Focusing Language Models For Automatic Speech Recognition

Daniele Falavigna, Roberto Gretter
FBK, Italy

The work leading to these results has received funding from the European Union under grant agreement n° 287658
Outline

• Problem definition
• Auxiliary data selection
  • TFxIDF
  • Proposed method
  • Perplexity based method
• Computational issues
  • TFxIDF vs proposed method
• Experiments
• Discussion
Problem definition

• Given a general purpose text corpus and a given speech to transcribe
  • Build a LM which is focused on the particular (unknown) topic of the speech
  • No need for instantaneous, but should be quick

• Approach:
  • Perform a first ASR pass
  • Use recognition output to select text data “similar” to the context
  • Build a focused language model
  • Use the focused language model in the next ASR pass
Recognition setup

- **off-line**
  - text corpus
  - baseline LM

  \[\text{automatic selection}\]

  \[\text{auxiliary corpus}\]
  - auxiliary LM

- **speech**
  - ASR first + second step
  - word graph

  \[\text{ASR word graph rescoring}\]

  \[\text{1-best}\]
terminology

- **text corpus**
  - composed by $N$ rows ($N$ documents)
  - average length of a document: $L_c$

- **dictionary**
  - composed by $t_d$ terms, $1 \leq d \leq D$

- **auxiliary corpus**
  - composed by rows of the text corpus, size: $K$ words

- **speech to recognize**
  - TED talks, average length: $L_t$
Auxiliary data selection

• rationale:
  • score each row in the text corpus against ASR output
  • sort rows according to score
  • select the first rows → auxiliary corpus (having size K)

• 3 approaches implemented and compared:
  • TFxIDF
  • Proposed method
  • Perplexity based method

• domain specific data (TED LM)
Auxiliary data selection: TFxIDF

• for each talk \(i\) and for each word \(t_d\) compute:

\[
c^i_d[t_d] = (1 + \log(t^i_d)) \log\left(\frac{D}{df_d}\right) \quad 1 \leq d \leq D
\]

- \(t^i_d\) = frequency of term \(t_d\) inside talk
- \(df_d\) = # of documents in the corpus containing \(t_d\)

• compute the same for each row \(R^n\) in the corpus, \(1 \leq n \leq N\)

• estimate a similarity score:

\[
s(C^i, R^n) = \frac{C^i \cdot R^n}{|C^i| \cdot |R^n|}
\]
Auxiliary data selection: Proposed method

- sort words in dictionary according to frequency
- discard most frequent words ($< D_1 = 100$)
  - they don’t carry semantic information
- discard most rare words ($> D_2 = 200K$)
  - too rare to help, include typos
- replace words in corpus by their index in dictionary
- sort indices in each row to allow quick comparison
- estimate a similarity score:

$$s'(C'^i, R'^n) = \frac{\text{common (} C'^i, R'^n \text{)}}{\text{dim}(C'^i) + \text{dim}(R'^n)}$$
Auxiliary data selection: Proposed method

- example:
  - I would like your advice about rule one hundred forty three concerning inadmissibility
  - 47 54 108 264 2837 63 1019 6 12
    65 24 4890 166476
  - 108 264 2837 1019 4890 166476
    (like your advice rule concerning inadmissibility)
  - 108 264 1019 2837 4890 166476
Auxiliary data selection: Proposed method

- similarity score computation:
  - the lower index increment

score = $\frac{3}{12}$
Auxiliary data selection: Perplexity based method

• train a 3-gram LM using ASR output

• estimate perplexity for each row in the corpus

• use perplexity as a similarity score
Auxiliary data selection: Run time computational complexity

- corpus size: \( N \) (5.7M) rows, average row length \( L \) (272)
- dictionary size: \( D \) (1.6M) \( (D_2=200K) \)

<table>
<thead>
<tr>
<th></th>
<th>TFxIDF</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arithmetic operations</td>
<td>( O(2 \times N \times L) )</td>
<td>( O(N \times L / 2) )</td>
</tr>
<tr>
<td>Memory requirements</td>
<td>( O(D + N \times L) )</td>
<td>---</td>
</tr>
</tbody>
</table>
Training data

• text corpus
  • google news
  • 5.7 M documents, 1.6 G words
  • 272 words per document
• LM for rescoring:
  • 4-gram backoff LM, modified shift
  • 1.6M unigrams, 73M bigrams, 120M 3-grams and 195M 4-grams.
• FSN for first & second step:
  • 200K words, 37M bigrams, 34M 3-grams, 38M 4-grams.
Test data

- TED talks (test sets of IWSLT 2011)
- auxiliary corpus and auxiliary LM computed for each talk

<table>
<thead>
<tr>
<th></th>
<th>dev-set (19 talks)</th>
<th>test-set (8 talks)</th>
</tr>
</thead>
<tbody>
<tr>
<td>#words</td>
<td>44505</td>
<td>12431</td>
</tr>
<tr>
<td>(min, max, mean)</td>
<td>(591, 4509, 2342)</td>
<td>(484, 2855, 1553)</td>
</tr>
</tbody>
</table>

- performance are reported as a function of $K$, the number of words used to train the auxiliary LMs
Results

- Perplexity as a function of $K$
  - 0 means no interpolation

K is expressed in Kwords

- Perplexity interpolating the baseline LM with a domain specific LM (trained on ted2011 text, 2 Mwords):
  - dev set: **158**
  - test set: **142**
Results

• WER as a function of K
  • 0 means no interpolation

K is expressed in Kwords

• WER interpolating the baseline LM with a domain specific LM (trained on ted2011 text, 2 Mwords):
  
  dev set: 18.7
  test set: 18.4
Conclusion

• Method for focusing LMs without using in-domain data
• Comparison between the proposed method and TFxIDF
  • similar performance
  • less demanding computational requirements
• Comparable results if using in-domain data
  • in this setting…
• Future work:
  • how to add new words (to reduce OOV?)
  • instantaneous LM focusing
Thank you for the attention
LM interpolation

- LM probability associated to every arc of the word graph:

$$P[w|h] = \sum_{j=1}^{J} \lambda_j P_j[w|h]$$

- $J =$ number of LMs to combine
- $\lambda_j =$ weights estimated to minimize the overall perplexity on a development set

The interpolation weights, $i$ base and $i$ aux, associated to the two LMs (LMbase and Lmi aux) are estimated so as to minimize the overall LM perplexity on the 1-best output (the same used to build the $ith$ query document), of the second ASR decoding step.