A New Sentence Compression Dataset and Its Use in an Abstractive Generate-and-Rank Sentence Compressor

Dimitris Galanis and Ion Androutsopoulos

Natural Language Processing Group
Department of Informatics
Athens University of Economics and Business
Introduction

- **Sentence compression**: produce a shorter form of a sentence, which is grammatical and retains the most important information.

- **Most work until now has focused on Extractive Compression**
  - Only word deletions are permitted.

- **Abstractive Compression**:
  - Reorderings/substitutions and deletions are permitted
  - More compressed output can be produced

- **Example**:  
  - **Src**: For all that, observers are unanimous that Swapo will emerge with a clear majority.
  - **Gen**: Despite that, all observers believe that Swapo will win.

- **Examples of applications of sentence compression**:
  - text summarization
  - displaying texts on small screens
Introduction

- Lack of datasets to train and evaluate abstractive sentence compressors.
- Lack of reliable automatic evaluation measures for abstractive compression.
- We present a new dataset that contains candidate extractive and abstractive compressions of source sentences.
- The compressions are annotated with grammaticality and meaning preservation scores.
- The dataset was used to train and evaluate our generate-and-rank compressor.
Generating extractive candidates.
  ◦ We used our extractive compression system (NAACL 2010)

Generate abstractive candidates by applying paraphrase rules on extractives (Zhao 2009).

Keep the 10 candidates of all with the highest Language Model (LM) scores.

Produced candidates were split in two parts
  ◦ Training part: 188 source, 1695 candidates
  ◦ Testing part: 158 source sentences, 1377 candidates
Human Annotations

• 16 judges
• Judges were asked to rate each candidate given its source in terms of:
  ◦ grammaticality (G)
  ◦ meaning preservation (M)
• 1-5 scale (1.rubbish,...,5.perfect)
• GM score: sum of G and M
• Each pair was annotated by one judge
Inter-annotator agreement

- 22 source sentences.
- 161 source-candidate pairs.
- 3 judges.

<table>
<thead>
<tr>
<th></th>
<th>candidate compressions</th>
<th>average Pearson correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extractive</td>
<td>112</td>
<td>0.71</td>
</tr>
<tr>
<td>Abstractive</td>
<td>49</td>
<td>0.64</td>
</tr>
<tr>
<td>All</td>
<td>161</td>
<td>0.69</td>
</tr>
</tbody>
</table>

Inter-annotator agreement on GM scores.
## Resulting dataset

<table>
<thead>
<tr>
<th></th>
<th>Training part</th>
<th></th>
<th>Test part</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>extractive candidates</td>
<td>abstractive candidates</td>
<td>total candidates</td>
<td>extractive candidates</td>
<td>abstractive candidates</td>
</tr>
<tr>
<td>GM score</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>13 (1.3%)</td>
<td>10 (1.3%)</td>
<td>23 (1.3%)</td>
<td>19 (1.9%)</td>
<td>2 (0.4%)</td>
</tr>
<tr>
<td>3</td>
<td>26 (2.7%)</td>
<td>28 (3.6%)</td>
<td>54 (3.1%)</td>
<td>10 (1.0%)</td>
<td>0 (0%)</td>
</tr>
<tr>
<td>4</td>
<td>55 (5.8%)</td>
<td>29 (5.1%)</td>
<td>94 (5.5%)</td>
<td>51 (5.3%)</td>
<td>26 (6.2%)</td>
</tr>
<tr>
<td>5</td>
<td>52 (5.5%)</td>
<td>65 (8.5%)</td>
<td>117 (6.9%)</td>
<td>77 (8.0%)</td>
<td>42 (10.0%)</td>
</tr>
<tr>
<td>6</td>
<td>102 (10.9%)</td>
<td>74 (9.7%)</td>
<td>176 (10.3%)</td>
<td>125 (13.0%)</td>
<td>83 (19.8%)</td>
</tr>
<tr>
<td>7</td>
<td>129 (13.8%)</td>
<td>128 (16.8%)</td>
<td>257 (15.1%)</td>
<td>151 (15.7%)</td>
<td>53 (12.6%)</td>
</tr>
<tr>
<td>8</td>
<td>157 (16.8%)</td>
<td>175 (23.0%)</td>
<td>332 (19.5%)</td>
<td>138 (14.3%)</td>
<td>85 (20.3%)</td>
</tr>
<tr>
<td>9</td>
<td>177 (18.9%)</td>
<td>132 (17.3%)</td>
<td>309 (18.2%)</td>
<td>183 (19.0%)</td>
<td>84 (20.1%)</td>
</tr>
<tr>
<td>10</td>
<td>223 (23.8%)</td>
<td>110 (14.4%)</td>
<td>333 (19.6%)</td>
<td>205 (21.3%)</td>
<td>43 (10.2%)</td>
</tr>
<tr>
<td>total</td>
<td>934 (55.1%)</td>
<td>761 (44.9%)</td>
<td>1,695 (100%)</td>
<td>959 (69.6%)</td>
<td>418 (30.4%)</td>
</tr>
</tbody>
</table>

Distribution of GM scores (grammaticality plus meaning preservation) in our dataset.
Experiments

- Compression rate (in chars)

\[ CR(c_{ij}|s_i) = \frac{|c_{ij}|}{|s_i|} \]

- Ideal score for a candidate

\[ GMC_\gamma(c_{ij}|s_i) = GM(c_{ij}|s_i) - \gamma \cdot CR(c_{ij}|s_i) \]

- We used Support Vector Regression (SVR) to train our abstractive compression system.
- We evaluate using avg. GM score and avg. CR.
Experiments (on test set)

Upper bound: Choose the candidate with the highest ideal score for different γ values

SVR trained systems: choose the candidate using their predictions

Lower bound: assign scores randomly
Experiments

- SVR models are trained using \( l \) training vectors and learn a function \( f : \mathbb{R}^n \rightarrow \mathbb{R} \).
- We use \( G_m c_{ij} s_i \) as \( y_i \).
- 3 SVR models were built.
- SVR-BASE
  - LM score of source
  - LM score of candidate
  - Extractive system’s score
  - CR of extractive
  - #Zhao’s rules applied
Experiments

- **SVR-BASE+PMI**
  - 3 more features
  - Avg. **Pointwise Mutual Information** for word pairs of source, extractive and abstractive.
  - PMI is used to assess if the words of a sentence have a high probability of co-occurring.
  - Use only content words (nouns, verbs, adjectives and adverbs).
  - Consider only word pairs at maximum distance of 10.
  - Distributions were learnt from a large corpus (AQUAINT).

\[
\text{PMI}(\sigma) = \frac{1}{N} \cdot \sum_{i,j} \text{PMI}(w_i, w_j)
\]

\[
\text{PMI}(w_1, w_2) = \log \frac{P(w_1, w_2)}{P(w_1) \cdot P(w_2)}
\]
**LDA**

- Latent Dirichlet Allocation (LDA) is a model of text generation.
- It assumes that a text is generated for a mixture of topics.
- An LDA model is trained on a large number of texts given a (predefined) number of topics.
- Once the LDA model is trained, given an unseen text \( d \) we can predict \( P(w|d) \).

\[
P(w|d) = \sum_t P(w|t) \cdot P(t|d)
\]

- \( P(w|t) \) is learnt during training.
- \( P(t|d) \) is inferred using various methods (Gibbs sampling).

\( t \): topic
Experiments

- **SVR-BASE+PMI+LDA**
  - LDA is used to assess if the words of the candidate sentences “discuss” the same topics as the source.
  - 3 more features: **LDA-based score** for source, extractive and abstractive
    - $d$ in our case corresponds to a sentence $s$ and not a document.
    - LDA score should be high for good compressions.

- We use the $P(t|s)$ of the source sentence also for candidates.
  - $P(w|s)$ should be high for words of good compressions
    - High $P(w|t)$
    - High $P(t|s)$

\[
LDA(\sigma|s_i) = \frac{1}{|\sigma|} \cdot \sum_{r=1}^{\sigma} \log P(w_r|s_i)
\]
Experiments (on test set)

Upper bound: Choose the candidate with the highest ideal score for different $\gamma$ values

SVR trained systems: choose the candidate using their predictions

Lower bound: assign scores randomly
Experiments with different train sizes

- The size of the training set positively affects the system’s performance.
Conclusions & Future work

- A new dataset was constructed for abstractive compression.
  - It can be used to train and evaluate compression systems.
- A new method for abstractive compression was presented.
- We plan to enrich our dataset with more abstractive candidates.
  - Use multiple translation engines (Zhao et. al. 2010)
- The dataset will soon be publicly available.
- Questions?