Grapheme versus phoneme in Spanish keyword spotting and spoken term detection

Javier T., Dong W., Simon K. and Joe F.
CSTR Listen meeting
11-June-2007
Introduction

• Keyword Spotting: Search for a set of keywords in large audio repositories.
• LVCSR drawbacks: Lots of data for training. Vocabulary-dependent. Low speed.
• To deal with the OOV in LVCSR systems.
• Spoken term detection: The list of keywords to search for is unknown in the indexing (recognition) phase.
• Objective: High detection rate and low false alarms.
• Purposed: Indexing and retrieve of audio information.
State-of-art and motivation

• All of them used phoneme-based units as acoustic modelling.
• Spanish grapheme-based approach LVCSR in Killer et.al. 2003 works well in terms of WER.
State-of-art and motivation

• Grapheme and phoneme are very closed in Spanish.
• In english a 10% increase in WER using graphemes instead of phoneme for LVCSR. For spoken term detection the rate is very similar.
• It can be thought that in Spanish if similar WER is achieved for LVCSR, final rate for spoken term detection systems based on graphemes outperforms the one achieved with phonemes.
Acoustic models

- Monophone, monographeme, triphone and trigrapheme.
- 47 allophones in Spanish (Quillis, 1998) to model the stressed vowels, the nasals vowels and other effects according to the phonological rules.
- 28 graphemes in Spanish. All except “h” which does not have a phonetic representation in Spanish.
- Triphone: Cross-word and were state-clustered HTK’s standard decision tree method with phonetically-motivated questions.
- Trigrapheme: Using only questions about single graphemes (“singleton questions”) due to best performance in Killer et.al. 2003.
- HTK 3.4 used in the work. Hvite as decoder.
Approach 1: Hybrid architecture

- For Keyword Spotting.
- Vocabulary and corpus dependent.
- Advantages: Best final rates.
- Drawbacks: Slow speed and corpus dependent.

Scheme of hybrid architecture
Approach 1: Hybrid architecture

• Keyword Spotting: Pseudo N-gram by Kim et.al.2004. To evaluate the performance of the system changing the frequency of keywords and filler models. Pseudo 6-gram for monophone and monographeme and Pseudo 12-gram for triphone and trigrapheme.

• Lexical access: Fissore et.al. over LVCSR (1989). Run twice. One from the set of costs trained on the corrects keywords and one from the set of costs trained on the false alarms in the development set.

Retrieves the best match keyword (kw’) from the cost correct and its cost and the cost of the best keyword from the false alarms.
Approach 1: Hybrid architecture

Sequence of phones/graphemes: \( S = s_1, s_2, \ldots, s^T \)

Transcription of the keyword: \( W = w^1, w^2, \ldots, w^R \)

For each substring:

\[
G(r, t) = \begin{cases} 
G(r - 1, t - 1) + C(sub(w^r, s^t)) & \text{if } s^t \neq s^{t-1} \\
G(r, t - 1) + IC(w^r, s^t) & \text{if } s^t = s^{t-1} \\
G(r - 1, t) + C(del(w^r)) & \text{otherwise}
\end{cases}
\]

\[IC(w^r, s^t) = \begin{cases} 
C(ins(w^r, s^t)) & \text{if } s^t \neq s^{t-1} \\
CC(sub(w^r, s^t)) & \text{otherwise}
\end{cases}\]

- Confidence measure:
  
  Kw: keyword proposed by the keyword spotting module.
  
  Kw’: keyword proposed with the less G from the cost corrects.
  
  SC: G from the cost corrects for kw’.
  
  SFA: Less G from the cost of the false alarms.

\[
k_w = \begin{cases} 
\text{accept} & \text{if } kw = kw' \text{ and } (SFA - SC) \geq \alpha \\
\text{reject} & \text{otherwise}
\end{cases}
\]
Approach 2: 1-Best decoding

- For spoken term detection.
- Vocabulary independent and corpus dependent.
- Faster than hybrid architecture but poorer results are achieved.
Approach 2: 1-Best decoding

• 1-best output lexical access: Take slice windows over the output of the phone/grapheme decoder and apply Fissore algorithm (approach 1) from the vocabulary for each window.

Algorithm:
(1) Calculate the cost $G$ for each keyword $K$ over each candidate window.
(2) Sort keyword hypotheses according to $G$, removing any for which the cost $G$ is greater than a threshold $G_{max}$.
(3) Remove overlapping keyword hypotheses: make a pass through the sorted keyword hypotheses starting with the highest-ranked keyword, removing all hypotheses with time-overlap greater than overlap%.
(4) Return all keyword hypotheses with cost less than $G_{best} + G_{beam}$, where $G_{best}$ refers to the cost of the highest-ranked keyword and $G_{beam}$ is beam width.
Approach 3: Lattice search

- For spoken term detection.
- Vocabulary and corpus independent.
- Poorer results than the previous approaches due to the lack of data used to train it.
- Search in lattice tries to compensate the errors made in the phone and grapheme decoding.

```
<kw><kw>...<kw>
```

```
Final output: <kw><kw>...<kw>
```
Approach 3: Lattice search

- Exact match: Search in lattice tool from Brno University Technology (Szoke et al. 2005).
  Confidence score in the exact match:
  \[ CKW = La(KW) + L(KW) + Lb(KW) − L_{best} \]

1) \( La(KW) \) is the log likelihood of the best path from the beginning of the lattice to the node of the first phone (or grapheme) of KW
2) \( L(KW) \) is the log likelihood of keyword KW, computed as the sum of the acoustic log likelihood of its constituent phones (or graphemes) plus the total language model log likelihood for this sequence, weighted by the language model scale factor.
3) \( Lb(KW) \) is the log likelihood of the best path from the node of the last phone (or grapheme) of KW to the end of the lattice.
4) \( L_{best} \) is the likelihood of the 1-best path in the whole lattice

\( La(KW) \) and \( Lb(KW) \) are computed using a forward or backward Viterbi search (respectively).
### Spanish Albayzin database

<table>
<thead>
<tr>
<th></th>
<th>Phonetic corpus</th>
<th>Geographic corpus</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Train set</strong></td>
<td>Training the acoustic models and phoneme and grapheme bigram language models. 4800 phonetically balanced sentences by 164 speakers</td>
<td>Training the lexical access module in approach 1 and 2 and tuning various parameters.</td>
</tr>
<tr>
<td><strong>Test set</strong></td>
<td>Tuning the insertion penalty and LM scale of the phone and grapheme decoder in approach 2 and 3. 2000 sentences by 40 speakers.</td>
<td>Computing the final evaluation metrics.</td>
</tr>
</tbody>
</table>

80 keywords chosen based on their high frequency of occurrence and suitability as a search term for information retrieval in this domain.
Evaluation

- Figure-of-Merit (FOM): Defined by NIST as an upper-bound estimate on word spotting accuracy averaged over 1 to 10 false alarms per hour.
- Occurrence-weighted value (Occ): Defined by NIST for the STD Evaluation 2006. Computed by accumulating a value for each correct detection and subtracting a cost for false alarms as follows:

\[
OCC = \sum_{t \in \text{terms}} \left[ VN_{\text{correct}}(t) - CNfa(t) \right] \\
\sum_{t \in \text{terms}} VN_{\text{true}}(t)
\]
Evaluation

• Actual term-weighted value (ATWV): Defined by NIST for the STD Evaluation 2006. Computed by averaging a weighted sum of miss and false alarms probabilities over terms as follows:

$$ATWV = 1 - \frac{\sum_{t \in \text{terms}} [P_{miss}(t) + \partial P_{fa}(t)]}{\sum_{t \in \text{terms}} 1}$$

$$\partial = \frac{C}{V} (P_{prior}(t)^{-1} - 1)$$

$$P_{prior}(t) = 10^{-4} \quad \frac{C}{V} = 0.1$$
# Results

<table>
<thead>
<tr>
<th>FOM</th>
<th>Monophone</th>
<th>Monographeme</th>
<th>Triphone</th>
<th>Trigrapheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Best</td>
<td>65.92</td>
<td>66.71</td>
<td>68.04</td>
<td>71.34</td>
</tr>
<tr>
<td>Hybrid</td>
<td>78.34</td>
<td>77.31</td>
<td>80.61</td>
<td>79.09</td>
</tr>
<tr>
<td>Lattice</td>
<td>43.11</td>
<td>56.99</td>
<td>45.99</td>
<td>62.49</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Occ.</th>
<th>Monophone</th>
<th>Monographeme</th>
<th>Triphone</th>
<th>Trigrapheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Best</td>
<td>0.557</td>
<td>0.531</td>
<td>0.479</td>
<td>0.535</td>
</tr>
<tr>
<td>Hybrid</td>
<td>0.703</td>
<td>0.703</td>
<td>0.701</td>
<td>0.670</td>
</tr>
<tr>
<td>Lattice</td>
<td>0.404</td>
<td>0.517</td>
<td>0.390</td>
<td>0.569</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ATWV</th>
<th>Monophone</th>
<th>Monographeme</th>
<th>Triphone</th>
<th>Trigrapheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Best</td>
<td>0.235</td>
<td>0.185</td>
<td>0.231</td>
<td>0.176</td>
</tr>
<tr>
<td>Hybrid</td>
<td>0.498</td>
<td>0.453</td>
<td>0.478</td>
<td>0.374</td>
</tr>
<tr>
<td>Lattice</td>
<td>0.147</td>
<td>0.114</td>
<td>0.121</td>
<td>0.198</td>
</tr>
</tbody>
</table>
Conclusions and future work

• 1-Best approach: Rates are very similar for all the acoustic models. Advantage of the cost matrix in the lexical access module. NIST evaluation.

• Hybrid approach: Rates again very similar for all approaches due to the amount of information provided.

• Lattice approach: No additional information is provided and rates are very different for each acoustic model. NIST evaluation.

• Grapheme outperforms clearly phonemes in spoken term detection: Lattice approach.

• Sometimes context-independent outperforms context-dependent for Occ and ATWV. Systems were tuned to achieved the optimal performance for FOM.
Conclusions and future work

• The more vocabulary-dependant the approach is, the better results are achieved.

• As future work:
  – Tuning more parameters in all approaches. Not only for FOM optimization but also for Occ and ATWV one.
  – Applied the grapheme-based approach for meetings domain.

• Questions, comments…???
References