A Dynamic Approach to the Selection of High Order N-Grams in Phonotactic Language Recognition

Mikel Penagarikano, Amparo Varona, Luis Javier Rodríguez-Fuentes, Germán Bordel

Software Technologies Working Group (http://gtts.ehu.es)
Department of Electricity and Electronics
University of the Basque Country
Leioa, Spain

email: mikel.penagarikano@ehu.es

ICASSP 2011, Prague, Czech Republic
May 24, 2011

GTTS Tecnologías Software

Outline

Motivation

- In Phonotactic Language Recognition, high-order n-grams are expected to contain more discriminant (language-specific) information.
- The number of n-grams grows exponentially as n increases.
- Due to robustness and computational bounds, most SVM-based phonotactic language recognition systems consider only low-order n-grams (up to $n = 3$).
- Dimensionality reduction techniques (PCA, LDA, etc.) cannot be directly applied to such a huge feature space.

GTTS Tecnologías Software

Background

Feature selection techniques have been previously used for high-order n-grams based Phonotactic Language Recognition:

- A wrapper/filter method was used to select the most discriminant $(n-1)$-grams (based on the SVM weigths) and expand them to get a *suboptimal* subset of n-grams.

  F. Richardson and W. Campbell, "Language recognition with discriminative keyword selection", in Proceedings of ICASSP, 2008.

Background

Feature selection techniques have been previously used for high-order n-grams based Phonotactic Language Recognition:

- A wrapper/filter method was used to select the most discriminant $(n-1)$-grams (based on the SVM weigths) and expand them to get a *suboptimal* subset of n-grams.

  F. Richardson and W. Campbell, "Language recognition with discriminative keyword selection", in Proceedings of ICASSP, 2008.

- The same wrapper/filter method was used along with two discriminative criteria: SVM separation margin and Chi-squared measure.

  Rong Tong, Bin Ma, Haizhou Li, and Eng Siong Chng, "Selecting phonotactic features for language recognition", in Proceedings of Interspeech, 2010.

In both cases, no improvement (or even degradation) was reported for features higher than 4-grams.

GTTS Tecnologías Software

Background

- In a previous work, we addressed a similar problem. We used *cross-decoder co-occurrences of phone n-grams* in Phonotactic Language Recognition, and the resulting feature space was very big.

  M. Penagarikano, A. Varona, L.J. Rodriguez-Fuentes, G. Bordel, "Using Cross-Decoder Co-Occurrences of Phone N-Grams in SVM-based Phonotactic Language Recognition", in Proceedings of Interspeech , 2010.

- **Key idea:** Build an sparse vector of expected feature counts using only the most frequent units.
- In the high-order n-grams scenario (huge feature space), even a simple frequency based feature selection becomes a challenge.

GTTS Tecnologías Software

Let's do frequency based feature selection

- Let h be the number of phonetic units of an acoustic decoder.

Let's do frequency based feature selection

- Let h be the number of phonetic units of an acoustic decoder.
- There exist h^n possible n-grams ($40^7 = 163.840.000.000$).

Let's do frequency based feature selection

- Let h be the number of phonetic units of an acoustic decoder.
- There exist h^n possible n-grams ($40^7 = 163.840.000.000$).
- Most of them will not be seen in training $\rightarrow$ forget them.

Let's do frequency based feature selection

- Let h be the number of phonetic units of an acoustic decoder.
- There exist h^n possible n-grams ($40^7 = 163.840.000.000$).
- Most of them will not be seen in training $\rightarrow$ forget them.
- Most of the seen features will have very low counts $\rightarrow$ forget them.

GTTS Tecnologías Software

Let's do frequency based feature selection

- Let h be the number of phonetic units of an acoustic decoder.
- There exist h^n possible n-grams ($40^7 = 163.840.000.000$).
- Most of them will not be seen in training $\rightarrow$ forget them.
- Most of the seen features will have very low counts $\rightarrow$ forget them.
- Simply select the most frequent features... SIMPLY??

GTTS Tecnologías Software

Let's do frequency based feature selection

- Let h be the number of phonetic units of an acoustic decoder.
- There exist h^n possible n-grams ($40^7 = 163.840.000.000$).
- Most of them will not be seen in training $\rightarrow$ forget them.
- Most of the seen features will have very low counts $\rightarrow$ forget them.
- Simply select the most frequent features... SIMPLY??
- Although its size being much less than h^n, the actual set of ngrams is too big even to store their cummulative counts.

GTTS Tecnologías Software

Let's do frequency based feature selection

- Let h be the number of phonetic units of an acoustic decoder.
- There exist h^n possible n-grams ($40^7 = 163.840.000.000$).
- Most of them will not be seen in training $\rightarrow$ forget them.
- Most of the seen features will have very low counts $\rightarrow$ forget them.
- Simply select the most frequent features... SIMPLY??
- Although its size being much less than h^n, the actual set of ngrams is too big even to store their cummulative counts.
- An estimation of the cummulative counts must be used instead.

GTTS Tecnologías Software

Estimation of cummulative counts

- Using the full training set Ω, build a table with cummulative counts of seen features.
- Periodically (every accumulated K counts), retain only those entries with counts higher than a given threshold τ (all the counts lower than τ are implicitly set to zero).
- The selection process is *suboptimal*, since many counts are discarded.
- To get the M most frequent features, K and τ must be tuned so that enough number of alive counts (at least, M) are kept at each update. Suitable values for K and τ produce tables with final sizes around $10 \times M$.

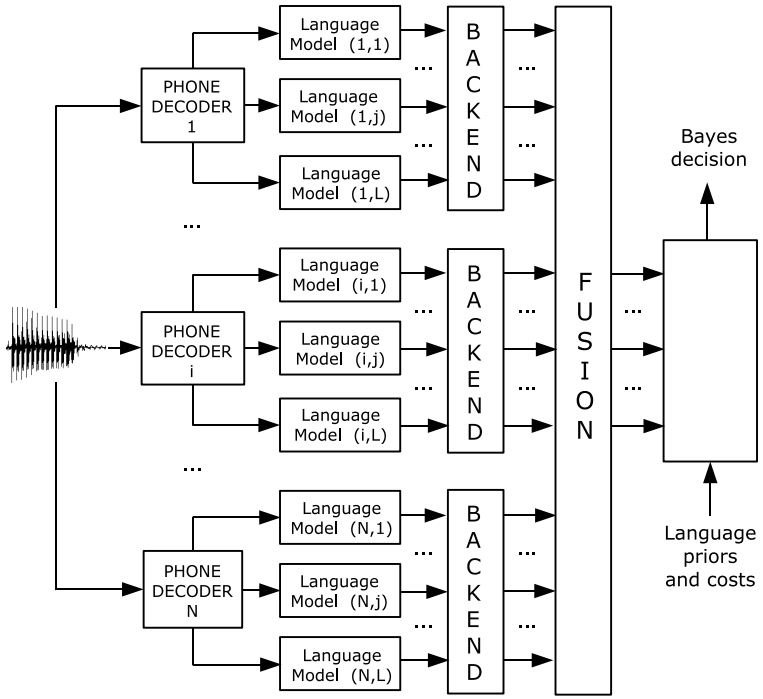
The proposed algorithm

table $\leftarrow \emptyset$
t $\leftarrow 0$
for $X \in \Omega$ **do**
    accumulate_counts(table,X)
    $t \leftarrow t +$ total_counts(X)
    if $t > K$ **then**
        t $\leftarrow 0$
        update(table,τ)
update(table,τ)
truncate(table,M)

Baseline Architecture

Common approach to phonotactic language recognition:

Phone Decoders + Lattices + SVM-based Language Models + Gaussian Backend + Linear Fusion



Training, development and test corpora

- Limited to those distributed by NIST to all LRE2007 participants
  - Call-Friend Corpus
  - OHSU Corpus provided by NIST for LRE05
  - development corpus provided by NIST for LRE07
- 10 conversations per language randomly selected for development purposes.
- Each development conversation was further split in segments containing 30 seconds of speech.
- Evaluation was carried out on the LRE07 evaluation corpus, specifically on the 30-second, closed-set condition.

GTTS Tecnologías Software

Software Components

Freely available software was used in all the stages

- **Phone Decoders:** The TRAPS/NN phone decoders developed by the Brno University of Technology (BUT) for Czech (CZ), Hungarian (HU) and Russian (RU).
- **Lattices:** HTK along with the BUT recipe.
- **SVM modeling:** *LIBLINEAR* (a fast linear-only version of libSVM). Modified by adding some lines of code to get the regression values (instead of class labels).
- **Gaussian Backend & Fusion:** *FoCal Multi-class* toolkit by Niko Brummer.

GTTS Tecnologías Software

Software Configuration

BUT TRAPS/NN CZ, HU & RU phone decoders

- Before decoding, an energy-based voice activity detector is applied to split and remove non-speech segments.
- Non-phonetic units: *int* (intermittent noise), *pau* (short pause) and *spk* (non-speech speaker noise) are mapped to *sil* (silent pause).
- Number of resulting phonemes: 43 (CZ), 59 (HU) and 49 (RU).
- BUT decoders are used to estimate posterior probabilities.

GTTS Tecnologías Software

Software Configuration

BUT TRAPS/NN CZ, HU & RU phone decoders

- Before decoding, an energy-based voice activity detector is applied to split and remove non-speech segments.
- Non-phonetic units: *int* (intermittent noise), *pau* (short pause) and *spk* (non-speech speaker noise) are mapped to *sil* (silent pause).
- Number of resulting phonemes: 43 (CZ), 59 (HU) and 49 (RU).
- BUT decoders are used to estimate posterior probabilities.

LIBLINEAR

- Phone lattices are modelled by means of Support Vector Machines
- SVM vectors consist of expected counts of phone n-grams extracted from the lattices, converted to frequencies and weighted with regard to their background probabilities as $w_i = \frac{1}{\sqrt{p(d_i|background)}}$.
- *One versus all* training.

From 3-grams to 4-grams

- The total number of features seen on training was about 2000000 for each decoder (1895778 for CZ, 2920755 for HU and 2300064 for RU).

From 3-grams to 4-grams

- The total number of features seen on training was about 2000000 for each decoder (1895778 for CZ, 2920755 for HU and 2300064 for RU).
- There was no need to use the feature selection algorithm.

From 3-grams to 4-grams

- The total number of features seen on training was about 2000000 for each decoder (1895778 for CZ, 2920755 for HU and 2300064 for RU).
- There was no need to use the feature selection algorithm.
- Not all of them appeared in the SVM vectors. The average size of the SVM vector was found to be about 70000.

From 3-grams to 4-grams

- The total number of features seen on training was about 2000000 for each decoder (1895778 for CZ, 2920755 for HU and 2300064 for RU).
- There was no need to use the feature selection algorithm.
- Not all of them appeared in the SVM vectors. The average size of the SVM vector was found to be about 70000.
- This system yielded 1.32% EER and $C_{LLR} = 0.22508$, meaning a relative improvement of 11% and 6%, respectively, compared to the trigram SVM system (baseline).

n-gram order	Feature Size	Average vector size	%EER	C_{LLR}
3	96416	13634	1.4932	0.23949
4	2372199	68627	1.3274	0.22508

GTTS Tecnologías Software

From 3-grams to 4-grams... with feature selection

n-gram order	M	Average vector size	%EER	C_{LLR}
3	96416	13634	1.4932	0.23949
4	2000000	68627	1.3274	0.22508
	1000000	66871	1.3269	0.22534
	500000	61963	1.3189	0.21874
	200000	49614	1.3417	0.22123
	100000	37635	1.3747	0.21861
	90000	35793	1.4229	0.22048
	80000	33754	1.4334	0.21997
	70000	31477	1.3768	0.22335
	60000	28931	1.3862	0.22197
	50000	26028	1.3536	0.22613
	40000	22689	1.3838	0.22834
	30000	18778	1.3676	0.22810
	20000	14076	1.4109	0.23404
	10000	8178	1.6077	0.25932
	5000	4507	1.6981	0.28028

GTTS Tecnologías Software

From 4-grams to n-grams... setup the algorithm

- We heuristically fixed $K = 10^6$ and $\tau = 10^{-5}$ to ensure that more than 2000000 features were kept at the end.
- $K = 10^6$ is equivalent to more than 2 hours of voiced audio.
- As n increases, the number of *live* n-grams decreases (Czech, $M = 100000$):

n-**gram**	**3**	**4**	**5**	**6**	**7**
live n-**grams**	69277	74733	18256	290	12

- 7-grams were used as higest order n-grams.

GTTS Tecnologías Software

From 4-grams to n-grams... results

n-gram order	M	Average vector size	%EER	C_{LLR}
3	96416	13634	1.4932	0.23949
4	100000	37635	1.3747	0.21861
	30000	18778	1.3676	0.22810
5	100000	41161	1.3267	0.22300
	30000	19195	1.3576	0.22613
6	100000	40823	1.3415	0.22366
	30000	19187	1.3671	0.23007
7	100000	39357	1.3451	0.22152
	30000	19119	1.3987	0.22973

GTTS Tecnologías Software

Summary

- A dynamic feature selection method has been proposed which allows to perform phonotactic SVM-based language recognition with high-order n-grams.
- Performance improvements with regard to a baseline trigram SVM system have been reported in experiments on the NIST LRE2007 database when applying the proposed algorithm to select the most frequent units up to 4-grams, 5-grams, 6-grams and 7-grams.
- The best performance was obtained when selecting the 100000 most frequent units up to 5-grams, which yielded 1.3267% EER (11.2% improvement with regard to using up to 3-grams).
- We are currently working on the evaluation of smarter selection criteria under this approach.

GTTS Tecnologías Software

Thank you!