Natural Language Engineering (2023), **29**, pp. 254–286
doi:10.1017/S1351324921000474

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

Towards improving coherence and diversity of slogan generation

Yiping Jin[1], Akshay Bhatia[2], Dittaya Wanvarie[1,*] and Phu T. V. Le[2]

[1]Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Bangkok, Thailand 10300 and [2]Knorex, 140 Robinson Road, #14-16 Crown @ Robinson, Singapore 068907
*Corresponding author. E-mail: Dittaya.W@chula.ac.th

(Received 11 February 2021; revised 13 December 2021; accepted 20 December 2021; first published online 4 February 2022)

Abstract

Previous work in slogan generation focused on utilising slogan skeletons mined from existing slogans. While some generated slogans can be catchy, they are often not coherent with the company's focus or style across their marketing communications because the skeletons are mined from other companies' slogans. We propose a sequence-to-sequence (seq2seq) Transformer model to generate slogans from a brief company description. A naïve seq2seq model fine-tuned for slogan generation is prone to introducing false information. We use company name delexicalisation and entity masking to alleviate this problem and improve the generated slogans' quality and truthfulness. Furthermore, we apply conditional training based on the first words' part-of-speech tag to generate syntactically diverse slogans. Our best model achieved a ROUGE-1/-2/-L F_1 score of 35.58/18.47/33.32. Besides, automatic and human evaluations indicate that our method generates significantly more factual, diverse and catchy slogans than strong long short-term memory and Transformer seq2seq baselines.

Keywords: Natural language generation; Sequence-to-sequence model; Slogan generation

1. Introduction

Advertisements are created based on the market opportunities and product functions (White 1972). Their purpose is to attract viewers' attention and encourage them to perform the desired action, such as going to the store or clicking the online ad. Slogans[a] are a key component in advertisements. Early studies in the fields of psychology and marketing revealed that successful slogans are **concise** (Lucas 1934) and **creative** (White 1972). Puns, metaphors, rhymes and proverbs are among the popular rhetorical devices employed in advertising headlines (Mieder and Mieder 1977; Phillips and McQuarrie 2009). However, as White (1972) noted, the creative process in advertising is 'within strict parameters', that is, the slogan must not diverge too much from the product/service it is advertising in its pursuit of creativity.

Another essential factor to consider is ads fatigue (Abrams and Vee 2007). An ad's effectiveness decreases over time after users see it repeatedly. It motivates advertisers to deliver highly personalised and contextualised ads (Vempati *et al*. 2020). While advertisers can easily provide a dozen alternative images and use different ad layouts to create new ads dynamically (Bruce *et al*. 2017), the ad headlines usually need to be manually composed. Figure 1 shows sample ads composed by professional ad creative designers, each having a different image and ad headline.

[a]We use 'slogan' and 'ad headline' interchangeably. A *slogan* is defined by its property as 'a short and memorable phrase used in advertising'. An *ad headline* is defined literarily by its function.

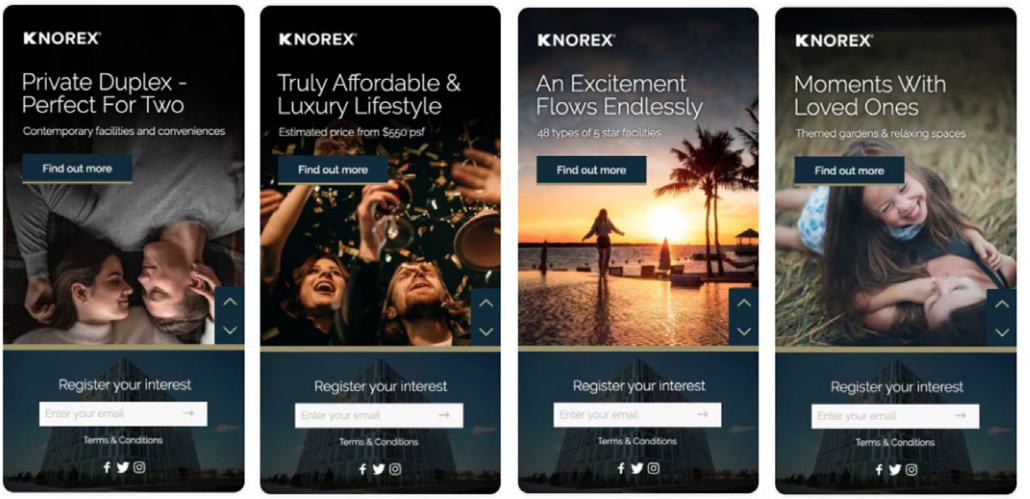

Figure 1. Sample ads for the same advertiser in the hospitality industry. The centring text with the largest font corresponds to the ad headline (slogan).

Previous work in automatic slogan generation focused almost exclusively on modifying existing slogans by replacing part of the slogan with new keywords or phrases (Özbal *et al*. 2013; Tomašic *et al*. 2014; Gatti *et al*. 2015; Alnajjar and Toivonen 2021). This approach ensures that the generated slogans are well formed and attractive by relying on skeletons extracted from existing slogans. For example, the skeleton 'The **NN** of Vintage' expresses that the advertised product is elegant. It can instantiate novel slogans like 'The Phone of Vintage' or 'The Car of Vintage'. However, the skeleton is selected based on the number of available slots during inference time and does not guarantee that it is coherent with the company or product. In this particular example, while some people appreciate vintage cars, the phrase 'The Phone of Vintage' might have a negative connotation because it suggests the phone is *outdated*. Such subtlety cannot be captured in skeletons represented either as part-of-speech (POS) tag sequences or syntactic parses.

In this work, we focus on improving **coherence** and **diversity** of a slogan generation system. We define coherence in two dimensions. First, the generated slogans should be consistent with the advertisers' online communication style and content. Therefore, albeit being catchy, a pun is likely not an appropriate slogan for a personal injury law firm. To this end, we propose a sequence-to-sequence (seq2seq) Transformer model to generate slogans from a brief company description instead of relying on random slogan skeletons.

The second aspect of coherence is that the generated slogans should not contain untruthful information, such as mistaking the company's name or location. Therefore, we delexicalise the company name and mask entities in the input sequence to prevent the model from introducing unsupported information.

Generating diverse slogans is crucial to avoid ads fatigue and enable personalisation. We observe that the majority of the slogans in our dataset are plain noun phrases that are not very catchy. It motivates us to explicitly control the syntactic structure through conditional training, which improves both diversity and catchiness of the slogans.

We validate the effectiveness of the proposed method with both quantitative and qualitative evaluation. Our best model achieved a ROUGE-1/-2/-L F_1 score of 35.58/18.47/33.32. Besides, comprehensive evaluations also revealed that our proposed method generates more truthful, diverse and catchy slogans than various baselines. The main contributions of this work are as follows:

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

- Applying a Transformer-based encoder–decoder model to generate slogans from a short company description.
- Proposing simple and effective approaches to improve the slogan's *truthfulness*, focusing on reducing entity mention hallucination.
- Proposing a novel technique to improve the slogan's syntactic *diversity* through conditional training.
- Providing a benchmark dataset and a competitive baseline for future work to compare with.

We structure this paper as follows. We review related work on slogan generation, seq2seq models and aspects in generation in Section 2. In Section 3, we present the slogan dataset we constructed and conduct an in-depth data analysis. We describe our baseline model in Section 4, followed by our proposed methods to improve truthfulness and diversity in Section 5 and Section 6. We report the empirical evaluations in Section 7. Section 8 presents ethical considerations and Section 9 concludes the paper and points directions for future work.

2. Related work

We review the literature in four related fields: (1) slogan generation, (2) sequence-to-sequence models, (3) truthfulness and (4) diversity in language generation.

2.1 Slogan generation

A slogan is a catchy, memorable and concise message used in advertising. Traditionally, slogans are composed by human copywriters, and it requires in-depth domain knowledge and creativity. Previous work in automatic slogan generation mostly focused on manipulating existing slogans by injecting novel keywords or concepts while maintaining certain linguistic qualities.

Özbal *et al.* (2013) proposed BRAINSUP, the first framework for creative sentence generation that allows users to force certain words to be present in the final sentence and to specify various emotion, domain, or linguistic properties. BRAINSUP generates novel sentences based on morphosyntactic patterns automatically mined from a corpus of dependency-parsed sentences. The patterns serve as general skeletons of well-formed sentences. Each pattern contains several empty slots to be filled in. During generation, the algorithm first searches for the most frequent syntactic patterns compatible with the user's specification. It then fills in the slots using beam search and a scoring function that evaluates how well the user's specification is satisfied in each candidate utterance.

Tomašic *et al.* (2014) utilised similar slogan skeletons as BRAINSUP capturing the POS tag and dependency type of each slot. Instead of letting the user specify the final slogan's properties explicitly, their algorithm takes a textual description of a company or a product as the input and parses for keywords and main entities automatically. They also replaced beam search with genetic algorithm to ensure good coverage of the search space. The initial population is generated from random skeletons. Each generation is evaluated using a list of 10 heuristic-based scoring functions before producing a new generation using crossovers and mutations. Specifically, the mutation is performed by replacing a random word with another random word having the same POS tag. Crossover chooses a random pair of words in two slogans and switches them. For example, input: ['Just *do* it', '*Drink* more milk'] ⇒ ['Just *drink* it', '*Do* more milk'].

Gatti *et al.* (2015) proposed an approach to modify well-known expressions by injecting a novel concept from evolving news. They first extract the most salient keywords from the news and expand the keywords using WordNet and Freebase. When blending a keyword into well-known expressions, they check the word2vec embedding (Mikolov *et al.* 2013) similarity between each keyword and the phrase it shall replace to avoid generating nonsense output. Gatti *et al.* (2015)

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

also used dependency statistics similar to BRAINSUP to impose lexical and syntactic constraints. The final output is ranked by the mean rank of semantic similarity and dependency scores, thus balancing the relatedness and grammaticality. In subsequent work, Gatti *et al*. (2017) applied a similar approach to modify song lyrics with characterising words taken from daily news.

Iwama and Kano (2018) presented a Japanese slogan generator using a slogan database, case frames and word vectors. The system achieved an impressive result in an ad slogan competition for human copywriters and was employed by one of the world's largest advertising agencies. Unfortunately, their approach involves manually selecting the best slogans from 10 times larger samples and they did not provide any detailed description of their approach.

Recently, Alnajjar and Toivonen (2021) proposed a slogan generation system based on generating nominal metaphors. The input to the system is a target concept (e.g., car), and an adjective describing the target concept (e.g., elegant). The system generates slogans involving a metaphor such as 'The Car Of Stage', suggesting that the car is as elegant as a stage performance. Their system extracts slogan skeletons from existing slogans. Given a target concept T and a property P, the system identifies candidate metaphorical vehicles[b] v. For each skeleton s and the $\langle T, v \rangle$ pair, the system searches for potential slots that can be filled. After identifying plausible slots, the system synthesises candidate slogans optimised using genetic algorithms similar to Tomašic *et al*. (2014).

Munigala *et al*. (2018) is one of the pioneer works to use a language model (LM) to generate slogans instead of relying on slogan skeletons. Their system first identifies fashion-related keywords from the product specifications and expands them to creative phrases. They then synthesise persuasive descriptions from the keywords and phrases using a large domain-specific neural LM. Instead of letting the LM generate free-form text, the candidates at each time step are limited to extracted keywords, expanded in-domain noun phrases and verb phrases as well as common functional words. The LM minimises the overall perplexity with beam search. The generated sentence always begins with a *verb* to form an imperative and persuasive sentence. Munigala *et al*. (2018) demonstrated that their system produced better output than an end-to-end long short-term memory (LSTM) encoder–decoder model. However, the encoder–decoder was trained on a much smaller parallel corpus of title text-style tip pairs compared to the corpus they used to train the LM.

Misawa *et al*. (2020) applied a Gated Recurrent Unit (GRU) (Cho *et al*. 2014) encoder–decoder model to generate slogans from a discription of a target item. They argued that good slogans should not be generic but distinctive towards the target item. To enhance distinctiveness, they used a reconstruction loss (Niu *et al*. 2019) by reconstructing the corresponding description from a slogan. They also employed a copying mechanism (See *et al*. 2017) to handle out-of-vocabulary words occurring in the input sequence. Their proposed model achieved the best ROUGE-L score of 19.38[c], outperforming various neural encoder–decoder baselines.

Similarly, Hughes *et al*. (2019) applied encoder–decoder with copying mechanism (See *et al*. 2017) to generate search ad text from the landing page title and body text. They applied reinforcement learning (RL) to directly optimise for the click-through rate. Mishra *et al*. (2020) also employed the same encoder–decoder model of See *et al*. (2017) to ad text generation. However, their task is to rewrite a text with a low click-through rate to a text with a higher click-through rate (e.g., adding phrases like 'limited time offer' or 'brand new').

Concurrent to our work, Kanungo *et al*. (2021) applied RL to a Transformer (Vaswani *et al*. 2017) using the ROUGE-L score as the reward. Their model generates ad headlines from *multiple* product titles in the same ad campaign. The generated headlines also need to generalise to multiple products instead of being specific to a single product. Their proposed method outperformed various LSTM and Transformer baselines based on overlap metrics and quality audits. Unfortunately,

[b] A metaphor has two parts: the tenor (target concept) and the vehicle. The vehicle is the object whose attributes are borrowed.

[c] The result was reported on a Japanese corpus. So it is not directly comparable to our work.

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

we could not compare with Kanungo *et al.* (2021) because they used a large private dataset consisting of 500,000 ad campaigns created on Amazon. Their model training is also time-expensive (over 20 days on an Nvidia V100 GPU).

Our approach is most similar to Misawa *et al.* (2020) in that we also employ an encoder–decoder framework with a description as the input. However, we differ from their work in two principled ways. Firstly, we use a more modern Transformer architecture (Vaswani *et al.* 2017), which enjoys the benefit of extensive pretraining and outperforms recurrent neural networks in most language generation benchmarks. We do not encounter the problem of generating generic slogans and out-of-vocabulary words (due to subword tokenisation). Therefore, the model is greatly simplified and can be trained using a standard cross-entropy loss. Secondly, we propose simple yet effective approaches to improve the truthfulness and diversity of generated slogans.

2.2 Sequence-to-sequence models

Sutskever *et al.* (2014) presented a seminal sequence learning framework using multilayer LSTM (Hochreiter and Schmidhuber 1997). The framework encodes the input sequence to a vector of fixed dimensionality, then decodes the target sequence based on the vector. This framework enables learning sequence-to-sequence (seq2seq)[d] tasks where the input and target sequence are of a different length. Sutskever *et al.* (2014) demonstrated that their simple framework achieved close to state-of-the-art performance in an English to French translation task.

The main limitation of Sutskever *et al.* (2014) is that the performance degrades drastically when the input sequence becomes longer. It is because of unavoidable information loss when compressing the whole input sequence to a fixed-dimension vector. Bahdanau *et al.* (2015) and Luong *et al.* (2015) overcame this limitation by introducing attention mechanism to LSTM encoder–decoder. The model stores a contextualised vector for each time step in the input sequence. During decoding, the decoder computes the attention weights dynamically to focus on different contextualised vectors. Attention mechanism overtook the previous state-of-the-art in English–French and English–German translation and yields much more robust performance for longer input sequences.

LSTM, or more generally recurrent neural networks, cannot be fully parallelised on modern GPU hardware because of an inherent temporal dependency. The hidden states need to be computed one step at a time. Vaswani *et al.* (2017) proposed a new architecture, the Transformer, which is based solely on multi-head self-attention and feed-forward layers. They also add positional encodings to the input embeddings to allow the model to use the sequence's order. The model achieved a new state-of-the-art performance, albeit taking a much shorter time to train than LSTM with attention mechanism.

Devlin *et al.* (2019) argued that the standard Transformer (Vaswani *et al.* 2017) suffers from the limitation that they are unidirectional and every token can only attend to previous tokens in the self-attention layers. To this end, they introduced BERT, a pretrained bidirectional Transformer by using a masked language model (MLM) pretraining objective. MLM masks some random tokens with a `[MASK]` token and provides a bidirectional context for predicting the masked tokens. Besides, Devlin *et al.* (2019) used the next sentence prediction task as an additional pretraining objective.

Despite achieving state-of-the-art results on multiple language understanding tasks, BERT does not make predictions autoregressively, reducing its effectiveness for generation tasks. Lewis *et al.* (2020) presented BART, a model combining a bidirectional encoder (similar to BERT) and an autoregressive decoder. This combination allows BART to capture rich bidirectional contextual representation and yield strong performance in language generation tasks. Besides MLM, Lewis *et al.* (2020) introduced new pretraining objectives, including masking text spans, token deletion,

[d] We use sequence-to-sequence and encoder-decoder interchangeably in this paper.

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

sentence permutation and document rotation. These tasks are particularly suitable for a seq2seq model like BART because there is no one-to-one correspondence between the input and target tokens.

Zhang *et al*. (2020a) employed an encoder–decoder Transformer architecture similar to BART. They introduced a novel pretraining objective specifically designed for abstractive summarisation. Instead of masking single tokens (like BERT) or text spans (like BART), they mask whole sentences (referred to as 'gap sentences') and try to reconstruct these sentences from their context. Zhang *et al*. (2020a) demonstrated that the model performs best when using important sentences selected greedily based on the ROUGE-F_1 score between the selected sentences and the remaining sentences. Their proposed model PEGASUS achieved state-of-the-art performance on all 12 summarisation tasks they evaluated. It also performed surprisingly well on a low-resource setting due to the relatedness of the pretraining task and abstractive summarisation.

While large-scale Transformer-based LMs demonstrate impressive text generation capabilities, users cannot easily control particular aspects of the generated text. Keskar *et al*. (2019) proposed CTRL, a conditional Transformer LM conditioned on control codes that influence the style and content. Control codes indicate the domain of the data, such as Wikipedia, Amazon reviews and subreddits focusing on different topics. Keskar *et al*. (2019) use naturally occurring words as control codes and prepend them to the raw text prompt. Formally, given a sequence of the form $x = (x_1, ..., x_n)$ and a control code c, CTRL learns the conditional probability $p_\theta(x_i|x_{<i}, c)$. By changing or mixing control codes, CTRL can generate novel text with very different style and content.

In this work, we use BART model architecture due to its flexibility as a seq2seq model and competitive performance on language generation tasks. We were also inspired by CTRL and applied a similar idea to generate slogans conditioned on additional attributes.

2.3 Truthfulness in language generation

While advanced seq2seq models can generate realistic text resembling human-written ones, they are usually optimised using a token-level cross-entropy loss. Researchers observed that a low training loss or a high ROUGE score do not guarantee the generated text is truthful with the source text (Cao *et al*. 2018; Scialom *et al*. 2019). Following previous work, we define *truthfulness* as the generated text can be verified through the source text without any external knowledge.

We did not find any literature specifically addressing truthfulness in slogan generation. Most prior works investigated abstractive summarisation because (1) truthfulness is critical in the summarisation task, and (2) the abstractive nature encourages the model to pull information from different parts of the source document and fuse them. Therefore, abstractive models are more prone to hallucination compared to extractive models (Durmus *et al*. 2020). Slogan generation and abstractive summarisation are analogous in that both tasks aim to generate a concise output text from a longer source text. Like summarisation, slogan generation also requires the generated content to be truthful. A prospect may feel annoyed or even be harmed by false information in advertising messages.

Prior work focused mostly on devising new metrics to measure the truthfulness between the source and generated text. Textual entailment (aka. natural language inference) is closely related to truthfulness. If a generated sequence can be inferred from the source text, it is likely to be truthful. Researchers have used textual entailment to rerank the generated sequences (Falke *et al*. 2019; Maynez *et al*. 2020) or remove hallucination from the training dataset (Matsumaru *et al*. 2020). Pagnoni *et al*. (2021) recently conducted a comprehensive benchmark on a large number of truthfulness evaluation metrics and concluded that entailment-based approaches yield the highest correlation with human judgement.

Another direction is to extract and represent the fact explicitly using information extraction techniques. Goodrich *et al*. (2019) and Zhu *et al*. (2021) extracted relation tuples using OpenIE

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

(Angeli *et al.* 2015), while Zhang *et al.* (2020b) used a domain-specific information extraction system for radiology reports. The truthfulness is then measured by calculating the overlap between the information extracted from the source and generated text.

In addition, researchers also employed QA-based approaches to measure the truthfulness of summaries. Eyal *et al.* (2019) generated slot-filling questions from the *source document* and measured how many of these questions can be answered from the generated summary. Scialom *et al.* (2019) optimised towards a similar QA-based metric directly using RL and demonstrated that it generated summaries with better relevance. Conversely, Durmus *et al.* (2020) and Wang *et al.* (2020) generated natural language questions from the *system-output summary* using a seq2seq question generation model and verified if the answers obtained from the source document agree with the answers from the summary.

Most recently, some work explored automatically correcting factual inconsistencies from the generated text. For example, Dong *et al.* (2020) proposed a model-agnostic post-processing model that either iteratively or autoregressively replaces entities to ensure semantic consistency. Their approach predicts a text span in the source text to replace an inconsistent entity in the generated summary. Similarly, Chen *et al.* (2021) modelled factual correction as a classification task. Namely, they predict the most plausible entity in the source text to replace each entity in the generated text that does not occur in the source text.

Our work is most similar to Dong *et al.* (2020) and Chen *et al.* (2021). However, their methods require performing additional predictions on each entity in the generated text using BERT. It drastically increases the latency. We decide on a much simpler approach of replacing each entity in both the source and target text with a unique mask token before training the model, preventing it from generating hallucinated entities in the first place. We can then perform a trivial dictionary lookup to replace the mask tokens with their original surface form.

2.4 Diversity in language generation

Neural LMs often surprisingly generate bland and repetitive output despite their impressive capability, a phenomenon referred to as neural text degeneration (Holtzman *et al.* 2019). Holtzman *et al.* (2019) pointed out that maximising the output sequence's probability is 'unnatural'. Instead, humans regularly use vocabulary in the low probability region, making the sentences less dull. While beam search and its variations (Reddy 1977; Li *et al.* 2016) improved over greedy encoding by considering multiple candidate sequences, they are still maximising the output probability by nature, and the candidates often differ very little from each other.

A common approach to improve diversity and quality of generation is to introduce randomness by sampling (Ackley *et al.* 1985). Instead of always choosing the most likely token(s) at each time step, the decoding algorithm samples from the probability distribution over the whole vocabulary. The shape of the distribution can be controlled using the temperature parameter. Setting the temperature to (0,1) shifts the probability mass towards the more likely tokens. Lowering the temperature improves the generation quality at the cost of decreasing diversity (Caccia *et al.* 2019).

More recently, top k-sampling (Fan *et al.* 2018) and nucleus sampling (Holtzman *et al.* 2019) were introduced to truncate the candidates before performing the sampling. Top k-sampling samples from a fixed most probable k candidate tokens, while nucleus (or top p) sampling samples from the most probable tokens whose probability sum is at least p. Nucleus sampling can dynamically adjust the top-p vocabulary size. When the probability distribution is flat, the top-p vocabulary size is larger, and when the distribution is peaked, the top-p vocabulary size is smaller. Holtzman *et al.* (2019) demonstrated that nucleus sampling outperformed various decoding strategies, including top-k sampling. Besides, the algorithm can generate text that matches the human perplexity by tuning the threshold p.

Welleck *et al.* (2019) argued that degeneration is not only caused by the decoding algorithm but also due to the use of maximum likelihood training loss. Therefore, they introduced an additional

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

unlikelihood training loss. Specifically, they penalise the model for generating words in previous context tokens and sequences containing repeating n-grams. The unlikelihood training enabled their model to achieve comparable performance as nucleus sampling using only greedy decoding.

It is worth noting that in language generation tasks, there is often a trade-off between relevance/quality and diversity (Gao *et al.* 2019; Zhang *et al.* 2021), both characteristics being crucial to slogan generation. Instead of relying on randomness, we generate syntactically diverse slogans with conditional training similar to CTRL (Keskar *et al.* 2019). Automatic and human evaluations confirmed that our method yields more diverse and interesting slogans than nucleus sampling.

3. Datasets

While large advertising agencies might have conducted hundreds of thousands of ad campaigns and have access to the historical ads with slogans (Kanungo *et al.* 2021), such a dataset is not available to the research community. Neither is it likely to be released in the future due to data privacy concerns.

On the other hand, online slogan databases such as Textart.ru[e] and Slogans Hub[f] contain at most hundreds to thousands of slogans, which are too few to form a training dataset, especially for a general slogan generator not limited to a particular domain. Besides, these databases do not contain company descriptions. Some even provide a list of slogans without specifying their corresponding company or product. They might be used to train a LM producing slogan-like utterance (Boigne 2020), but it will not be of much practical use because we do not have control over the generated slogan's content.

We observe that many company websites use their company name plus their slogan as the HTML page title. Examples are 'Skype | Communication tool for free calls and chat' and 'Virgin Active Health Clubs - Live Happily Ever Active'. Besides, many companies also provide a brief description in the 'description' field in the HTML <meta> tag[g]. Therefore, our model's input and output sequence can potentially be crawled from company websites.

We crawl the title and description field in the HTML <meta> tag using the Beautiful Soup library[h] from the company URLs in the Kaggle 7+ Million Company Dataset[i]. The dataset provides additional fields, but we utilise only the company name and URL in this work. The crawling took around 45 days to complete using a cloud instance with two vCPUs. Out of the 7M companies, we could crawl both the <meta> tag description and the page title for 1.4M companies. This dataset contains much noise due to the apparent reason that not all companies include their slogan in their HTML page title. We perform various cleaning/filtering steps based on various keywords, lexicographical and semantic rules. The procedure is detailed in Appendix A.

After all the cleaning and filtering steps, the total number of (description, slogan) pairs is 340k, at least two orders of magnitude larger than any publicly available slogan database. We reserve roughly 2% of the dataset for validation and test each. The remaining 96% is used for training (328k pairs). The validation set contains 5412 pairs. For the test set, the first author of this paper manually curated the first 1467 company slogans in the test set, resulting in 1000 plausible slogans (68.2%). The most frequent cases he filtered out are unattractive 'slogans' with a long list of products/services, such as 'Managed IT Services, Network Security, Disaster Recovery', followed by the cases where HTML titles containing alternative company names that failed to be delexicalised and

[e] http://www.textart.ru/database/slogan/list-advertising-slogans.html
[f] https://sloganshub.org/
[g] https://www.w3schools.com/tags/tag_meta.asp
[h] https://www.crummy.com/software/BeautifulSoup/
[i] https://www.kaggle.com/peopledatalabssf/free-7-million-company-dataset/

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

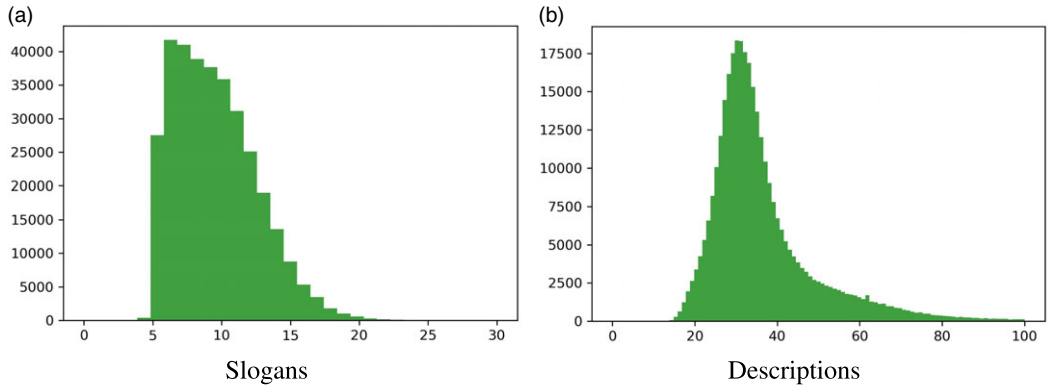

Figure 2. Distribution of the number of tokens in (a) slogans and (b) descriptions.

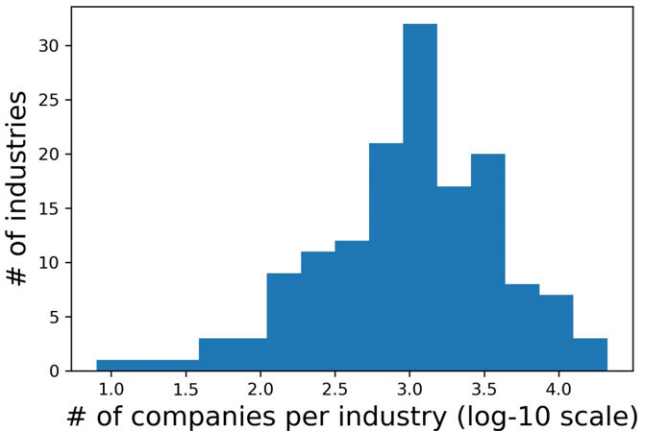

Figure 3. Distribution of the number of companies belonging to each industry in log-10 scale.

some other noisy content such as address. We publish our validation and manually curated test dataset for future comparisons[j].

We perform some data analysis on the training dataset to better understand the data. We first tokenise the dataset with BART's subword tokeniser. Figure 2 shows the distribution of the number of tokens in slogans and descriptions. While the sequence length of the description is approximately normally distributed, the length of slogans is right-skewed. It is expected because slogans are usually concise and contain few words. We choose a maximum sequence length of 80 for the description and 20 for the slogan based on the distribution.

The training dataset covers companies from 149 unique industries (based on the 'industry' field in the Kaggle dataset). Figure 3 shows the distribution of the number of companies belonging to each industry on a log-10 scale. As we can see, most industries contain between 10^2 (100) and $10^{3.5}$ (3162) companies. Table 1 shows the most frequent 10 industries with the number of companies and the percentage in the dataset. The large number of industries suggests that a model trained on the dataset will have observed diverse input and likely generalise to unseen companies.

Furthermore, we investigate the following questions to understand the nature and the abstractness of the task:

[j]https://github.com/YipingNUS/slogan-generation-dataset

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

Table 1. The most frequent 10 industries in the training dataset

Industry	# of companies	%
Information Technology and Services	21,149	6.4%
Marketing and Advertising	15,691	4.8%
Construction	12,863	3.9%
Computer Software	11,367	3.5%
Real Estate	10,207	3.1%
Internet	10,025	3.1%
Health, Wellness and Fitness	9513	2.9%
Financial Services	8480	2.6%
Automotive	8351	2.5%
Retail	8217	2.5%

(1) What percentage of the slogans can be generated using a purely extractive approach, that is, the slogan is contained in the description?
(2) What percentage of the unigram words in the slogans occur in the description?
(3) What percentage of the descriptions contain the company name? (We removed the company name from all the slogans).
(4) What percentage of the slogans and descriptions contain entities? What are the entity types?
(5) What percentage of the entities in the slogans do not appear in the description?
(6) Is there any quantitative difference between the validation and manually curated test set that makes either of them more challenging?

First, 11.2% of the slogans in both the validation and the test set are contained in the descriptions (we ignore the case when performing substring matching). It indicates that approximately 90% of the slogans require different degrees of abstraction. On average, 62.7% of the word unigrams in the validation set slogans are contained in their corresponding descriptions, while the percentage for the test set is 59.0%.

63.1% and 66.6% of the descriptions in the validation and test set contain the company name. It shows that companies tend to include their name in the description, and there is an opportunity for us to tap on this regularity.

We use Stanza (Qi *et al.* 2020) fine-grained named entity tagger with 18 entity types to tag all entities in the descriptions and slogans. Table 2 presents the percentage of text containing each type of entity[k]. Besides `ORGANIZATION`, the most frequent entity types are `GPE`, `DATE`, `CARDINAL`, `LOCATION` and `PERSON`. Many entities in the slogans do not appear in the corresponding description. It suggests that training a seq2seq model using the dataset will likely encourage entity hallucinations, which are commonly observed in abstractive summarisation. We show sample (description, slogan) pairs belonging to different cases in Table 3.

[k] Details of the entity types can be found in the Ontonotes documentation: https://catalog.ldc.upenn.edu/docs/LDC2013T19/OntoNotes-Release-5.0.pdf.

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

Table 2. The percentage of descriptions and slogans containing each type of entity. 'Slog - Desc' indicates entities in the slogan that are not present in the corresponding description

	Valid dataset			Test dataset		
Entity type	Desc	Slogan	Slog - Desc	Desc	Slogan	Slog - Desc
ORGANIZATION	65.6	31.3	27.8	63.9	30.2	29.7
GPE	36.7	19.4	7.5	33.5	20.2	7.5
DATE	16.4	1.3	1.0	18.2	1.9	1.4
CARDINAL	10.2	1.4	0.8	10.4	1.1	0.6
LOCATION	4.6	1.1	0.6	4.6	1.1	0.7
PERSON	4.2	2.5	1.6	3.3	1.3	0.9
PRODUCT	4.2	0.2	0.1	4.2	0.4	0.4
NORP	2.6	0.9	0.4	3.8	0.6	0.1
FACILITY	2.5	0.5	0.4	2.9	0.1	0.1
TIME	2.0	0.02	–	1.4	–	–
WORK OF ART	1.5	0.4	0.3	1.7	0.6	0.5
PERCENT	1.3	0.09	0.09	1.9	–	–
ORDINAL	1.3	0.2	0.1	1.4	0.3	0.2
MONEY	0.7	0.2	0.1	0.8	–	–
QUANTITY	0.5	–	–	0.4	0.1	–
EVENT	0.5	0.2	0.2	0.3	0.8	0.8
LAW	0.3	–	–	0.3	–	–
LANGUAGE	0.3	0.09	0.02	0.2	0.1	–

The only notable difference that might make the test dataset more challenging is that it contains a slightly higher percentage of unigram words not occurring in the description than the validation dataset (41% vs. 37.3%). However, this difference is relatively small, and we believe the performance measured on the validation dataset is a reliable reference when a hand-curated dataset is not available.

4. Model

We apply a Transformer-based seq2seq model to generate slogans. The model's input is a short company description. We choose BART encoder–decoder model (Lewis *et al*. 2020) with a bidirectional encoder and an autoregressive (left-to-right) decoder. BART enjoys the benefit of capturing bidirectional context representation like BERT and is particularly strong in language generation tasks.

We use DistilBART[1] with 6 layers of encoders and decoders each and 230M parameters. The model was a distilled version of BART_{LARGE} trained by the HuggingFace team, and its architecture

[1] https://huggingface.co/sshleifer/distilbart-cnn-6-6

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

Table 3. Sample (description, slogan) pairs belonging to different cases from the validation set. We highlight the exact match words in bold

Remark	Slogan	Description
Slogan in desc	Total Rewards Software	Market Total Rewards to Employees and Candidates with **Total Rewards Software**. We provide engaging Total Compensation Statements and Candidate Recruitment.
100% unigrams in desc	Algebra Problem Solver	Free math **problem solver** answers your **algebra** homework questions with step-by-step explanations.
57% unigrams in desc	Most Powerful Lead Generation Software for Marketers	**Powerful lead generation software** that converts abandoning visitors into subscribers with our dynamic marketing tools and Exit Intent technology.
33% unigrams in desc	Business Process Automation	We help companies become more efficient by automating processes, impact **business** outcomes with actionable insights & capitalize on growth with smart applications.
0% unigrams in desc	Build World-Class Recreation Programs	Easily deliver personalised activities that enrich the lives of residents in older adult communities. Save time and increase satisfaction.
Entities in desc	Digital Agency In $\mathbf{Auckland}_{[GPE]}$ & $\mathbf{Wellington}_{[GPE]}$, New Zealand	Catch Design is an independent digital agency in **Auckland** and **Wellington**, NZ. We solve business problems through the fusion of design thinking, creativity, innovation and technology.
Entities not in desc	Leading Corporate Advisory Services Provider In $\mathbf{Singapore}_{[GPE]}$ & $\mathbf{HongKong}_{[GPE]}$	Offers Compliance Advisory services for Public listed companies, Private companies, NGOs, Offshore companies and Limited Liability Partnerships (LLPs).

is equivalent to BART_{BASE}. We choose this relatively small model to balance generation quality and latency because our application requires generating multiple variations of slogans in real time in a web-based user interface.

The seq2seq slogan generation from the corresponding description is analogous to abstractive summarisation. Therefore, we initialise the model's weights from a *fine-tuned* summarisation model on the CNN/DailyMail dataset (Hermann *et al.* 2015) instead of from a pretrained model using unsupervised learning objectives. We freeze up to the second last encoder layer (including the embedding layer) and fine-tune the last encoder layer and the whole decoder. Based on our experiments, it significantly reduced the RAM usage without sacrificing performance.

5. Generating truthful slogans

As we highlighted in Section 2.3, generating slogans containing false or extraneous information is a severe problem for automatic slogan generation systems. In this section, we propose two approaches to improve the quality and truthfulness of generated slogans, namely delexicalising company names (Section 5.1) and masking named entities (Section 5.2).

5.1 Company name delexicalisation

Slogans should be concise and not contain extraneous information. Although we removed the company names from all slogans during preprocessing (described in Appendix A), we observe that a baseline seq2seq model often copies the company name from the description to the slogan. Table 4 shows two such example generated by the seq2seq model. Both examples seem to be purely extractive except for changing the case to title case. The second example seems especially repetitive

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

Table 4. Examples of generated slogans containing the company name

Company Name:	Eftpos Warehouse
Description:	The latest Eftpos Warehouse & Point of Sale tech for the lowest prices. Flexible monthly rental options with the backing of our dedicated, on-call support team.
Generated Slogan:	Eftpos Warehouse & Point of Sale Tech
Company Name:	MCB Financial Services
Description:	Financial Advisers Norwich, Norfolk - MCB Financial Services Norwich are committed to helping you with your financial needs.
Generated Slogan:	Financial Advisers Norwich, Norfolk - MCB Financial Services Norwich

Table 5. An example description before and after performing delexicalisation

Company:	Atlassian Corporation Plc
Description:	Millions of users globally rely on **Atlassian** products every day for improving software development, project management, collaboration and code quality.
Surface Form:	Atlassian
Delexicalised Description:	Millions of users globally rely on <company> products every day for improving software development, project management, collaboration and code quality.

and is not a plausible slogan. As shown in Section 3, over 60% of the descriptions contain the company name. Therefore, a method is necessary to tackle this problem.

We apply a simple treatment to prevent the model from generating slogans containing the company name – delexicalising company name mentions in the description and replacing their surface text with a generic mask token <company>. After the model generates a slogan, any mask token is substituted with the original surface text[m].

We hypothesise that delexicalisation helps the model in two ways. Firstly, it helps the model avoid generating the company name by masking it in the input sequence. Secondly, the mask token makes it easier for the model to focus on the surrounding context and pick salient information to generate slogans.

The company name is readily available in our system because it is required when any new advertiser registers for an account. However, we notice that companies often use their shortened names instead of their official/legal name. Examples are 'Google LLC' almost exclusively referred to as 'Google' and 'Prudential Assurance Company Singapore (Pte) Limited' often referred to as 'Prudential'. Therefore, we replace the longest prefix word sequence of the company name occurring in the description with a <company> mask token. The process is illustrated in Algorithm 1 (we omit the details handling the case and punctuations in the company name for simplicity).

Besides the delexicalised text, the algorithm also returns the surface text of the delexicalised company name, which will replace the mask token during inference. It is also possible to use a more sophisticated delexicalisation approach, such as relying on a knowledge base or company directory such as Crunchbase to find alternative company names. However, the simple substitution algorithm suffices our use case. Table 5 shows an example description before and after delexicalisation.

[m] Since the <company> token never occurs in slogans in our dataset, we have not observed a single case where the model generates a sequence containing the <company> token. We include the substitution for generality.

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

Algorithm 1: Prefix matching for delexicalising company names.

```
Input: company_name, text, MASK_TOKEN
Result: delexicalised_text, surface_form
delexicalised_text = text;
surface_form = company_name + " ";
while surface_form.contains(" ") do
    surface_form = surface_form.substring(0, surface_form.lastIndexOf(" "));
    if text.contains(surface_form) then
        delexicalised_text = text.replace(surface_form, MASK_TOKEN);
        break;
    end
end
```

5.2 Entity masking

Introducing irrelevant entities is a more challenging problem compared to including company names in the slogan. It has been referred to as entity hallucination in the abstractive summarisation literature (Nan *et al*. 2021). In a recent human study, Gabriel *et al*. (2021) showed that entity hallucination is the most common type of factual errors made by Transformer encoder–decoder models.

We first use Stanza (Qi *et al*. 2020) to perform named entity tagging on both the descriptions and slogans. We limit to the following entity types because they are present in at least 1% of both the descriptions and slogans based on Table 2: `GPE`, `DATE`, `CARDINAL`, `LOCATION` and `PERSON`. Additionally, we include `NORP` (nationalities/religious/political group) because a large percentage of entities of this type in the slogan can be found in the corresponding description. We observe that many words are falsely tagged as `ORGANIZATION`, which is likely because the slogans and descriptions often contain title-case or all-capital texts. Therefore, we exclude `ORGANIZATION` although it is the most common entity type.

Within each (description, slogan) pair, we maintain a counter for each entity type. We compare each new entity with all previous entities of the same entity type. If it is a substring of a previous entity or vice versa, we assign the new entity to the previous entity's ID. Otherwise, we increment the counter and obtain a new ID. We replace each entity mention with a unique mask token `[entity_type]` if it is the first entity of its type or `[entity_type id]` otherwise. We store a reverse mapping and replace the mask tokens in the generated slogan with the original entity mention. We also apply simple rule-based post-processing, including completing the closing bracket (`']'`) if it is missing and removing illegal mask tokens and mask tokens not present in the mapping[n].

During experiments, we observe that when we use the original upper-cased entity type names, the seq2seq model is prone to generating illegal tokens such as `[gPE]`, `[GPA]`. Therefore, we map the tag names to a lower-cased word consisting of a single token (as tokenised by the pretrained tokeniser). The mapping we use is {`GPE`:country, `DATE`:date, `CARDINAL`:number, `LOCATION`:location, `PERSON`:person, `NORP`:national}. Table 6 shows an example of the entity masking process.

As shown in Table 2, a sizeable proportion of the entities in the slogans are not present in the description. We discard a (description, slogan) pair from the *training* dataset if any of the entities in the slogan cannot be found in the description. This procedure removes roughly 10% of the training data but encourages the model to generate entities present in the source description instead of fabricated entities. We do not apply filtering to the validation and test set so that the result is comparable with other models.

[n] We also remove all the preceding stop words before the removed mask token. In most cases, they are prepositions or articles such as '**from the** `[country]`' or '**in** `[]`'.

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

Table 6. An example description and slogan before and after entity masking. Note that the word 'Belgian' in the slogan is replaced by the same mask token as the same word in the description

Description:	PR-Living **Belgium** family-owned furniture brand with production facilities in **Waregem** where it brings the best of **Belgian**-inspired Design Upholstery & Furniture pieces to the global consumers.
Slogan:	A **Belgian** furniture brand
Entities:	'Belgium': `GPE`, 'Waregem': `GPE`, 'Belgian': `NORP`
Masked Description:	PR-Living `[country]` family-owned furniture brand with production facilities in `[country1]` where it brings the best of `[national]`-inspired Design Upholstery & Furniture pieces to the global consumers.
Masked Slogan:	A `[national]` furniture brand
Reverse Mapping:	`[country]`: 'Belgium', `[country1]`: 'Waregem', `[national]`: 'Belgian'

Table 7. The most frequent 10 POS tag sequences for slogans in the training dataset

POS tag sequence	Frequency	Example
NNP NNP NNP	12,892	Emergency Lighting Equipment
NNP NNP NNP NNP	5982	Personal Injury Lawyers Melbourne
NNP NNP NNPS	4217	Rugged Computing Solutions
JJ NN NN	3109	Flexible Office Space
NNP NNP NN	2789	Bluetooth Access Control
NNP NNP NNS	2632	Manchester Law Firms
NNP NNP NNP CC NNP NNP	2190	Local Programmatic Advertising & DSP Platform
NNP NNP CC NNP NNP NNP	2157	Retro Candy & Soda Pop Store
NN NN NN	2144	Footwear design consultancy
NNP NNP NNP NNP NNP	1662	Commercial Construction Company New England

6. Generating diverse slogans with syntactic control

Generating **diverse** slogans is crucial to avoid ads fatigue and enable personalisation. However, we observe that given one input description, our model tends to generate slogans similar to each other, such as replacing some words or using a slightly different expression. Moreover, the outputs are often simple noun phrases that are not catchy.

To investigate the cause, we perform POS tagging on all the slogans in our training dataset. Table 7 shows the most frequent POS tag sequences among the slogans[o]. Only one (#46) out of the top 50 POS tag sequences is not a noun phrase (VB PRP$ NN, e.g., Boost Your Business). It motivates us to increase the generated slogans' diversity using syntactic control.

Inspired by CTRL (Keskar *et al.* 2019), we modify the generation from $P(slogan|description)$ to $P(slogan|description, ctrl)$ by conditioning on an additional syntactic control code. To keep the

[o] We refer readers to https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html for the description of each POS tag.

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

Table 8. Full list of syntactic control codes

Code	Frequency	Meaning
NN	208,061	All types of nouns
JJ	44,926	All types of adjectives and adverbs
VB	37,331	Verbs of any form or tense
DT	17,645	Determiners
PR	8484	Personal or possessive pronouns
OTHER	7644	Any other tags not included above, such as numbers, prepositions and question words

cardinality small, we use the coarse-grained POS tag[P] of the first word in the slogan as the control code. Additionally, we merge adjectives and adverbs and merge all the POS tags that are not among the most frequent five tags. Table 8 shows the full list of control codes.

While we can use the fine-grained POS tags or even the tag sequences as the control code, they have a long-tail distribution, and many values have only a handful of examples, which are too few for the model to learn from. Munigala *et al.* (2018) applied a similar idea as ours to generate persuasive text starting with a verb. However, they apply rules to restrict a generic LM to start with a verb. We apply conditional training to learn the characteristics of slogans starting with words belonging to various POS tags.

We prepend the control code to the input sequence with a special `</s>` token separating the control code and the input sequence. We use the control code derived from the target sequence during training while we randomly sample control codes during inference to generate syntactically diverse slogans. Our method differs from Keskar *et al.* (2019) in two slight ways: 1) CTRL uses an autoregressive Transformer similar to GPT-2 (Radford *et al.* 2019) while we use an encoder-decoder Transformer with a bidirectional encoder. 2) The control codes were used during pretraining in CTRL while we prepend the control code only during fine-tuning for slogan generation.

7. Experiments

We conduct a comprehensive evaluation of our proposed method. In Section 7.1, we conduct a quantitative evaluation and compare our proposed methods with other rule-based and encoder-decoder baselines in terms of ROUGE -1/-2/-L F_1 scores. We report the performance of a larger model in Section 7.2. We specifically study the truthfulness and diversity of the generated slogans in Sections 7.3 and 7.4. Finally, we conduct a fine-grained human evaluation in Section 7.5 to further validate the quality of the slogans generated by our model.

We use the DistilBART and BART_{LARGE} implementation in the Hugging Face library (Wolf *et al.* 2020) with a training batch size of 64 for DistilBART and 32 for BART_{LARGE}. We use a cosine decay learning rate with warm-up (He *et al.* 2019) and a maximum learning rate of 1e-4. The learning rate is chosen with Fastai's learning rate finder (Howard and Gugger 2020).

[P] Corresponding to the first two characters of the POS tag, thus ignoring the difference between proper versus common noun, plural versus singular, different verb tenses and the degree of adjectives and adverbs.

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

We train all BART models for three epochs. Based on our observation, the models converge within around 2–3 epochs. We use greedy decoding unless otherwise mentioned. We also add a repetition penalty $\theta = 1.2$ following Keskar *et al.* (2019).

7.1 Quantitative evaluation

We leave the diversity evaluation to Section 7.4 because we have only a single reference slogan for each input description in our dataset, which will penalise systems generating diverse slogans. We compare our proposed method with the following five baselines:

- *first sentence*: predicting the first sentence from the description as the slogan, which is simple but surprisingly competitive for document summarisation (Katragadda *et al.* 2009). We use the sentence splitter in the Spacy library[q] to extract the first sentence.
- *first-k words*: predicting the first-*k* words from the description as the slogan. We choose *k* that yields the highest ROUGE-1 F_1 score on the validation dataset. We add this baseline because the first sentence of the description is usually much longer than a typical slogan.
- *Skeleton-Based* (Tomašic *et al.* 2014): a skeleton-based slogan generation system using genetic algorithms and various heuristic-based scoring functions. We sample a random compatible slogan skeleton from the training dataset and realise the slogan with keywords extracted from the company description. We follow Tomašic *et al.* (2014)'s implementation closely. However, we omit the database of frequent grammatical relations and the bigram function derived from Corpus of Contemporary American English because the resources are not available.
- *Encoder–Decoder* (Bahdanau *et al.* 2015): a strong and versatile GRU encoder–decoder baseline. We use identical hyperparameters as Misawa *et al.* (2020) and remove the reconstruction loss and copying mechanism to make the models directly comparable. Specifically, the model has a single hidden layer for both the bidirectional encoder and the autoregressive decoder. We apply a dropout of 0.5 between layers. The embedding and hidden dimensions are 200 and 512 separately, and the vocabulary contains 30K most frequent words. The embedding matrix is randomly initialised and trained jointly with the model. We use Adam optimiser with a learning rate of 1e-3 and train for 10 epochs (The encoder–decoder models take more epochs to converge than the Transformer models, likely because the models are randomly initialised).
- *Pointer-Generator* (See *et al.* 2017): encoder–decoder model with copying mechanism to handle unknown words. Equivalent to Misawa *et al.* (2020) with the reconstruction loss removed.
- Misawa *et al.* (2020): a GRU encoder–decoder model for slogan generation with additional reconstruction loss to generate distinct slogans and copying mechanism to handle unknown words.

Table 9 presents the ROUGE -1/-2/-L scores of various models on both the validation and the manually curated test dataset.

The first-*k* words baseline achieved a reasonable performance, showing a certain degree of overlap between slogans and descriptions. Figure 4 shows how the first-*k* words baseline's ROUGE F_1 scores change by varying *k*. It is obvious that not the larger *k*, the better. The best ROUGE scores are achieved when *k* is in the range (9, 12). The first-*k* words baseline also achieved higher ROUGE scores than the first sentence baseline, although it may output an incomplete phrase due to the truncation.

[q] https://spacy.io/

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

Table 9. The ROUGE F_1 scores for various models on the validation and test dataset. DistilBART denotes the base model introduced in Section 4. '+delex' and '+ent' means adding company name delexicalisation (Section 5.1) and entity masking (Section 5.2)

	Valid dataset			Test dataset		
	R1	R2	RL	R1	R2	RL
First sentence	26.12	13.03	23.88	25.50	12.73	23.47
First-k words ($k = 11$)	27.50	13.72	25.33	25.76	12.68	24.02
Skeleton-based	16.09	1.62	14.01	16.61	1.79	14.72
Encoder-Decoder	24.85	9.38	24.01	23.91	9.31	23.28
Pointer-Generator	26.42	10.15	25.63	26.24	10.67	25.65
Misawa *et al*. (2020)	24.14	9.19	23.37	26.01	10.00	25.39
DistilBART	36.74	18.87	33.97	34.95	17.38	32.47
DistilBART -finetuning	25.12	11.83	23.20	24.34	11.46	22.61
DistilBART -pretraining	22.85	6.49	20.70	22.16	6.29	20.78
DistilBART+delex	37.37	19.51	34.69	35.06	17.79	32.52
DistilBART+delex+ent	**37.76**	**19.69**	**35.17**	**35.58**	**18.47**	**33.32**

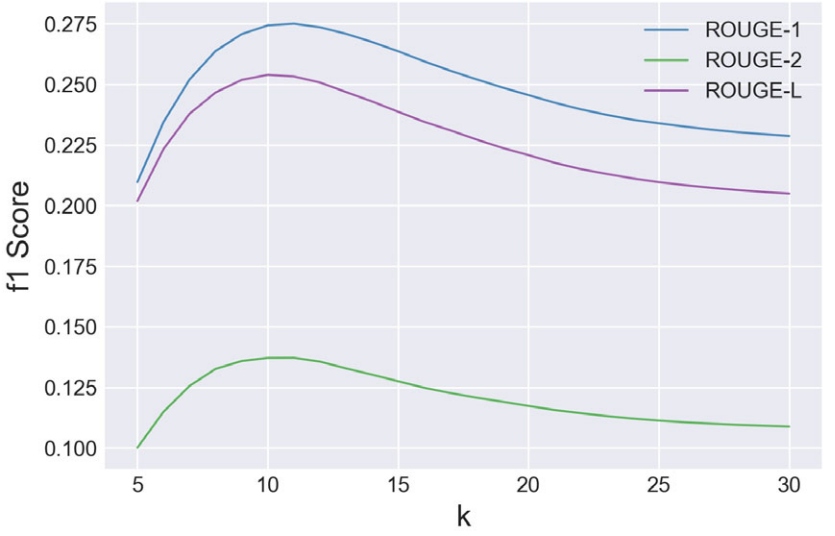

Figure 4. The ROUGE -1/-2/-L scores of the first-k word baseline by varying k.

The skeleton-based method had the worst performance among all baselines. While it often copies important keywords from the description, it is prone to generating ungrammatical or non-sensical output because it relies on POS sequence and dependency parse skeletons and ignores the context.

Comparing the three GRU encoder–decoder baselines, it is clear that the copying mechanism in Pointer-Generator improved the ROUGE scores consistently. However, the reconstruction loss introduced in Misawa *et al*. (2020) seems to reduce performance. We hypothesise that the slogan is much shorter than the input description. Therefore, reconstructing the description from the slogan may force the model to attend to unimportant input words. Overall, the Pointer-Generator

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

Table 10. Sample generated slogans by various systems. 'Gold' is the original slogan of the company. The DistilBART model uses both delexicalisation and entity masking

Gold:	Fast, Fresh & Tasty Mexican Food
First-k words:	We may not be the only burrito in town, but we've
Skeleton-based:	Bar to Your Burrito in Town
Pointer-Generator:	The World 's First Class Action Club
DistilBART:	The Best Burrito in Town
Gold:	A better UK energy supplier
First-k words:	Welcome to Powershop, a better gas and energy supplier. We offer
Skeleton-based:	Your Gas in Competitive Electricity, Energy Deals, Offer and Website
Pointer-Generator:	Gas and Electricity Supplier
DistilBART:	Gas and Energy Supplier
Gold:	Saigon Food Tours and City Tours Led by Women
First-k words:	Top-ranked Ho Chi Minh City food tours from the first company
Skeleton-based:	Food Company & Culture Tours Female
Pointer-Generator:	Food Tours & Food Tours
DistilBART:	Food Tours in Vietnam
Gold:	Top Engineering Colleges In Tamilnadu
First-k words:	top Engineering colleges in Tamil Nadu based on 2020 ranking. Get
Skeleton-based:	We Info Colleges!
Pointer-Generator:	Engineering College in Dehradun Uttarakhand
DistilBART:	Top Engineering Colleges in Tamil Nadu

baseline's performance is on par with the first-*k* words baseline but pales when comparing with any Transformer-based model.

Both delexicalisation and entity masking further improved DistilBART's performance. The final model achieved a ROUGE -1/-2/-L score of 35.58/18.47/33.32 on the curated test set, outperforming the best GRU encoder–decoder model by almost 10% in ROUGE score.

Table 10 provides a more intuitive overview of various models' behaviour by showing the generated slogans from randomly sampled company descriptions. We can observe that while the first-*k* words baseline sometimes has substantial word overlap with the original slogan, its style is often different from slogans. Pointer-Generator and DistilBART sometimes generate similar slogans. However, Pointer-Generator is more prone to generating repetitions, as in the third example. It also hallucinates much more. In the first example, the company is a Mexican restaurant. The slogan generated by Pointer-Generator is fluent but completely irrelevant. In the last example, it hallucinated the location of the school, while DistilBART preserved the correct information.

We report the result of two additional baselines to isolate the impact of fine-tuning and pretraining:

- *DistilBART -finetuning*: a DistilBART model fine-tuned on CNN/DM *summarisation* task. We set the maximum target length to 15 (if we do not limit the maximum length, the model tends to copy the entire description because the summaries in the CNN/DM dataset are much longer than slogans).
- *DistilBART -pretraining*: a DistilBART model trained from scratch on slogan generation using randomly initialised weights. We follow the exact training procedure as DistilBART except training the model longer for eight epochs till it converges.

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

Table 11. The ROUGE F_1 scores by scaling up the model size. Both models use delexicalisation and entity masking

			Valid dataset			Test dataset		
	params	epoch time	R1	R2	RL	R1	R2	RL
DistilBART	230M	45 mins	**37.76**	**19.69**	**35.17**	**35.58**	**18.47**	**33.32**
$\text{BART}_{\text{LARGE}}$	406M	86 mins	35.36	17.71	32.95	33.43	16.51	31.34

The model trained on CNN/DM dataset tends to generate a verbatim copy of the description till it reaches the maximum token length. It demonstrates that despite the similarity between abstractive summarisation and slogan generation, a pretrained summarisation model cannot generate plausible slogans in a zero-shot setting.

On the other hand, DistilBART trained from scratch has much worse training loss and ROUGE scores, highlighting the importance of pretraining. In addition, it tends to hallucinate much more and sometimes generates repetitions. Interestingly, its performance was worse than all of the GRU encoder–decoder baselines. We conjecture that it is because the larger Transformer model requires more data when trained from scratch.

7.2 Larger model results

Following Lewis *et al.* (2020) and Zhang *et al.* (2020a), we report the performance of a larger model, BART_{LARGE}[r]. Compared to DistilBART, BART_{LARGE} has both more layers (L: 6 → 12) and a larger hidden size (H: 768 → 1024). We follow the exact training procedure as DistilBART. Table 11 compares the performance of DistilBART and BART_{LARGE}.

We were surprised to observe that BART_{LARGE} underperformed the smaller DistilBART model by roughly 2% ROUGE score. During training, BART_{LARGE} had a lower training loss than DistilBART, but the validation loss plateaued to roughly the same value, suggesting that the large model might be more prone to overfitting the training data. We did not conduct extensive hyperparameter tuning and used the same learning rate as DistilBART. Although we cannot conclude that DistilBART is better suited for this task, it seems that using a larger model does not always improve performance.

7.3 Truthfulness evaluation

In this section, we employ automatic truthfulness evaluation metrics to validate that the methods proposed in Section 5 indeed improved the truthfulness. As briefed in Section 2.3, there are mainly three categories of automatic truthfulness evaluation metrics, namely entailment, information extraction and QA. We focus on entailment-based metrics because (1) they yield the highest correlation with human judgement based on a recent benchmark (Pagnoni *et al.* 2021), (2) the slogans are often very short and sometimes do not contain a predicate, making it impossible to automatically generate questions for a QA-based approach and extract (subject, verb and object) tuples for an information extraction-based approach.

The first model we use is an entailment classifier fine-tuned on the Multi-Genre NLI (MNLI) dataset (Williams *et al.* 2018) following Maynez *et al.* (2020). However, we use a fine-tuned RoBERTa_{LARGE} checkpoint[s] (Liu *et al.* 2019) instead of BERT_{LARGE} (Devlin *et al.* 2019), since

[r] https://huggingface.co/facebook/bart-large-cnn
[s] https://huggingface.co/roberta-large-mnli

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

Table 12. The truthfulness scores of the baseline distillBART model and our proposed method (the numbers are in per cent). The p-value of a two-sided paired t-test is shown in the bracket

	Valid dataset		Test dataset	
	Entailment	FactCC	Entailment	FactCC
DistilBART	75.89	70.23	70.87	71.21
DistilBART+delex+ent	**83.25** (2.3e-63)	**73.09** (5.1e-6)	**81.61** (1.0e-21)	**75.71** (8.4e-4)

it achieved higher accuracy on the MNLI dataset (90.2 vs. 86.6). We calculate the entailment probability between the input description and the generated slogan to measure truthfulness.

The second model we use is a pretrained FactCC (Kryscinski *et al*. 2020) classifier, which predicts whether a generated summary is consistent with the source document. It was trained on a large set of synthesised examples by adding noise into reference summaries using manually defined rules such as entity or pronoun swap. FactCC is the best-performing metric in Pagnoni *et al*. (2021)'s benchmark. It was also used in several subsequent works as the automatic truthfulness evaluation metric (Cao *et al*. 2020; Dong *et al*. 2020). We use the predicted probability for the category 'consistent' to measure truthfulness.

Table 12 presents the mean entailment and FactCC scores for both the validation and the test dataset. Both metrics suggest that our proposed method yields more truthful slogans w.r.t the input descriptions than a DistilBART baseline with strong statistical significance.

Compared to the result in Section 7.1, there is a larger gap between our proposed method and the baseline DistilBART model. It is likely because n-gram overlap metrics like ROUGE are not very sensitive to local factual errors. For example, suppose the reference sequence is 'Digital Marketing Firm in New *Zealand*', and the predicted sequence is 'Digital Marking Firm in New *Columbia*', it will receive a high ROUGE-1/-2/-L score of 83.3/80.0/83.3. However, entailment and factuality models will identify such factual inconsistencies and assign a very low score.

7.4 Diversity evaluation

In Section 6, we proposed a method to generate syntactically diverse slogans using control codes. First, we want to evaluate whether the control codes are effective in the generation. We calculate the *ctrl accuracy*, which measures how often the first word in the generated slogan agrees with the specified POS tag.

We apply each of the six control codes to each input in the test set and generate various slogans using greedy decoding. We then apply POS tagging on the generated slogans and extract the coarse-grained POS tag of the first word in the same way as in Section 6. We count it as successful if the coarse-grained POS tag matches the specified control code. Table 13 presents the ctrl accuracy for each of the control codes.

The control code distribution in our training dataset is very skewed, as shown in Table 8. The most frequent code (NN) contains more than 27 times more data than the least frequent code (OTHER). Therefore, we conducted another experiment by randomly upsampling examples with codes other than NN to 100k. We then trained for one epoch instead of three epochs to keep the total training steps roughly equal. We show the result in the second row of Table 13.

Besides, we compare with the nucleus sampling (Holtzman *et al*. 2019) baseline. We use top-$p = 0.95$ following Holtzman *et al*. (2019), because it is scaled to match the human perplexity[t]. We generate an equal number of slogans (six) as our method, and the result is presented in the third

[t]We use the default temperature of 1.0 and disable the top-*k* filter.

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

Table 13. The syntactic control accuracy, diversity and abstractiveness scores of various methods. The best score for each column is highlighted in bold. All models use neither delexicalisation nor entity masking to decouple the impact of different techniques

	Ctrl accuracy							
	NN	JJ	VB	DT	PR	OTHER	Diversity	Abstractive
W/O upsampling	**92.56**	37.12	**61.47**	**93.96**	**97.28**	**90.64**	**46.69**	**45.04**
Upsampling	91.14	**42.35**	48.69	71.83	96.88	55.53	44.81	43.85
Nucleus sampling	70.32	11.27	7.85	5.23	0.80	1.31	27.97	27.01

row of Table 13. We note that since nucleus sampling does not condition on the control code, the ctrl accuracies are not meant to be compared directly with our method but serve as a random baseline without conditional training.

We calculate the diversity as follows: for each set of generated slogans from the same input, we count the total number of tokens and unique tokens. We use Spacy's word tokenisation instead of the subword tokenisation. Besides, we lowercase all words, so merely changing the case will not be counted towards diversity. The diversity score for each set is the total number of unique tokens divided by the total number of tokens. We average the diversity scores over the whole test set to produce the final diversity score. We note that a diversity score of close to 100% is unrealistic because important keywords and stop words will and should occur in various slogans. However, a diversity score of close to 1/6 (16.67%) indicates that the model generates almost identical slogans and has very little diversity.

The result shows that our method achieved close to perfect ctrl accuracy except for the control code JJ and VB. Although some control codes like PR and OTHER have much fewer examples, they also have fewer possible values and are easier to learn than adjectives and verbs (e.g., there are a limited number of pronouns). The strong syntactic control accuracy validated recent studies' finding that pretrained LMs capture linguistic features internally (Tenney *et al.* 2019; Rogers *et al.* 2020).

Upsampling seems to help with neither the ctrl accuracy nor the diversity. Compared with our method, nucleus sampling has much lower diversity. Although it performs sampling among the top-p vocabulary, it will almost always sample the same words when the distribution is peaked. Increasing the temperature to above 1.0 can potentially increase the diversity, but it will harm the generation quality and consistency (Holtzman *et al.* 2019).

In addition, we calculate the abstractiveness as the number of generated slogan tokens that are not present in the input description divided by the number of generated slogan tokens, averaging over all candidates and examples in the test set. We can see that as a by-product of optimising towards diversity, our model is also much more abstractive.

Finally, we invite an annotator to manually assess the quality of the generated slogans[u]. We randomly sample 50 companies from the test set and obtain the 6 generated slogans from both our proposed method and nucleus sampling, thus obtaining 300 slogan pairs. We then ask the annotator to indicate which slogan is better (with the 'can't decide' option). We randomised the order of the slogans to eliminate positional bias. We present the annotation UI in appendix C and the annotation result in Table 14.

All control codes except 'NN' yielded significantly better slogans than the nucleus sampling baseline with $p = 0.05$. It is expected because 'NN' is most common in the dataset, and using the control code 'NN' will yield similar output as greedy decoding or nucleus sampling. While

[u] The annotator is an NLP researcher who is proficient in English and was not involved in the development of this work.

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

Table 14. Pair-wise evaluation result of each control code versus the nucleus sampling baseline. The p-value is calculated using two-sided Wilcoxon signed-rank test. 'Better' means the annotator indicates that the slogans generated by our method is better than nucleus sampling

Code	Better	Can't decide	Worse	*p*-Value
NN	28	3	19	0.189
JJ	32	3	15	0.013
VB	36	0	14	1.86e-03
DT	41	1	8	2.43e-06
PR	37	0	13	6.89e-04
OTHER	39	0	11	7.50e-05
Overall	213	7	80	7.85e-15

Table 15. Generated slogans with different control codes (randomly sampled)

Desc:	Helping eCommerce business growing their sales & revenues. Specialist in product feeds, shopping ads, conversion optimisation, SEO and website personalisation.
NN:	eCommerce Business Growth & Revenue Optimization Experts
JJ:	Ecommerce Marketing Agency in London & Essex
VB:	Helping eCommerce Business Grow Their Sales & Revenues
DT:	The eCommerce Experts
PR:	Your eCommerce Partner for Growth & Success!
OTHER:	How to Grow Your Business with eCommerce
Desc:	We are experts in: Web/Mobile/Desktop apps Development. Innovative technologies.
NN:	Web and Mobile App Development Company in India
JJ:	Innovative Technologies. Web and Mobile Apps Development Company
VB:	Leading Mobile App Development Company in India
DT:	Achieving Digital Transformation in the Cloud with Mobile Apps Development
PR:	We are experts in mobile apps development
OTHER:	Where technology meets creativity

Munigala *et al.* (2018) claimed that sentences starting with a verb are more persuasive, sentences starting with other POS tags may also have desirable characteristics for slogans. For example, starting with an adjective makes it more vivid; starting with a determiner makes it more assertive; starting with a pronoun makes it more personal. Surprisingly, the annotator also rated slogans generated with the control code 'OTHER' highly despite it groups many long-tail POS tags. The 'OTHER' control code often generates slogans starting with a question word, an ordinal number (e.g., '#1') or the preposition 'for' (e.g., 'For All Your Pain Relief Needs').

To give the reader a better sense of the system's behaviour, we present samples the system generated with different control codes in Table 15. We can see that the first word in the slogan may not always match the POS tag specified by the control code. However, the generated slogans are diverse in both syntactic structure and content.

Besides generating more diverse and higher quality slogans, another principal advantage of our approach over nucleus sampling is that we have more control over the syntactic structure of the generated slogan instead of relying purely on randomness.

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

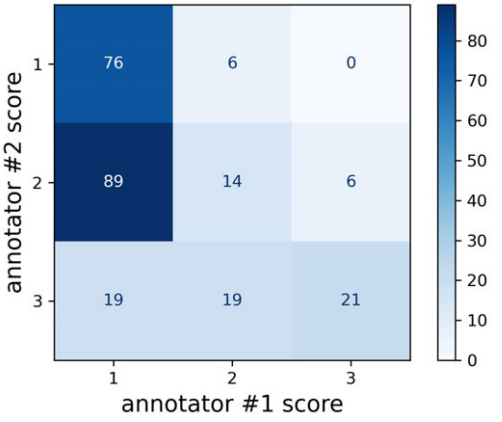

Figure 5. Confusion matrix of the catchiness scores assigned by the two annotators.

7.5 Human evaluation

Based on the evaluation we conducted in previous sections, we include all the methods we introduced in Sections 5 and 6 in our final model, namely, company name delexicalisation, entity masking and conditional training based on the POS tag of the first slogan token. We incorporate an additional control code 'ENT' to cover the cases where a reference slogan starts with an entity mask token. Based on the result in Section 7.4, we randomly sample a control code from the set {JJ, VB, DT, PR, OTHER} during inference time. Finally, we replace the entity mask tokens in the slogan (if any) using the reverse dictionary induced from the input description to produce the final slogan as described in Section 5.2.

We randomly sampled 50 companies from the test set (different from the sample in Section 7.4) and obtained the predicted slogans from our model, along with four other baselines: first sentence, skeleton-based, Pointer-Generator and DistilBART. Therefore, we have in total 250 slogans to evaluate. We invited two human annotators to score the slogans independently based on three fine-grained aspects: coherence, well-formedness and catchiness. They assign scores on a scale of 1–3 (poor, acceptable, good) for each aspect.

We display the input description along with the slogan so that the annotators can assess whether the slogan is coherent with the description. We also randomise the slogans' order to remove positional bias. The annotation guideline is shown in Appendix B and the annotation UI is presented in Appendix C.

We measure the inter-annotator agreement using Cohen's kappa coefficient (Cohen 1960). The κ value for coherence, well-formedness and catchiness are 0.493 (moderate), 0.595 (moderate) and 0.164 (slight) separately. The 'catchiness' aspect has a low κ value because it is much more subjective. While the annotators generally agree on an unattractive slogan, their standards for catchiness tend to differ. It can be illustrated in Figure 5 where the agreement is high when the assigned score is 1 (poor). However, there are many examples where annotator 1 assigned score 1 (poor) and annotator 2 assigned score 2 (acceptable). There are only 19 slogans (7.6%) where the annotators assigned opposite labels. Therefore, we believe the low agreement is mainly due to individual differences rather than annotation noise.

We average the scores assigned by the two annotators and present the result in Table 16.

The first sentence baseline received low well-formedness and catchiness scores. As we mentioned earlier, the first sentence of the description is often much longer than a typical slogan, failing to satisfy the conciseness property of slogans. Lucas (1934) observed that longer slogans are also less memorable and attractive, which is validated by the low catchiness score.

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

Table 16. Human evaluation on three aspects: coherence, well-formedness and catchiness. We average the scores assigned by the two annotators. The best score for each aspect is highlighted in bold (we exclude the first sentence baseline for the 'coherent' aspect because it is 'coherent' by definition). ** indicates statistical significance using a double-sided paired t-test with p-value=0.005 comparing with our proposed method

System	Coherent	Well-formed	Catchy
First sentence	3.00**	1.49**	1.19**
Skeleton-based	2.31**	1.36**	1.41**
Pointer-Generator	2.63**	2.07**	1.51**
DistilBART	2.89	**2.81**	1.87**
Ours	**2.91**	2.79	**2.22**

The skeleton-based approach improved the catchiness over the first sentence baseline by a reasonable margin. However, it received the lowest well-formedness score due to the limitations of skeletons, causing it to generate nongrammatical or nonsensical slogans occasionally. Moreover, it has a much lower coherence score than either GRU or Transformer seq2seq models, which is our primary motivation to apply the seq2seq framework instead of relying on random slogan skeletons.

The Pointer-Generator baseline outperformed the previous two baselines across all aspects. On the one hand, it demonstrates the capability of modern deep learning models. On the other hand, it surfaces the limitations of word overlap-based evaluation metrics. Based on the ROUGE scores reported in Section 7.1 alone, we could not conclude the superiority of the Pointer-Generator model over the first sentence or first-k words baseline.

The DistilBART model improved further over the Pointer-Generator baseline, especially in the well-formedness aspect. It is likely due to the extensive pretraining and its ability to generate grammatical and realistic text.

Our proposed method received similar coherence and well-formedness scores as DistilBART. However, it outperformed all other methods in catchiness by a large margin. Although the improvement of coherence is not statistically significant, it does not necessarily mean the delexicalisation and entity masking techniques are not helpful. As we discussed in Section 7.4, our method generates substantially more diverse slogans, and the generation is much more abstractive than the DistilBART baseline. Previous work highlighted the trade-off between abstractiveness and truthfulness (Durmus *et al*. 2020). By combining the approaches to improve truthfulness and diversity, our proposed method generates more catchy and diverse slogans without sacrificing truthfulness or well-formedness.

8. Ethical considerations

All three annotators employed in this study are full-time researchers at Knorex. We explained to them the purpose of this study and obtained their consent. They conducted the annotation during working hours and are paid their regular wage.

Marketing automation is a strong trend in the digital advertising industry. AI-based copywriting is a challenging and crucial component in this process. We take generating counterfactual advertising messages seriously as it might damage the advertiser's brand image and harm the prospects. The model proposed in this work generates better quality and more truthful slogans than various baselines. However, we cannot yet conclude that the generated slogans are 100% truthful, just like most recently proposed language generation models. This work is being

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

integrated into a commercial digital advertising platform[v]. In the initial version, advertisers are required to review and approve the slogans generated by the system. They can also make modifications as necessary before the ads go live.

9. Conclusion

In this work, we model slogan generation using a seq2seq Transformer model with the company's description as input. It ensures coherence between the generated slogan and the company's marketing communication. In addition, we applied company name delexicalisation and entity masking to improve the generated slogans' truthfulness. We also introduced a simple conditional training method to generate more diverse slogans. Our model achieved a ROUGE -1/-2/-L F_1 score of 35.58/18.47/33.32 on a manually curated slogan dataset. Comprehensive evaluations demonstrated that our proposed method generates more truthful and diverse slogans. A human evaluation further validated that the slogans generated by our system are significantly catchier than various baselines.

As ongoing work, we are exploring other controllable aspects, such as the style (Jin *et al*. 2020) and the sentence parse (Sun *et al*. 2021). Besides, we are also working on extending our method to generating longer texts (Hua *et al*. 2021) which can be used as the body text in advertising.

Acknowledgement. Yiping is supported by the scholarship from 'The 100th Anniversary Chulalongkorn University Fund for Doctoral Scholarship' and also 'The 90th Anniversary Chulalongkorn University Fund (Ratchadaphiseksomphot Endowment Fund)'. We would like to thank our colleagues Vishakha Kadam, Khang Nguyen and Hy Dang for conducting the manual evaluation on the slogans. We would like to thank the anonymous reviewers for their careful reading of the manuscript and constructive criticism.

References

Abrams Z. and **Vee E.** (2007). Personalized ad delivery when ads fatigue: An approximation algorithm. In *Proceedings of the International Workshop on Web and Internet Economics*, Bangalore, India. Springer, pp. 535–540.

Ackley D.H., **Hinton G.E.** and **Sejnowski T. J.** (1985). A learning algorithm for boltzmann machines. *Cognitive Science* **9**(1), 147–169.

Alnajjar K. and **Toivonen H.** (2021). Computational generation of slogans. *Natural Language Engineering* **27**(5), 575–607.

Angeli G., **Premkumar M.J.J**. and **Manning C.D.** (2015). Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Beijing, China. Association for Computational Linguistics, pp. 344–354.

Bahdanau D., **Cho K.** and **Bengio Y.** (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA, USA.

Boigne J. (2020). Building a slogan generator with gpt-2. Available at https://jonathanbgn.com/gpt2/2020/01/20/slogan-generator.html (accessed 14 January 2020).

Bruce N.I., **Murthi B.** and **Rao R.C.** (2017). A dynamic model for digital advertising: The effects of creative format, message content, and targeting on engagement. *Journal of Marketing Research* **54**(2), 202–218.

Caccia M., **Caccia L.**, **Fedus W.**, **Larochelle H.**, **Pineau J.** and **Charlin L.** (2019). Language gans falling short. In *Proceedings of the International Conference on Learning Representations*, New Orleans, Louisiana.

Cao M., **Dong Y.**, **Wu J.** and **Cheung J.C.K.** (2020). Factual error correction for abstractive summarization models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pp. 6251–6258.

Cao Z., **Wei F.**, **Li W.** and **Li S.** (2018). Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, New Orleans, Louisiana.

Chen S., **Zhang F.**, **Sone K.** and **Roth D.** (2021). Improving faithfulness in abstractive summarization with contrast candidate generation and selection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online. Association for Computational Linguistics, pp. 5935–5941.

[v] knorex.com

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

Cho K., **van Merriënboer B.**, **Gulcehre C.**, **Bahdanau D.**, **Bougares F.**, **Schwenk H.** and **Bengio Y.** (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, Doha, Qatar. Association for Computational Linguistics, pp. 1724–1734.

Cohen J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* **20**(1), 37–46.

Devlin J., **Chang M.-W.**, **Lee K.** and **Toutanova K.** (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume* 1 *(Long and Short Papers)*, Minneapolis, Minnesota. Association for Computational Linguistics, pp. 4171–4186.

Dong Y., **Wang S.**, **Gan Z.**, **Cheng Y.**, **Cheung J.C.K.** and **Liu J.** (2020). Multi-fact correction in abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pp. 9320–9331.

Durmus E., **He H.** and **Diab M.** (2020). Feqa: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 5055–5070.

Eyal M., **Baumel T.** and **Elhadad M.** (2019). Question answering as an automatic evaluation metric for news article summarization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume* 1 *(Long and Short Papers)*, Minneapolis, Minnesota. Association for Computational Linguistics, pp. 3938–3948.

Falke T., **Ribeiro L.F.**, **Utama P.A.**, **Dagan I.** and **Gurevych I.** (2019). Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics, pp. 2214–2220.

Fan A., **Lewis M.** and **Dauphin Y.** (2018). Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia. Association for Computational Linguistics, pp. 889–898.

Gabriel S., **Celikyilmaz A.**, **Jha R.**, **Choi Y.** and **Gao J.** (2021). GO FIGURE: A meta evaluation of factuality in summarization. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online. Association for Computational Linguistics, pp. 478–487.

Gao X., **Lee S.**, **Zhang Y.**, **Brockett C.**, **Galley M.**, **Gao J.** and **Dolan W.B.** (2019). Jointly optimizing diversity and relevance in neural response generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, USA. Association for Computational Linguistics, pp. 1229–1238.

Gatti L., **Özbal G.**, **Guerini M.**, **Stock O.** and **Strapparava C.** (2015). Slogans are not forever: Adapting linguistic expressions to the news. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, Buenos Aires, Argentina, pp. 2452–2458.

Gatti L., **Özbal G.**, **Stock O.** and **Strapparava C.** (2017). To sing like a mockingbird. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, Valencia, Spain. Association for Computational Linguistics, pp. 298–304.

Goodrich B., **Rao V.**, **Liu P.J.** and **Saleh**, **M.** (2019). Assessing the factual accuracy of generated text. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, Alaska. Association for Computing Machinery, pp. 166–175.

He T., **Zhang Z.**, **Zhang H.**, **Zhang Z.**, **Xie J.** and **Li M.** (2019). Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA. Institute of Electrical and Electronics Engineers, pp. 558–567.

Hermann K.M., **Kocisky T.**, **Grefenstette E.**, **Espeholt L.**, **Kay W.**, **Suleyman M.** and **Blunsom P.** (2015). Teaching machines to read and comprehend. *In Advances in Neural Information Processing Systems*, Montreal, Canada, pp. 1693–1701.

Hochreiter S. and **Schmidhuber J.** (1997). Long short-term memory. *Neural Computation* **9**(8), 1735–1780.

Holtzman A., **Buys J.**, **Du L.**, **Forbes M.** and **Choi Y.** (2019). The curious case of neural text degeneration. In *Proceedings of the International Conference on Learning Representations*, New Orleans, Louisiana.

Howard J. and **Gugger S.** (2020). Fastai: A layered API for deep learning. *Information* **11**(2), 108.

Hua X., **Sreevatsa A.** and **Wang L.** (2021). DYPLOC: Dynamic planning of content using mixed language models for text generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume* 1: *Long Papers)*, Online. Association for Computational Linguistics, pp. 6408–6423.

Hughes J.W., **Chang K.-h.** and **Zhang R.** (2019). Generating better search engine text advertisements with deep reinforcement learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, Alaska. Association for Computing Machinery, pp. 2269–2277.

Iwama K. and **Kano Y.** (2018). Japanese advertising slogan generator using case frame and word vector. In *Proceedings of the 11th International Conference on Natural Language Generation*, Tilburg, The Netherlands. Association for Computational Linguistics, pp. 197–198.

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

Jin D., **Jin Z.**, **Zhou J.T.**, **Orii L.** and **Szolovits P.** (2020). Hooks in the headline: Learning to generate headlines with controlled styles. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 5082–5093.

Kanungo Y.S., **Negi S.** and **Rajan A.** (2021). Ad headline generation using self-critical masked language model. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers*. Association for Computational Linguistics, pp. 263–271.

Katragadda R., **Pingali P.** and **Varma V.** (2009). Sentence position revisited: A robust light-weight update summarization 'baseline' algorithm. In *Proceedings of the Third International Workshop on Cross Lingual Information Access: Addressing the Information Need of Multilingual Societies (CLIAWS3)*, Boulder, Colorado. Association for Computational Linguistics, pp. 46–52.

Keskar N.S., **McCann B.**, **Varshney L.**, **Xiong C.** and **Socher R.** (2019). CTRL - A conditional transformer language model for controllable generation. arXiv preprint arXiv:1909.05858.

Kryscinski W., **McCann B.**, **Xiong C.** and **Socher R.** (2020). Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pp. 9332–9346.

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*, Online. Association for Computational Linguistics, pp. 7871–7880.

Li J., **Monroe W.** and **Jurafsky D.** (2016). A simple, fast diverse decoding algorithm for neural generation. arXiv preprint arXiv:1611.08562.

Liu Y., **Ott M.**, **Goyal N.**, **Du J.**, **Joshi M.**, **Chen D.**, **Levy O.**, **Lewis M.**, **Zettlemoyer L.** and **Stoyanov V.** (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Lucas D.B. (1934). The optimum length of advertising headline. *Journal of Applied Psychology* **18**(5), 665.

Luong M.-T., **Pham H.** and **Manning C. D.** (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal. Association for Computational Linguistics, pp. 1412–1421.

Matsumaru K., **Takase S.** and **Okazaki N.** (2020). Improving truthfulness of headline generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 1335–1346.

Maynez J., **Narayan S.**, **Bohnet B.** and **McDonald R.** (2020). On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 1906–1919.

Mieder B. and **Mieder W.** (1977). Tradition and innovation: Proverbs in advertising. *Journal of Popular Culture* **11**(2), 308.

Mikolov T., **Sutskever I.**, **Chen K.**, **Corrado G.S.** and **Dean J.** (2013). Distributed representations of words and phrases and their compositionality. *In Advances in neural information processing systems*, volume **26**, Lake Tahoe, Nevada, USA, pp. 3111–3119.

Misawa S., **Miura Y.**, **Taniguchi T.** and **Ohkuma T.** (2020). Distinctive slogan generation with reconstruction. In *Proceedings of Workshop on Natural Language Processing in E-Commerce*, Barcelona, Spain. Association for Computational Linguistics, pp. 87–97.

Mishra S., **Verma M.**, **Zhou Y.**, **Thadani K.** and **Wang W.** (2020). Learning to create better ads: Generation and ranking approaches for ad creative refinement. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. Association for Computing Machinery, pp. 2653–2660.

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 The Web Conference 2018*, Lyon, France. Association for Computing Machinery, pp. 335–342.

Nan F., **Nallapati R.**, **Wang Z.**, **Nogueira dos Santos C.**, **Zhu H.**, **Zhang D.**, **McKeown K.** and **Xiang B.** (2021). Entity-level factual consistency of abstractive text summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, pp. 2727–2733.

Niu X., **Xu W.** and **Carpuat M.** (2019). Bi-directional differentiable input reconstruction for low-resource neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, USA. Association for Computational Linguistics, pp. 442–448.

Özbal G., **Pighin 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)*, Sofia, Bulgaria. Association for Computational Linguistics, pp. 1446–1455.

Pagnoni A., **Balachandran V.** and **Tsvetkov Y.** (2021). Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pp. 4812–4829.

Phillips B.J. and **McQuarrie E.F.** (2009). Impact of advertising metaphor on consumer belief: Delineating the contribution of comparison versus deviation factors. *Journal of Advertising* **38**(1), 49–62.

Qi P., **Zhang Y.**, **Zhang Y.**, **Bolton J.** and **Manning C.D.** (2020). Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Online. Association for Computational Linguistics, pp. 101–108.

Radford A., **Wu J.**, **Child R.**, **Luan D.**, **Amodei D.** and **Sutskever I.** (2019). Language models are unsupervised multitask learners. *OpenAI blog* **1**(8), 9.

Reddy R. (1977). *Speech understanding systems: A summary of results of the five-year research effort*. Carnegie Mellon University.

Rogers A., **Kovaleva O.** and **Rumshisky A.** (2020). A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics* **8**, 842–866.

Scialom T., **Lamprier S.**, **Piwowarski B.** and **Staiano J.** (2019). Answers unite! unsupervised metrics for reinforced summarization models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China. Association for Computational Linguistics, pp. 3237–3247.

See A., **Liu P.J.** and **Manning C.D.** (2017). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume* 1: *Long Papers)*, Vancouver, Canada. Association for Computational Linguistics, pp. 1073–1083.

Sun J., **Ma X.** and **Peng N.** (2021). AESOP: Paraphrase generation with adaptive syntactic control. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics, pp. 5176–5189.

Sutskever I., **Vinyals O.** and **Le Q.V.** (2014). Sequence to sequence learning with neural networks. *In Advances in Neural Information Processing Systems*, volume **27**, Montreal, Quebec, Canada, pp. 3104–3112.

Tenney I., **Das D.** and **Pavlick E.** (2019). Bert rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics, pp. 4593–4601.

Tomašic P., **Znidaršic M.** and **Papa G.** (2014). Implementation of a slogan generator. In *Proceedings of 5th International Conference on Computational Creativity*, volume 301, Ljubljana, Slovenia, pp. 340–343.

Vaswani A., **Shazeer N.**, **Parmar N.**, **Uszkoreit J.**, **Jones L.**, **Gomez A.N.**, **Kaiser Ł.** and **Polosukhin I.** (2017). Attention is all you need. *In Advances in neural information processing systems*, pp. 5998–6008, Long Beach, CA, USA.

Vempati S., **Malayil K.T.**, **Sruthi V.** and **Sandeep R.** (2020). Enabling hyper-personalisation: Automated ad creative generation and ranking for fashion e-commerce. In *Fashion Recommender Systems*. Springer, pp. 25–48.

Wang A., **Cho K.** and **Lewis M.** (2020). Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 5008–5020.

Welleck S., **Kulikov I.**, **Roller S.**, **Dinan E.**, **Cho K.** and **Weston J.** (2019). Neural text generation with unlikelihood training. In *Proceedings of the International Conference on Learning Representations*, New Orleans, Louisiana.

White G.E. (1972). Creativity: The X factor in advertising theory. *Journal of Advertising* **1**(1), 28–32.

Williams A., **Nangia N.** and **Bowman S.** (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Wolf T., **Debut L.**, **Sanh V.**, **Chaumond J.**, **Delangue C.**, **Moi A.**, **Cistac P.**, **Rault T.**, **Louf R.**, **Funtowicz M.**, **Davison J.**, **Shleifer S.**, **von Platen P.**, **Ma C.**, **Jernite Y.**, **Plu J.**, **Xu C.**, **Le Scao T.**, **Gugger S.**, **Drame M.**, **Lhoest Q.** and **Rush A.** (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online. Association for Computational Linguistics, pp. 38–45.

Zhang H., **Duckworth D.**, **Ippolito D.** and **Neelakantan A.** (2021). Trading off diversity and quality in natural language generation. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, Online. Association for Computational Linguistics, pp. 25–33.

Zhang J., **Zhao Y.**, **Saleh M.** and **Liu P.** (2020a). Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the International Conference on Machine Learning*. PMLR, pp. 11328–11339.

Zhang Y., **Merck D.**, **Tsai E.**, **Manning C.D.** and **Langlotz C.** (2020b). Optimizing the factual correctness of a summary: A study of summarizing radiology reports. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 5108–5120.

Zhu C., **Hinthorn W.**, **Xu R.**, **Zeng Q.**, **Zeng M.**, **Huang X.** and **Jiang M.** (2021). Enhancing factual consistency of abstractive summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online. Association for Computational Linguistics, pp. 718–733.

Appendix A. Details of data cleaning

We perform the following steps in sequence to obtain clean (description, slogan) pairs.

(1) Delexicalise the company name in both the description and the HTML page title.
(2) Remove all non-alphanumeric characters at the beginning and the end of the HTML page title.
(3) Filter by blocked keywords/phrases. Sometimes, the crawling is blocked by a firewall, and the returned title is 'Page could not be loaded' or 'Access to this page is denied'. We did a manual analysis of a large set of HTML page titles and came up with a list of 50 such blocked keywords/phrases.
(4) Remove prefix or suffix phrases indicating the structure in the website, such as 'Homepage - ', '| Welcome page', 'About us'.
(5) Split the HTML page title with special characters[w]. Select the longest chunk as the candidate slogan that either does not contain the company name or has the company name at the beginning (in which case we will strip off the company name and not affect the fluency).
(6) Deduplicate the slogans and keep only the first occurring company if multiple companies have the same slogan.
(7) Filter based on the length of the description and the slogan. The slogan must contain between 3 and 12 words, while the description must contain at least 10 words.
(8) Concatenate the description and the slogan and detect their language using an open-source library[x]. We keep the data only if its detected language is English.
(9) Filter based on lexicographical features, such as the total punctuations in the slogan must not exceed three, the longest word sequence without any punctuation must be at least three words. We come up with these rules based on an analysis of a large number of candidate slogans.
(10) Filter based on named entity tags. We use Stanza (Qi *et al.* 2020) to perform named entity recognition on the candidate slogans. Many candidates contain a long list of locations names. We discard a candidate if over 30% of its surface text consists of named entities with the tag. `'GPE'`.

Table 17 provides examples of the cleaning/filtering process.

Appendix B. Slogan annotation guideline for human evaluators

You will be shown five generated slogans for the same company in sequence at each time. They were generated using different models and rules. For each slogan, please rate on a scale of 1–3 (poor, acceptable and good) for each of the three aspects (coherent, well-formed and catchy). Please ensure you rate all the aspects before moving on to the next slogan. Please also ensure your rating standard is consistent both among the candidate slogans for the same company and across different companies.

Please note that the order of the slogans is randomly shuffled. So you should not use the order information to make a judgement.

The details and instructions are as follows:

[w] We use one or more consecutive characters in the set {|, <, >, −, /}.
[x] https://github.com/shuyo/language-detection

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

Table 17. Sample descriptions and slogans before and after the data cleaning. Note that '-' indicates the algorithm fails to extract a qualified slogan and the example will be removed

Company: Knorex (https://www.knorex.com/)
Desc: Cross-channel marketing cloud platform augmented by Machine Learning. Single dashboard to automate and optimize all your campaigns across digital marketing funnels in one place.
HTML Title: Cross-Channel Marketing Automation Platform | Universal Marketing | More Than Just A DSP | Knorex.com
Extracted Slogan: Cross-Channel Marketing Automation Platform
Explaination: Longest chunk after splitting; not filtered by any of the rules.
Company: GoPro (https://gopro.com/en/us/)
Desc: Discover the world's most versatile action cameras + accessories. Possibilities are endless with waterproof, live streaming, stabilizing features + more.
HTML Title: GoPro | World's Most Versatile Cameras | Shop Now & Save
Extracted Slogan: World's Most Versatile Cameras
Explaination: Longest chunk after splitting; not filtered by any of the rules.
Company: Adfuel Media Inc. (https://goadfuel.com/)
Desc: Adfuel is North America's Digital Marketing agency providing Digital Marketing Services, Programmatic Advertising, Geo-Fencing, and Campaign Budgeting by using a universal advertising platform.
HTML Title: Digital Marketing Agency in Miami | Digital Marketing Agency in Ontario
Extracted Slogan: Digital Marketing Agency in Ontario
Explaination: Longest chunk after splitting; contains one named entity not exceeding 30% of the surface text.
Company: BMW (https://www.bmw.com/en/index.html)
Desc: Dive into new worlds with BMW, get inspired, and experience the unknown, the unusual and some useful things, too.
HTML Title: BMW.com | The international BMW Website
Extracted Slogan: -
Explaination: The company name appears in the middle of the candidate slogan 'The international BMW Website' (Rule 5).

Coherent

A slogan should be coherent with the company description. There are two criteria that it needs to satisfy to be coherent. Firstly, it needs to be *relevant* to the company. For example, the following slogan is incoherent because there is no apparent link between the description and the generated slogan.

Slogan: The best company rated by customers
Description: Knorex is a provider of performance precision marketing solutions

Secondly, the slogan should not introduce *unsupported* information. Namely, if the description does not mention the company's location, the slogan should not include a location. However, there are cases where the location can be inferred, although the exact location does not appear in the description. We provide some hypothetical examples and the ratings you should provide.

Description: Knorex is a provider of performance precision marketing solutions based in **California**.
Slogan 1: **US**-based Digital Marketing Company (3, because California infers the company is in the US).

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

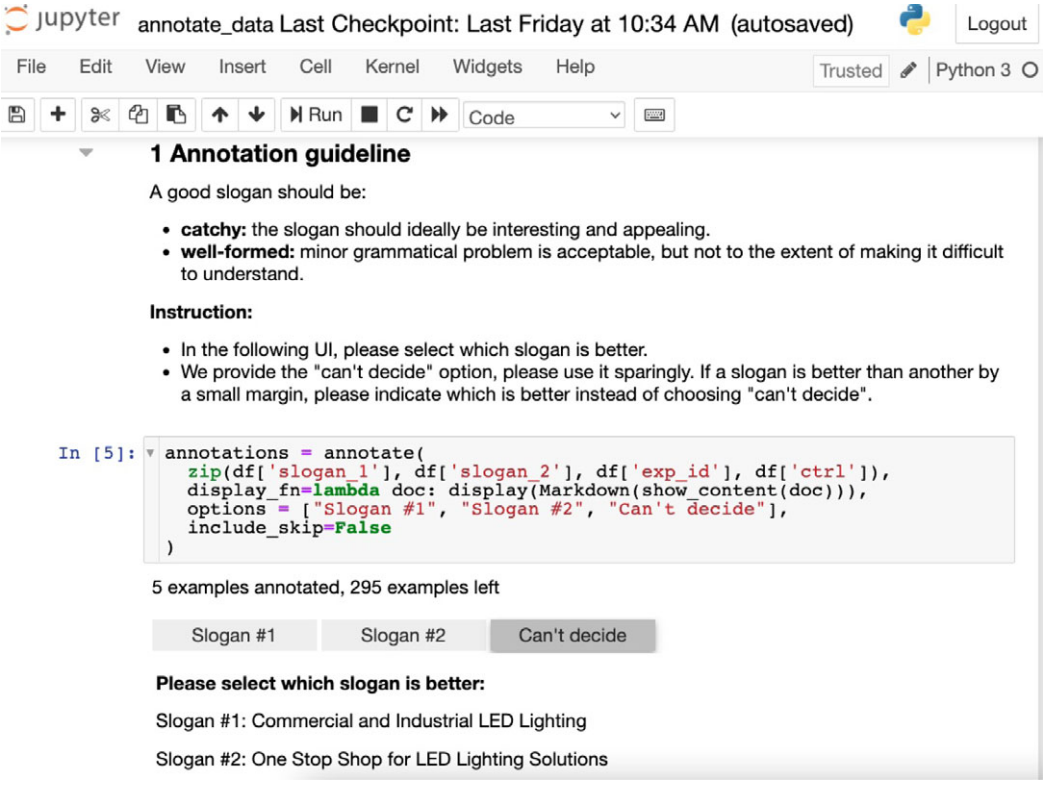

Figure 6. User interface for pair-wise slogan ranking described in Section 7.4. One of the candidate slogans uses our proposed syntactic control code, while another candidate uses nucleus sampling. We randomise the order of the slogans to eliminate positional bias.

Slogan 2: Digital Marketing Company (3, the slogan does not have to cover all the information in the description).

Slogan 3: Digital Marketing Company in **Palo Alto** (2, it may be true but we can't verify based on the description alone).

Slogan 4: Digital Marketing Company in **China** (1, it is false).

Please focus on verifying *factual* information (location, number, year, etc.) instead of *subjective* description. Expressions like 'Best . . .' or 'Highest-rated . . .' usually do not affect the coherence negatively.

Well-formed

A slogan should be well-formed, with appropriate *specificity* and *length*. It should also be *grammatical* and *make sense*. Examples that will receive poor scores in this aspect:

- Paragraph-like slogans (because it's not concise and inappropriate to form a slogan).
- Very short slogans that are unclear what message they convey (e.g., 'Electric Vehicle').
- Slogans containing severe grammatical errors or do not make sense (e.g., slogans that look like a semi-random bag of words).

https://doi.org/10.1017/S1351324921000474 Published online by Cambridge University Press

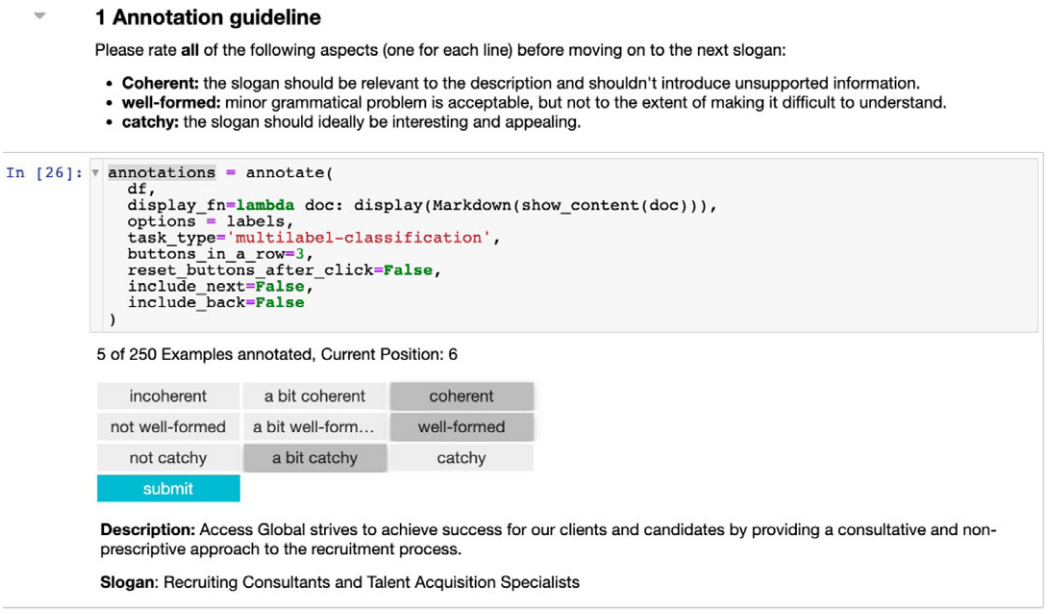

Figure 7. User interface for fine-grained slogan evaluation described in Section 7.5. We randomise the order of the slogans to eliminate positional bias.

Catchy

A slogan should be catchy and memorable. Examples are using metaphor, humour, creativity, or other rhetorical devices. So slogan A below is better than slogan B (for the company M&Ms).

Slogan A: Melts in Your Mouth, Not in Your Hands
Slogan B: Multi-Coloured Button-Shaped Chocolates

Lastly, please perform the labelling independently, especially do not discuss with the other annotator performing the same task. Thank you for your contribution!

Appendix C. Annotation interface

We implement the human annotation interfaces using Jupyter notebook widgets implemented in Pigeon[y]. Figure 6 shows the UI for pair-wise ranking task conducted in Section 7.4 and Figure 7 shows the UI for fine-grained evaluation conducted in Section 7.5.

[y] https://github.com/agermanidis/pigeon

Cite this article: Jin Y, Bhatia A, Wanvarie D and Le PTV (2023). Towards improving coherence and diversity of slogan generation. *Natural Language Engineering* **29**, 254–286. https://doi.org/10.1017/S1351324921000474