Task

Given a small number of keywords generate a sentence

Approach

Frame it as a cooperative communication game and develop an unsupervised learning approach

Two competing goals

\[
\text{cost}(x, \alpha) = \mathbb{E}_{q_a(z|x)}[\# \text{tokens}(z)] \quad \text{Number of keywords}
\]

\[
\text{loss}(x, \alpha, \beta) = \mathbb{E}_{q_a(z|x)}[- \log p_{\beta}(x|z)] 
\quad \text{Reconstruction error}
\]

Modeling with autoencoders

Encoder (User) \quad Encode x into z by keeping a subset of tokens

Decoder (System) \quad Predict distribution over x conditioning on z

Multi-objective optimization

Linear \quad \min_{\alpha, \beta} \mathbb{E}[\text{cost}(x, \alpha)] + \mathbb{E}[\text{loss}(x, \alpha, \beta)] \quad \times

Constrained \quad \min_{\alpha, \beta} \mathbb{E}[\text{cost}(x, \alpha)] \quad \text{subject to } \mathbb{E}[\text{loss}(x, \alpha, \beta)] \leq \epsilon \quad \checkmark

Optimization using policy gradient

Loss function \quad \text{Lagrangian of constrained objective}

\[
J(\alpha, \beta, \lambda) = \mathbb{E}[\text{cost}(x, \alpha)] + \lambda \mathbb{E}[\text{loss}(x, \alpha, \beta)] - \epsilon
\]

Policy gradient \quad \text{REINFORCE with baseline using single sample}

\[
\nabla_{\alpha} J(\alpha, \beta, \lambda) = \mathbb{E}[q_{a}(z|x)] \nabla_{\alpha} \log q_a(z \mid x) \cdot G(x, z)]
\]

\[
G(x, z) = \frac{\# \text{tokens}(z)}{\text{per-example cost}} + \frac{\lambda(- \log p_{\beta}(x|z))}{\text{per-example loss}}
\]

Why constrained objective?


Why joint training of encoder and decoder?

Better tradeoff between cost and loss

Unstable training

Suboptimal tradeoff

Arbitrary value for lambda

Selecting random keywords results in bad decoder

Jointly trained encoder drops more than just stopwords

Correlation with part-of-speech

Higher kept rates for content words

User study

This keyword-based autocomplete can save time by nearly 50% compared to fully typing sentences

<table>
<thead>
<tr>
<th>Part-of-Speech</th>
<th>Examples</th>
<th>Kept (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determiner</td>
<td>the, a, this</td>
<td>4</td>
</tr>
<tr>
<td>Conjunction</td>
<td>and, but</td>
<td>5</td>
</tr>
<tr>
<td>Pronoun</td>
<td>it, you, we</td>
<td>10</td>
</tr>
<tr>
<td>Verb</td>
<td>love, recommend</td>
<td>28</td>
</tr>
<tr>
<td>Adverb</td>
<td>very, pretty</td>
<td>35</td>
</tr>
<tr>
<td>Adjective</td>
<td>delicious</td>
<td>36</td>
</tr>
<tr>
<td>Noun</td>
<td>service, food</td>
<td>40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time for typing</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keywords: 3.85 seconds</td>
<td>Paraphrased: 81%</td>
</tr>
<tr>
<td>Sentences: 5.76 seconds</td>
<td>Exact match: 18%</td>
</tr>
</tbody>
</table>

MaskGAN: Better Text Generation via Filling in the _____ (Fedus et al., 2018)
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Devlin et al., 2019)
BottleSum: Unsupervised and Self-supervised Sentence Summarization using the Information Bottleneck Principle (West et al., 2019)