pronunciation variants to the lexicon," in *Proc. Language Resources and Evaluation Conference (LREC)*, Valletta, Malta, 2010, pp. 2149–2154.

[3] A. Font Llitjos and A. W. Black, "Knowledge of language origin improves pronunciation accuracy of proper names." in *Proc. INTERSPEECH*, Aalborg, Denmark, 2001, pp. 1919–1922.

[4] B. Réveil, J.-P. Martens, and B. Dhoore, "How speaker tongue and name source language affect the automatic recognition of spoken names," in *Proc. INTERSPEECH*, Brighton, UK, 2009, pp. 2995–2998.

[5] M. F. Spiegel, "Pronouncing surnames automatically," in *Proc. of the American Voice I/O Society (AVIOS) Conference*, 1985, pp. 109–132.

[6] G. R. Botha and E. Barnard, "Factors that affect the accuracy of text-based language identification," *Computer Speech & Language*, vol. 26, no. 5, pp. 307–320, 2012.

[7] O. Giwa and M. H. Davel, "N-gram based language identification of individual words," in *Proc. Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)*, Johannesburg, South Africa, 2013, pp. 15–21.

[8] A. Bhargava and G. Kondrak, "Language identification of names with SVMs," in *Proc. North American Chapter of the Association of Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Los Angeles, CA, 2010, pp. 693–696.

[9] A. Hategan, B. Barliga, and I. Tabus, "Language identification of individual words in a multilingual automatic speech recognition system," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, 2009, pp. 4357–4360.

[10] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[11] R. Damper, Y. Marchand, M. Adamson, and K. Gustafson, "A comparison of letter-to-sound conversion techniques for English text-to-speech synthesis," in *Proc. of the Institute of Acoustics (IOA)*, 1998, pp. 245–254.

[12] P. Taylor, "Hidden Markov models for grapheme to phoneme conversion." in *Proc. INTERSPEECH*, Lisbon, Portugal, 2005, pp. 1973–1976.

[13] O. Giwa, M. H. Davel, and E. Barnard, "A Southern African corpus for multilingual name pronunciation," in *Proc. Annual Symposium of the Pattern Recognition Association of South Africa (PRASA)*, Vanderbijlpark, South Africa, 2011, pp. 49–53.

[14] M. H. Davel, W. D. Basson, C. van Heerden, and E. Barnard, "NCHLT Dictionaries: Project Report," Multilingual Speech Technologies, North-West University, Tech. Rep., May 2013. [Online]. Available: https://sites.google.com/site/nchltspeechcorpus/home

[15] C. J. van Heerden, "Efficient training of Support Vector Machines and their hyperparameters," Ph.D. Thesis, North-West University, Potchefstroom, 2013.