

Generating Slogans with Linguistic Features using Sequence-to-Sequence Transformer

Yeoun Yi

Seoul National University
yeounyi@gmail.com

Hyopil Shin

Seoul National University
hpshin@snu.ac.kr

Abstract

Previous work generating slogans depended on templates or summaries of company descriptions, making it difficult to generate slogans with linguistic features. We present LexPOS, a sequence-to-sequence transformer model that generates slogans given phonetic and structural information. Our model searches for phonetically similar words given user keywords. Both the sound-alike words and user keywords become lexical constraints for generation. For structural repetition, we use POS constraints. Users can specify any repeated phrase structure by POS tags. Our model-generated slogans are more relevant to the original slogans than those of baseline models. They also show phonetic and structural repetition during inference, representative features of memorable slogans.

1 Introduction

Advertising slogans share many linguistic features, such as phonetic or structural repetition (Musté et al. (2015)). These factors make slogans more memorable (Reece et al. (1994)). However, most previous works on slogan generation depended on templates or summaries of company descriptions, making it difficult to generate slogans with linguistic features.

We present LexPOS, a sequence-to-sequence (seq2seq) transformer model with an additional POS encoder. It models the phonetic and structural repetition in slogans, using lexical and POS constraints. When given keywords and POS tags of

the desired output structure as input, the model finds words that sound and mean similar to the user keywords. The model-generated slogans include both the user keywords and one sound-alike word. They also reflect the POS constraints. For instance, if a user inputs the word ‘cake’ and [‘VERB’, ‘DET’, ‘NOUN’, ‘PUNCT’, ‘VERB’, ‘DET’, ‘NOUN’, ‘PUNCT’], the output could be ‘Bake a cake, bake a smile’. It includes the word ‘cake’ and its sound-alike word ‘bake’ and has repeated verb phrases. The source code, pretrained weights, and data are available online¹.

This paper primarily makes the following contributions:

- Generating slogans taking linguistic features into account.
- Utilizing a pretraining method of BART and T5 to model lexical constraints.
- Proposing a novel approach to model structural constraints by adding a POS encoder.

2 Previous Work

Most of the previous work in slogan generation focused on modifying templates. BRAINSUP, proposed by Özbal et al. (2014), is the first study to generate customized slogans with lexical, emotional, and domain constraints. BRAINSUP utilizes morpho-syntactic patterns mined from corpus as templates. It first selects the most compatible template and fills the empty slots in the template according to user specifications. Before returning the results, it evaluates the candidate slogans with various metrics, including phonetic repetition. However, it can only determine whether the same phonetic features were used or not.

¹ <https://github.com/yeounyi/LexPOS>

Munigala et al. (2018) presented a model to generate persuasive sentences from fashion product descriptions. It expands fashion-related keywords from inputs and generates sentences using a domain-specific neural language model (LM). Keywords from inputs, expanded keywords, and common functional words are the only candidates at each time step of the LM. The overall perplexity is minimized with beam search. The limitation is that only imperatives can be generated, as the sentences always begin with a verb.

Jin et al. (2021) introduced a sequence-to-sequence transformer model to generate diverse slogans from company descriptions. They considered slogan generation as abstractive summarization of company descriptions and chose the BART-style encoder-decoder model (Lewis et al. (2020)) with a bidirectional encoder and an autoregressive decoder. To prevent unrelated company names from appearing in slogans, they delexicalized all the company names. In addition, they trained a model conditioned on the first words' POS tag, generating syntactically diverse slogans.

Unlike previous works, we do focus on linguistic features and not depend on templates at the same time. We take phonetic and structural repetition into account, factors that make slogans memorable and unique.

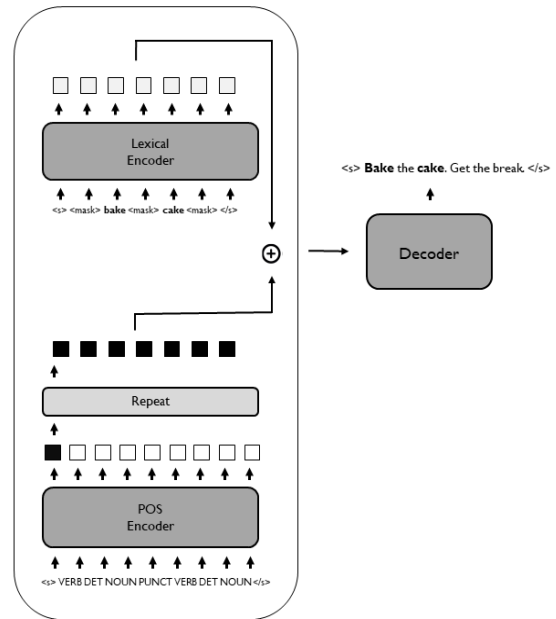
3 Model

Our model first forms the lexical constraints. During training, it uses the given lexical constraints as it is. During inference, it searches for sound-alike words of user keywords. We use the phonetic vector representation proposed by Parrish (2017). The phonetic vector uses interleaved phonetic feature bigrams extracted from phonetic transcriptions and it covers all the words in CMU Pronouncing Dictionary². The model also considers the semantic similarity of sound-alike words, to improve the naturalness of the outputs. We use pretrained Glove embeddings (Pennington et al. 2014) for semantic similarity. After we compute cosine similarity to select the top 100 phonetically similar words, we sort them in semantic similarity. We exclude words that are not present in Glove embeddings (Pennington et al. 2014) or BART tokenizer vocabulary, not to use

unfamiliar words. We select the first three words to each form the lexical constraints, together with user keywords. Unlike lexical constraints, POS constraints don't need further processing during training and inference. POS constraints during inference can be manually specified or popular POS structures from data would be recommended.

After processing the lexical constraints, the Transformer architecture (Vaswani et al., 2017) comes in. The Transformer architecture has achieved state-of-the-art results on various natural language processing tasks. We apply a Transformer-based sequence-to-sequence model because we need encoders for constraints and decoders for generation. To leverage the power of pretrained transformers, we utilized the pretrained weights of BART and T5 (Raffel et al., 2020) released by HuggingFace³. We choose BART and T5 because both models were pretrained by denoising consecutive spans of corrupted tokens, meaning they can generate natural sentences using lexical constraints.

The only architectural difference is that our model has an additional encoder. One encoder encodes the lexical constraints, and the other encodes the POS constraints. The weights of the POS encoder are randomly initialized, and the vocabulary size of the POS encoder is limited to 20. The vocabulary includes the spaCy⁴ POS tags and <s>, </s>, <pad> tokens.



² <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>

³ <https://huggingface.co/transformers>

⁴ <https://spacy.io/>

Slogan	Lexical Input	POS Input
Breakfast of Champions.	<mask> breakfast <mask> champions <mask>	['NOUN', 'ADP', 'NOUN', 'PUNCT']
The Best a Man Can Get.	<mask> best <mask> man <mask>	['DET', 'ADJ', 'DET', 'NOUN', 'AUX', 'VERB', 'PUNCT']
Think Different.	<mask> different <mask> think <mask>	['VERB', 'ADJ', 'PUNCT']
America Runs on <name>.	<mask> <name> <mask> america <mask> runs <mask>	['PROPN', 'VERB', 'ADP', 'PROPN', 'PUNCT']

Table 1: Example of data. Lexical constraints are bolded in lexical inputs. Special tokens are omitted.

Figure 1: Architecture of LexPOS model.

To incorporate the POS constraints into the rest of the model, the last hidden states of <s> token in the POS encoder are repeated with the length of the last hidden states in the lexical encoder. We choose the last hidden state of <s> token because it is widely assumed to include representative information of all tokens. These two hidden states are summed and given to the decoder. Then, the decoder generates slogans with the given lexical and POS constraints. Figure 1 presents the architecture of our proposed model.

4 Data

Our training objective is to implement lexical and POS constraints. The desired model-generated slogans should follow the lexical constraints and the POS structural constraints.

We crawl 30,759 unique slogans from online slogan databases such as Textart.ru⁵, Slogans Hub⁶, Slogan List⁷, Think Slogans⁸, and Slogans Point⁹. Unlike previous works focusing on commercial slogans, our dataset covers both commercial and public slogans. Public slogans include slogans for health, women’s rights, the environment, and more. 45.43% of our slogans are commercial, 54.56% are public. Company names in commercial slogans are delexicalized using a custom special token <name>, following Jin et al. (2021). We reserve 20% of the data for validation.

The lexical inputs are lexical constraints surrounded by <mask> tokens. Just like the pretraining method of BART and T5, our model predicts consecutive spans of <mask> tokens.

The lexical constraints are limited to verbs, nouns, proper nouns, and adjectives. We extract them from original slogans using spaCy. Then, we randomly delete lexical constraints, when there are 5 or more of them to keep the average ratio of the number of lexical constraints to the total number of words in the original slogan below 50%. The average ratio is 41.90%.

Unlike the pretraining method of BART and T5, we shuffle the lexical constraints to make the model predict natural ordering. If we don’t shuffle them, we need to permute lexical constraints during inference. For instance, if the user keyword is ‘cake’ and its sound-alike word is ‘bake’, we need both ‘<mask> cake <mask> bake <mask>’ and ‘<mask> bake <mask> cake <mask>’ as lexical inputs. The number of permutations would increase dramatically as the number of user keywords increase. To address this issue, we shuffle the lexical constraints.

The POS inputs are POS constraints themselves. We use spaCy POS tagging results of the original slogan as POS inputs. Table 1 shows the example of data.

5 Experiments

Following previous work, we conduct a quantitative evaluation using ROUGE (Lin (2004)) F1 scores and compare our model with the original sequence-to-sequence model baselines. We also compute the included lexical constraints rates and POS F1 scores to check how well the given constraints are applied. The included lexical constraints rates are the rates of the lexical constraints included in model-generated slogans.

⁵<http://www.textart.ru/database/slogan/map.html>

⁶<https://sloganshub.org/>

⁷<https://www.sloganlist.com/>

⁸<https://www.thinkslogans.com/>

⁹<http://www.sloganspoint.com/>

	Rouge 1	Rouge 2	Rouge L	Included Lexical Constraints Rates	POS F1
baseline BART	0.4894	0.2282	0.4577	0.9515	0.7992
baseline T5	0.4461	0.1805	0.4150	0.9406	0.7737
LexPOS BART	0.6204	0.3703	0.5921	0.9469	0.9176
LexPOS T5	0.5700	0.3039	0.5339	0.9399	0.8809

Table 2: The quantitative evaluation of various models. Best scores are bolded.

POS F1 scores are computed by comparing the POS inputs and POS tagging results of model-generated slogans.

Table 2 presents the quantitative evaluation result. Our best model achieved a ROUGE-1/-2/-L F1 score of 62.04/37.03/59.21, 94.69 for the included lexical constraints rates, and 91.76 for POS F1 scores. The performance discrepancy between BART and T5 could be explained by their pretraining methods. BART was pretrained using Sentence Permutation, which restores the original order of shuffled sentences, while T5 was not.

Table 3 shows the sample of generated slogans from validation data. The results of the LexPOS model are more relevant to the original slogans.

Gold: The Power of being Global. BART: Global Power. LexPOS BART: The power of global.
Gold: <name>. Keep Walking. BART: I’m walking <name>. LexPOS BART: <name>. Walking on.
Gold: How about a nice <name>? BART: Be nice to <name>. LexPOS BART: Always be a nice <name>.
Gold: All things are difficult before they are easy. BART: Difficult things are never easy. LexPOS BART: The difficult things are made easy by you.
Gold: Some bruises are on the inside. Stop bullying. BART: Stop bullying on the inside and stop bruises on the outside. LexPOS BART: The bruises are on the inside, stop bullying.

Table 3: Sample generated slogans from validation data. “Gold” is the original slogan.

Table 4 shows the inference results. We use beam search and adjust the temperature to generate natural slogans.

Keywords: bakery, sandwich POS: [VERB, DET, NOUN, PUNCT, VERB, DET, NOUN, PUNCT] Output: Switch the bakery, the sandwich switches .
Keywords: airline, cheap POS: [ADJ, NOUN, PUNCT, ADJ, NOUN, PUNCT] Output: Fast Airline. Keep Cheap .
Keywords: save, energy POS: [VERB, DET, NOUN, PUNCT, VERB, DET, NOUN, PUNCT] Output: Save energy, save faces .
Keywords: brunch, cafe POS: [NOUN, ADP, NOUN, PUNCT, NOUN, ADP, NOUN, PUNCT] Output: Brunch at Brightness , Crunch at Cafe.
Keywords: unique, fashion, brand POS: [NOUN, ADP, NOUN, PUNCT, NOUN, ADP, NOUN, PUNCT] Output: Brand of unique fashion . Passion for fashion .

Table 4: Sample generated slogans from user keywords and POS constraints. One of the user keywords and its sound-alike word are bolded.

The model-generated slogans include both the user keywords and one selected sound-alike word, fully reflecting users’ intentions. The results also show phonetic and structural repetition, representative features of memorable slogans.

6 Conclusion

In this work, we generate slogans with phonetic and structural repetition using LexPOS model, a transformer-based sequence-to-sequence model with an additional POS encoder. It generates slogans using sound-alike words given user keywords. The model-generated slogans also follow structural constraints thanks to the POS encoder. To our knowledge, it is the first model to generate slogans without templates, taking linguistic features into account. Future work

should implement other linguistic features shown in slogans.

References

- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 6000-6010.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, M. Matena, Yanqi Zhou, W. Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1-140:67.
- Jin, Y., Bhatia, A., Wanvarie, D., and Le, P. T. V. 2021. Toward Improving Coherence and Diversity of Slogan Generation. *arXiv*.
- Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871-7880.
- Lin, Chin-Yew. 2004. ROUGE: A Package for Automatic Evaluation of summaries. *Proceedings of the ACL Workshop: Text Summarization Braches Out 2004*. 10.
- Munigala, V., Mishra, A., Tamilselvam, S. G., Khare, S., Dasgupta, R., and Sankaran, A. 2018. PersuAIDE! An Adaptive Persuasive Text Generation System for Fashion Domain. In *Companion Proceedings of the Web Conference 2018*, pp. 335-342, Lyon, France.
- Musté, P., Stuart, K., & Botella, A. 2015. Linguistic Choice in a Corpus of Brand Slogans: Repetition or Variation. *Procedia - Social and Behavioral Sciences*, 198, 350-358.
- Özbal, G., D., and Strapparava, C. 2013. BRAINSUP: Brainstorming Support for Creative Sentence Generation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1446-1455, Sofia, Bulgaria.
- Parrish, A. 2017. Poetic Sound Similarity Vectors Using Phonetic Features. In *The AIIDE-17 Workshop on Experimental AI in Games*, pp. 99-106.
- Pennington, Jeffrey & Socher, Richard & Manning, Christopher. 2014. Glove: Global Vectors for Word Representation. *EMNLP*. 14. 1532-1543. 10.3115/v1/D14-1162.
- Reece, B.B., B.G. Vanden Bergh, and H. Li. 1994. What makes a slogan memorable and who remembers it? *Journal of Current Issues and Research in Advertising* 16, no. 2: 41-57.