Our goal is to inject the fine-grained topical preference of users on the chosen topics continuation conditioned on his presidency … Obama writes a new book” Input Prompt: “Barack Obama writes a new book”

Output Continuation 2: on an spirituality and the role of religion in society … What does this continuation sound? 


We really want to read the mind of LM and control it.

I hope the book would be about elections.

Transformer-based Language Models

Output Continuation 1: that shares his perspective on his presidency … What about this? Next What about him? Next

Introduction

Our Model

Model Architecture (Testing)

• Option Generator
  • Predict the topic embedding in a GloVe space
  • Use 3 words closest to each topic embedding to visualize the topic

• Conditional Text Generator
  • Convert the selected GloVe into the size of the hidden sampling
  • Using top-k sampling

Model Training

• Training without any label data or predefined topics

• Option Generator
  • By pulling the topics and the future words closer
  • Topic -> cluster center
  • By pushing the topics and randomly selected words away

• Conditional Text Generator
  • Use the GloVe of future words as the condition

Our Framework

1. User: Specify a prompt
2. LM: suggest topics
   • Topics are prompt-dependent and fine-grained
3. User: Choose topics
4. LM: generate the continuation conditioned on the chosen topics

Repeat -> plot graph

Our Method

Experiments

Qualitative Comparison

Input Prompt: Barack Obama writes a new book”

Output Continuation 1: that shares his perspective on his presidency … Obama writes a new book”

Output Continuation 2: on a spirituality and the role of religion in society …


I hope the book would be about elections.

I really want to read the mind of LM and control it.

Model Architecture (Testing)

• Option Generator
  • Predict the topic embedding in a GloVe space
  • Use 3 words closest to each topic embedding to visualize the topic

• Conditional Text Generator
  • Convert the selected GloVe into the size of the hidden sampling
  • Using top-k sampling

Model Training

• Training without any label data or predefined topics

• Option Generator
  • By pulling the topics and the future words closer
  • Topic -> cluster center
  • By pushing the topics and randomly selected words away

• Conditional Text Generator
  • Use the GloVe of future words as the condition

↓ Training

↑ Testing

Conditional Text Generator Evaluation

Test Generator Method Token Word Type Fluencty Diversity Novelty Overall

GPT2 1.37 1.48 1.37 1.48 1.37 1.48 1.37 1.48

GPT2 1.27 0.84 0.64 1.48 39.80 80.22 1.00 24.99 0.77 48.69 6.15 0.61 0.11

Conclusion

• Decompose a novel framework into two novel components
  • Option Generator -> topics are relevant but novel
  • Conditional Text Generator -> Text is fluent and relevant

• Codes are available at https://github.com/iesl/interactive_LM

References

• Haw-Shiuwan Chang, Arbel Araigal, and Andrew McCallum. 2021 Extending Multi-Sense Word Embedding to Phrases and Sentences for Unsupervised Semantic Applications. In AAAI 2021