ARIEL UNIVERSITY

MASTER THESIS PROPOSAL

Abstractive Narrative Generation

Author:
Yuri SAFOVICH

Supervisor:
Dr. Amos AZARIA

A thesis proposal submitted in partial fulfillment of the requirements for the degree of Master

in the

Department of Computer Science

March 10, 2019

Declaration of Authorship

I, Yuri SAFOVICH, hereby declare that this thesis proposal entitled, "Abstractive Narrative Generation" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

"Substitute 'damn' every time you're inclined to write 'very'; your editor will delete it and the writing will be just as it should be."

Mark Twain

"Easy reading is damn hard writing."

Nathaniel Hawthorne

Ariel University

Abstract

Faculty of Natural Sciences
Department of Computer Science

Master

Abstractive Narrative Generation

by Yuri SAFOVICH

In the field of interactive storytelling, AI planning was first proposed for tasks of narrative generation and is the dominant approach. Abstractive generation of text was suggested and shown with advances in deep neural networks. We describe the task of sentence expansion and enhancement, in which a sentence provided by a human is expanded in some creative way. Sentence expansion and enhancement may serve as an authoring tool, or by dynamic media, conversational agents, advertising, or as a pipeline component. The expansion or enhancement should be understandable, believably grammatical, and optionally meaning-preserving. We implement a neural sentence expander, with optional style priming, trained on sentence compressions generated from a corpus of modern fiction. We use a modified MLE objective, and decode at test time with controlled novelty sampling. We run our sentence expander on sentences provided by human subjects and have humans evaluate these expansions. We show that while the generation methods are inferior in percentage to professional human writers, they are comparable to original human input sentences, and preferred over baselines.

Acknowledgements

I wish to thank my thesis advisor, Dr. Amos Azaria, for his guidance and foresight, his tremendous patience, and for listening to my ideas and showing me why they don't work. I wish to thank Ariel University for the personal scholarship that financially supported me during the research. Thanks to the administrative staff of the Department of Computer Science, and the faculty, for teaching me, despite my objections.

Contents

Declaration of Authorship **iii**

Abstract **vii**

Acknowledgements **ix**

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Approach . . . 2

2 Background **5**
2.1 Related Work . . . 7
2.1.1 Sentence Compression . . . 7
2.1.2 Sentence Generation . . . 7

3 Methods **9**
3.1 Methods . . . 9
3.1.1 Modified Objective . . . 9
3.1.2 Controlled Sampling . . . 9

4 Experiments **13**
4.1 Experiments . . . 13
4.1.1 Corpus . . . 13
4.1.2 Optimization . . . 13
4.1.3 Setup . . . 14
Compressor . . . 14
Data . . . 14
Training . . . 14
Test . . . 14
4.1.4 Results . . . 15
Metrics . . . 17
Discussion . . . 18

5 Conclusion **19**
5.1 Conclusion and Future Work . . . 19

Bibliography **21**

List of Figures

3.1 Decoding the human input “the tree came alive and started talking" to get “then shortly off my left , the huge bare beige christmas tree came alive and started talking again". Zoom in for outputs. This uses a truncated parabola model modified by a size-3 window accumulator (value in orange). Corrected temperature τ is in green. 11

List of Tables

1.1 Example sentence expansion and enhancement outputs for human inputs. 2
1.2 Example kernels and original sentences. 3

4.1 Sampling method evaluations. 16
4.2 Example expansions for human input, with approval ratios. 16
4.3 Example kernels and original sentences, with approval ratios. 17
4.4 Beam search (width 10) with manually evaluated entailment and InferSent distances. 17
4.5 Example outputs, general and primed style. 18

List of Abbreviations

Char-RNN	Character-based RNN
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
ILP	Integer Linear Programming
LM	Language Model
LSTM	Long Short-Term Memory (RNN)
m	million
NLP	Natural Language Processing
NN	Neural Network
PoS	Part-of-Speech
PMI	Pointwise Mutual Information
NER	Named Entity Recognition
RNN	Recurrent Neural Network
RNNLM	Recurrent Neural Network Language Model
seq2seq	sequence-to-sequence RNN architecture
SVM	Support Vector Machine

Chapter 1

Introduction

1.1 Motivation

Computational creativity has always been a popular idea; automatic storytelling and poetry have been attempted from early in computing. Narratology, the study of storytelling, divides it into story (plot) and discourse (style, chronology of presentation), also known as fabula and syuzhet (Propp, 1928). Most research has been focused on the former (Li et al., 2014). For example, a set of elements and actions is given, with preconditions and postconditions, and then the construction of a story plot is a search problem. The use of deep neural networks permits generating the language in an integrative way with respect to plot elements. Without some direction, however, the outputs lack innate meaning. Consider the following example of fully abstractive generation via character-based language model recurrent neural networks (Char-RNN) (Sutskever, Martens, and Hinton, 2011), a method with some outside media publicity:

> while he was giving attention to the second advantage of school building a 2-for-2 stool killed by the Cultures saddled with a half- suit defending the Bharatiya Fernall 's office . Ms . Claire Parters will also have a history temple for him to raise jobs until naked Prodiena to paint baseball partners ,

Such artifacts as the above are human-readable metrics, a language modeling result used for comparative analysis in neural network research. To have better grounding for a generative model, we draw on human participation for input. This enables more nuanced output than in pure generation, and identifies an applied goal.

In the sentence expansion problem, an agent, model, or system is given an input of a (human composed) sentence and must output a longer sentence that would be preferred over the original sentence by a human judge (while mostly preserving the content of the original sentence). For example, we may better appreciate if, given an input sentence such as "hello world", a generative model in a certain mood would produce for us "Oh , hello , a world of peace ." (This is an actual model output). An ideal model would not only consistently pass a Turing test, but also be "more human" in analogical terms. It could then be applied as a non-strictly preserving writing aid, where sentence expansion would convert a "summary" tell into show. It can also be used as component in a conversational agent, adaptive media, and so on.

A practical example for the usefulness of computational creativity is the automatic generation of dynamic content, including text, in computer games. Static, human-authored game dialogue systems can be likened to curated interaction with

conversational agents, and are an ideal target for a neural-assisted creative writing interface, such as we propose. These dialogue systems also do not require advance plot integration planning. In the literature, most narrative generation methods have been extractive, meaning chosen words or connections are present in some source schema, and logic-, graph- or template-based (Li et al., 2013; Young and Moore, 1994; Riedl and Young, 2010; Chambers and Jurafsky, 2008; Chambers and Jurafsky, 2011). The topic of sentence enhancement would include slightly modifying words and concepts. In the context of deep learning, abstractive generation is easier, and may be interesting also for the potential relationship with general creativity in AI.

1.2 Approach

With the recent advances in hardware and neural networks, interest has grown in abstractive text generation models, due to capacity for generalization, long-range interactions, and to reduce manual knowledge modeling. A common problem with the essentially statistical systems is generation of safe sequences that are relevant for many inputs (Li et al., 2016). Fully abstractive generation may result in good-looking but meaningless or irrelevant text (Sutskever, Martens, and Hinton, 2011; Bowman et al., 2016). Some authors have mixed neural components in extractive work, and newer models, including generative adversarial networks (GANs) and variational autoencoders, learn to generate text with diversity through additional parameters in generation or training. However success relative to standard neural language models is debatable (Semeniuta, Severyn, and Gelly, 2018; Cífka et al., 2018) for tasks applicable to both (not including e.g. autoencoder reconstruction). At inference, on the decoder side of encoder-decoder seq2seq (Sutskever, Vinyals, and Le, 2014), beam search tends to produce more generic additions while random sampling is more unreliable, in particular for the task we propose.

TABLE 1.1: Example sentence expansion and enhancement outputs for human inputs.

Input	the woman glared at the child .
Output	the old woman glared down at the younger child .
Input	the robot looked at jake and smirked , it seemed .
Output	the little gray crew looked out at jake and scowled, it seemed like a greatly bad-tempered one.

In this work, we expand on human input, transforming complete but possibly unembellished sentences to give them a general or specific style (see Table 1.1). Given an input (*the robot looked at jake...*), RNN models add content or context, and style as form or decoration. In terms of the narratological split to story and discourse the inputs are story clauses. In the example, "robot" expands to "little gray crew[man]" demonstrating abstractive generation.

The RNN seq2seq with attention (Bahdanau, Cho, and Bengio, 2014) models are trained to transform sentence "kernels" to their source form. These sentence kernels are obtained by using sentence compression techniques on a corpus of modern (mid to late 20th century) fiction, which is scraped from online resources. (The corpus is further described in Section 4.1.1.)

See Table 1.2 for examples of compression kernels. Using the seq2seq platform for our expansion task, we first modify MLE loss to emphasize learning new words,

TABLE 1.2: Example kernels and original sentences.

Kernel	smoke belched from the pipe .
Original	blue smoke belched from the chromed exhaust pipe .
Kernel	are you back ?
Original	are you back with the revolutionary lover ?

making extended training possible. We then investigate simple alternative test-time sampling methods to better control randomness in decoding. These changes improve output quality, in terms of the average preference of human judges.

We run our sentence expander system on crowd-sourced input. We show that our best method of sentence expansion results in sentences that are preferred by human users as often as the crowd-sourced input; that is 20% more of total expansions than with other, baseline system output sentences.

To summarize, the main contributions of this work are three-fold. First, we define the problem of sentence expansion and enhancement and explain its importance. Second, we present a method that allows the generation of a large dataset for the sentence expansion problem, by using sentence compression techniques on a given corpus. Third, we present a method for automatic sentence expansion, which is based on several novel ideas, and show that humans prefer sentences produced by our system to original input sentences significantly more often than with baseline systems.

We describe our experimental results and give human-rated examples in Section 4.1.4.

Chapter 2

Background

A general overview of deep learning is (Schmidhuber, 2015). Very briefly, a DNN is an NN with more layers than the previously usual few. An RNN is the "deepest" NN because the neurons are interconnected, vaguely resembling a brain, and can "pulse" indefinitely. RNNs are not necessarily best for natural language processing (NLP), as (Dauphin et al., 2017) show, but are interesting for their ability to learn and map relationships between arbitrary sequences. However, training gradients then quickly vanish. One robust, albeit comparatively complex, descendant of RNN is Long Short-Term Memory (Hochreiter and Schmidhuber, 1997)), meant to learn longer sequences effectively.

Two important developments facilitate the use of sentences as input instead of trivially vectorized characters (or images). First is the seq2seq architecture, an NN system with an encoder and a decoder, with term sequence input for the encoder, intermediate hidden state output vector as decoder input, and term sequence output from the decoder (Cho et al., 2014; Sutskever, Vinyals, and Le, 2014).

Seq2seq follows Skip-gram (Mikolov et al., 2013) pre-training word embedding, which represents word context (or concept) as the word, in a vector. *Skip* refers to its order-neutrality. Given some word, consider the product of probabilities of the words in its context, being in its context (of length c words in either direction). The Skip-gram training maximizes the average of the (equivalent) sum of log-probabilities over all words as it computes the vectors.

Skip-gram is one of two methods by Mikolov et al. comprising word2vec. The other is a bag-of-words embedding. As later used by (Rush, Chopra, and Weston, 2015), it is a vector in which an element (indicator) is 1 if the associated word is present (in the sentence), and otherwise 0. A bag-of-words encoder takes the 1-components in the vector and multiplies them by a learned weight vector to fit into a hidden layer. Specifically, the vector is the product of a uniform distribution vector $\mathbf{p}^T$ and a matrix of weights, dividing the elements by the number of words.

An attention-based encoder (Bahdanau, Cho, and Bengio, 2014) is similar to the bag-of-words one except that $\mathbf{p}^T$ is now proportional to a product of the original input encoding and an encoding of existing output sentence words (the *context*). There is also a smoothing (by averaging) window applied to the source encoding. The result is a soft (as opposed to hard) alignment between source and target words.

Skip-Thought Vectors (Kiros et al., 2015) or sentence2vec are an extension of Skip-gram (*word2vec*) to sentences. This uses an objective function combining words of previous and next sentences in the hidden state sub-elements for a particular sentence. This can be used to generate (nonsensical) stories, when trained on the highly formulaic genre of romance novels:

> she grabbed my hand . " come on . " she fluttered her bag in the air . "
> i think we 're at your place . i can't come get you . " he locked himself

> back up . “ no . she will . ” kyrian shook his head . “ we met ... that congratulations ... said no . ” the sweat on their fingertips ’s deeper from what had done it all of his flesh hard did n’t fade . cassie tensed between her arms suddenly grasping him as her sudden her senses returned to its big form . her chin trembled softly as she felt something unreadable in her light . it was dark . my body shook as i lost what i knew and be betrayed and i realize just how it ended . it was n’t as if i did n’t open a vein . this was all my fault , damaged me . i should have told toby before i was screaming . i should ’ve told someone that was an accident . never helped it . how can i do this , to steal my baby ’s prints ? ”

To do better, there could be notions of coherence in narrative, to be extracted syntactically or otherwise from a sentence or lesser semantic particle. Mutual information (from information theory; or PMI, for pointwise mutual) evaluated on verb argument co-reference chains, is one such tool, introduced in (Chambers and Jurafsky, 2008) and called narrative coherence, based on the local coherence of (Grosz, Joshi, and Weinstein, 1995). They show that the event space can be clustered in discrete sets. They use procedural methods (directed graph) to produce scripts ordering the sets.

The ILP (integer linear programming) model described in 2.1.1 has an algorithm learning weights, in one of its forms (called the *discriminative*), although by contrast with a DNN method, it is less significant to its performance than a set of linguistic output constraints, such as the following: if the arguments of a verb are in the output, the verb must also be. The constraints are formulated as ILP inequalities using indicators and English part-of-speech (PoS) tagging (e.g. noun, adverb, conjunction), which is itself inherently imperfect.

On metrics: perplexity, the most general metric used in DL NLP (beside training steps), is 2^H, H being the model-corpus cross-entropy (e.g. average entropy for words in a sequence). The probability distribution is taken from the LM; typically in an NN meaning almost the same as its objective function (*loss*). Thus for example the char-LSTM result of Józefowicz et al. (Józefowicz et al., 2016), reporting state-of-the-art perplexity of 30.6 with their One Billion Word newswire corpus (Chelba et al., 2014) of shuffled sentences, produces mostly well-formed sentences, according to their examples. (This result is context-insensitive.)

ROUGE (Lin, 2004) is a commonly used set of recall-oriented metrics for summarization. Recall is the proportion of results that is relevant; compare with precision, the proportion of relevant items found. The machine translation metric BLEU (Papineni et al., 2002) is precision-oriented. F1 score balances recall and precision. The n-gram (2-grams are: foo bar; bar foo) variant of ROUGE is intuitively defined as the ratio of shared n-grams (by output and reference) to total n-grams (reference; for multiple references taking the closest one). The NAMAS evaluation uses ROUGE-1, ROUGE-2, and ROUGE-L, that is, letter, bigram, and LCS (longest common subsequence) variants of ROUGE, as they were used also in DUC-2004 and DUC-2003 datasets (Over, Dang, and Harman, 2007). (ROUGE-L, incidentally, is not a good metric (Toutanova et al., 2016)). DUC is a document understanding conference and competition that published small evaluation datasets for summarization. DUC-2004 is 500 news articles with 4 human-authored summaries for each.

2.1 Related Work

2.1.1 Sentence Compression

Given an input sentence, sentence compression produces a shorter sentence preserving meaning. Text deletion-based (i.e. extractive) models, used extensively, and newer, abstractive models also, employ word and phrase substitution and reordering, learned from data. A typically used corpus for abstractive compression or summarization training is (Annotated) English Gigaword (Napoles, Gormley, and Durme, 2012), which comprises ∼10m documents (4b words) of newswire (with headlines) and auto-generated syntactic annotations. A CNN seq2seq with attention summarizer by (Rush, Chopra, and Weston, 2015), trained on Gigaword, first-sentence to headline, in their evaluation outperforms an extractive ILP-based (integer linear programming) model by (Clarke and Lapata, 2008) and other baselines, although this is likely due to the nature of newswire headlines. (Toutanova et al., 2016) compare various metrics, including human evaluators and using four compression systems, and report an opposite relationship on a multi-genre corpus (based on MASC (Ide et al., 2008)), wherein ILP is state-of-the-art. The latter is a learning algorithm in one form but performance relies on linguistic constraints. (Filippova and Altun, 2013) report a method for building a parallel corpus for extractive compression from news headlines and first sentences. (Filippova et al., 2015) use it to learn LSTM deletion sequences (left to right) with a 2m pair news corpus, reporting 30% (versus 20%) perfect match, showing that syntactic features are not required in these DNN models. (Cohn and Lapata, 2008) showed an abstractive compression tree transduction model, learning substitution grammar rule weights with structural SVM. This model does not appear to be robust (Nomoto, 2009; Toutanova et al., 2016).

More recently, (F'evry and Phang, 2018) show controllable-length neural compression without a parallel training corpus by denoising autoencoders, learning to reconstruct a sentence from a list of its words and, as noise, some from another. Desired length is a decoder input.

2.1.2 Sentence Generation

Generation of full text sentences from a mapping, as in translation, is reliably done with seq2seq, which is considered mature, and we focus on it. Various decoding methods for diversity are found in the field of dialogue generation; however, not many are applicable to the task of expansion.

Fan et al. (Fan, Lewis, and Dauphin, 2018) collected a dataset of short (700 word) stories written for a sentence premise. They generate stories from sentence-length GAN-generated prompts. They use convolutional seq2seq with training-time model fusion (Sriram et al., 2018), and incorporate gated self-attention heads at different frequencies. Their outputs spread the prompt's concepts over many sentences. We train their story expansion seq2seq model on our sentence dataset for a baseline.

Style transfer and conversational models generate sentences from sentences, the latter with context. (Wang et al., 2017) use an inverted objective and decode using an MLP sample selector. They focus on a topic by feeding a grid-based topic embedding to the decoder.

In this work no attempt was made to learn latent literary style separately from meaning; arguably content makes the style. (Prabhumoye et al., 2018) learn the sentence itself as a latent variable before adversarially generating against style classifiers.

(Su et al., 2018) sample sentences in a MCMC process with a discriminator for constraints such as sentiment. (Ranzato et al., 2016) optimize sequence decoding based on BLEU or ROUGE scores, using reinforcement learning. (Zhao, Zhao, and Eskénazi, 2017) integrate expert knowledge in training a conditional VAE generating dialogue responses, with a classification such as opinion statement, yes/no question, agreement, etc.

Chapter 3

Methods

3.1 Methods

3.1.1 Modified Objective

Were we to use the seq2seq as is, it would result in a model that simply copies the input and does not expand it at all. This is because all the words that appear in a kernel (the compressed sentence) appear in the original (which we use as the expanded sentence). Merely learning to copy the input provides a relatively low loss (perplexity 8 at 50k steps, batch size 24, with dropout at 0.2). Early stopping salvages an inadequately trained model. Hence, we must provide an incentive for the model to add additional words not seen in the input, which appear in the expanded sentence.

To this end, we modify the negative log-likelihood loss of seq2seq to increase the importance of learning new words. The cross-entropy of every word in the target (expansion) that is not in the source is multiplied by a factor. We empirically chose 10 to balance effect and convergence. Perplexity becomes unusable for comparisons (immaterial under the circumstances (Theis, Oord, and Bethge, 2015); in any case, there is no one true story). Preserving words of the input sentence becomes challenging for the model; however, this does not necessarily diminish from the goal of improving the input. In addition, some sentences fail to terminate within the length limit (50 tokens). Nevertheless, output is longer and more diverse, even without random sampling. Let $\mathtt{I}$ denote the indicator function, T and S the unions of ground truth and source tokens, respectively, and $\lambda = 9$. The modified cross-entropy is given by

$$-\sum_{t}(1 + \lambda \mathtt{I}_{T-S}(w_t)) \log p(w_t|w_1, \ldots, w_{t-1}) \tag{3.1}$$

With this change, the model no longer degenerates to copying its input and can be trained for as long as desired.

3.1.2 Controlled Sampling

Random sampling is known to give diverse output. However, with softmax temperatures in the optimum range for our dataset ($0.3 - 0.7$), in poor expansions we observe arbitrary digressions suddenly halted by the attention search, and on the other end failures of randomness degenerating to the greedy (argmax) answer. As shown later (in Table 4.1), different sampling temperatures, width 10 beam search, and greedy decoding are empirically equivalent in effectiveness.

Generally, authors have an idea and see where to take it. In addition, principal plot twists, including the setting's setup, occur a significant distance from each other.

Therefore, we examine the concept of an interest curve in the setting of a single sentence.

First, to improve on standard random sampling we aim for a fixed degree of overall novelty in a sentence, so that it does not depend on sentence length. We use an accumulator, calculating per-word novelty as the difference from the softmax maximum. That is, with p and τ as the probability and corrected temperature at step t,

$$nov_w = \max_y \texttt{softmax}(\frac{\log p(y)}{\tau}) - \texttt{softmax}(\frac{\log p(w)}{\tau}) \tag{3.2}$$

We investigated adjusting temperature by rationing novelty over expected length with different self-adjusting curves, and by averaging a moving window. Two parabolic models we explored adjust the corrected temperature τ using Equation 3.3, where the left hand side is an integral over τ for temperature under the curve, and t is the remaining novelty (target minus accumulated). Solving for one of the parabola's parameters, b^2 or c, with the other set experimentally as a constant hyperparameter. a is the time (current step) divided by expected length.

$$\int_a^1 \left(b^2(x-0.5)^2 + c\right)\, dx = t \tag{3.3}$$

Due to spikes at the curve ends, this is combined with top 40 sampling, which reduces irrelevant generation and prevents the novelty quota from being exhausted too early. We adjust the free parameter, and in some models, scale, to reduce error between average expansion novelty and the target. We do not generate UNKs, and to reduce repetitiveness we penalize repeated words in a 5-token history, doubly penalizing content words.

The two models are named after the calculated parameter, b^2 or c, in Table 4.1. Other, less successful models were also implemented. One is an exponential on remaining novelty (up to expected length), which spikes τ somewhere inside the sentence, determined by a coefficient. Another is an a windowed accumulator, of size 3 or 5 tokens, balancing novelty, following either the target novelty or a parabola. The latter is illustrated in Figure 3.1. As shown, τ, the calculated temperature, starts high and words are chosen such that each diverges more from the original sentence: shortly, off, my. After 4 tokens τ drops as it enters the center of the parabola at half the expected length, producing words that connect the digression with the original sentence: left, comma, the. Before generating "tree" τ rises again producing the words huge, bare, beige, Christmas. At this point the curve is truncated to a limiting value $\epsilon = 0.1$. The value in orange shows the accumulated novelty over the past 3 tokens. The accumulator's effect is weak in this case and model, but can be seen on the parabola-like shape at steps 4 and 7, where respectively it decreases and increases τ to compensate for too much and too little novelty.

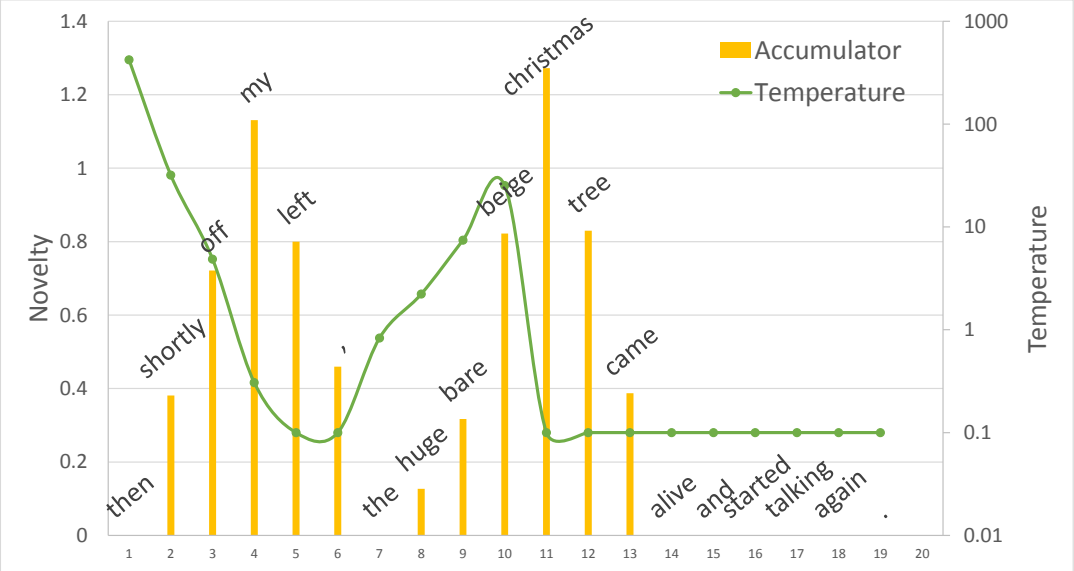

FIGURE 3.1: Decoding the human input “the tree came alive and started talking" to get “then shortly off my left , the huge bare beige christmas tree came alive and started talking again". Zoom in for outputs. This uses a truncated parabola model modified by a size-3 window accumulator (value in orange). Corrected temperature τ is in green.

Chapter 4

Experiments

4.1 Experiments

4.1.1 Corpus

Large public corpora of English fiction (e.g. Google Books, Internet Archive) have well-known quality issues with formatting, OCR, and content categorization (fiction? etc.). One exception is Project Gutenberg, which is proofread. Project Gutenberg's 19th and early 20th century public domain fiction and nonfiction has rather old styles of English, which we found transferred noticeably in our model.

Instead, we assembled a suitable corpus by scraping online resources for proofread 20th century English fiction. The collection consists of approximately 600m words and 41m sentences, 45% of which is speculative fiction. (It is noted that with many neologisms and domain terms in science fiction, many "literary" words may not appear in a 100k regular vocabulary.)

4.1.2 Optimization

In attempts at optimizing the quality of training, and to reduce the tendency for diverging phrase interpretations and expansions, the corpus was experimentally split into groups by topics. One method used was K-means clustering with a bag-of-words approach. Sentences were word-stemmed (with NLTK's (Bird, Klein, and Loper, 2009) Lancaster stemmer) and vectorized by either TF-IDF (in this scenario, how specific a word is to its sentence) or hashing (ignoring IDF). Latent semantic analysis (an SVD method for reducing dimensionality in term-document relationships) was optionally employed at several dimension parameters (50,100,200,300). The silhouette coefficient of cluster cohesion (defined as the average of scaled point distances to nearest different cluster) was highest (0.621) in the case with 10 clusters (for "genres"), with LSA to 200 components, 10k features, and counting words showing in 0.001% to 1% of sentences. In most cases the clusters were moderately self-similar in appearance and often could potentially be classified as, for example, "military", "bar/pub", "anatomy", or "Star Wars". Such clusters can be used as scenting sets for style priming. Unfortunately, in all cases one cluster was much larger than all others combined. As the clusters could not be balanced, this approach could not be used to split the corpus. Domain adaptation via other methods, such as in (Axelrod, He, and Gao, 2011), is an avenue for future work.

In a different approach, we considered genre qualifications embedded in the corpus. The romance genre, 7% of corpus, is considered fairly homogeneous and we trained a model on this subset, but did not find it competitive. Outputs were significantly and perhaps unsurprisingly colored by a focus on relations between objects (whether spatio-temporal or social); this however meant there were fewer novel

"idea" objects introduced, conflicting with the rationale for abstractive generation, as well as the reducing interest for other genre input. Because this would limit the possible forms of output, the genre subset was not used further.

4.1.3 Setup

Compressor

To generate sentence kernels, the system of (Rush, Chopra, and Weston, 2015) was tested using published code and data. Given the highly restricted nature of news prose, in its published configuration this system does not summarize fiction-style sentences convincingly compared to the ILP-based system by (Clarke and Lapata, 2008), which we then used (notwithstanding CPU-based processing not well-suited for large corpora). A standard KN-smoothing LM (Kneser and Ney, 1995) with $1e^{-}7$ pruning was used for compression.

Data

The corpus was cleaned of outliers (such as computing-related prose) and languages beside English using stopwords and inspection. Text was extracted and preprocessed to segmented sentence form by custom tokenization and segmentation, followed by CoreNLP (Manning et al., 2014). A large number of exceptional cases in punctuation or style across time, authors, and proofreaders requires that the process is imperfect and we saw some output with-excessive-hyphenation, among other issues.

In the implementation, due to a flaw in the compressor, 15% of collected data unfortunately could not be processed (books containing undetermined punctuation patterns). Additionally, neutral punctuation, especially quotation marks, often compressed incorrectly. 10% of sentences are in quotes; 3% of test set sentences have quotes outside words. Quotes were consequently removed; however, a model trained with them does generate dialogue and narration together.

Target compression was set to default 40%; average was 31%; together there are 1b words. A subset with 17m sentences where at least 30% reduction occurred has average length +1 and average compression 45%. The use of this set corresponds to the technique of separating short items from a neural model, and has similar observed advantage. This subset is the base for training. 3000 sentences were held for development. Sets were shuffled and lowercased, digits replaced by #.

Training

Models with 4 layers LSTM 1024 encoder and decoder were trained for a fixed 1m steps with batch size 24 and 0.2 dropout. Vocabulary size is 50k with SentencePiece. Names were not removed, in order that they may be synthesized directly (and thematically). Inevitably, a subword BPE vocabulary (Sennrich, Haddow, and Birch, 2016) tended to produce many nonsense words in our experiments (SentencePiece produced less). Sentences over 50 words (1.5%) had words in excess truncated.

Test

A test set of 100 sentences was collected from 20 workers on MTurk. Workers were asked to author a "short sentence that might have appeared in some imaginary story", with no example given, in batches of 5 per Web form. The adjective "short"

was used to discourage very long sentences that were stories, which were nevertheless seen in some of the answers. Lengths are balanced (with a mean of 12 words, standard deviation 5.2). The mean was affected by the size or width of workers' input text area in multiple rounds of collection. Sentences with profanity or political entities were filtered out. For each input and its expansion, 3 unique US workers with $> 96\%$ approval and 500 HITs were asked to choose the one they "think is better or more interesting", again with no example answers to minimize researcher bias. Sentences were shuffled and choice order randomized. Comparisons were also batched in groups of 5 per Web form to accommodate analysis. Since the task is very subjective in nature, workers often disagreed (Krippendorff's α, a chance-adjusted measure of reliability (Krippendorff, 1980), is 0.13 on a combined set of all preferences).

Expansions that failed to terminate within the length limit (50 tokens) or have clearly unnatural repetitiveness, detected via the regular expression `(.{10,})[^\r\n]{0,15}\1`, were removed, and the input sentences replaced. For completeness, we note the approval rate on these sentences is 28%, $p = 0.03$ by paired t-test. Removals reduced when penalties for repetitiveness were set.

4.1.4 Results

Table 4.1 presents the results in terms of human preferences of the expanded sentence over the input sentence, along with any metrics that have reached statistical significance for that sampling method. In all models, except the "original" model, the input sentences were the 100 sentences written by the Mechanical Turk workers.

"Parabola" refers to our method arising from Equation 3.3, with c or b^2 as the variable solved for. Baselines include:

1. using the modified objective, random sampling with temperature, beam search (width 10), and greedy search;
2. kernels held out from training, with original "expansions" (Table 4.3);
3. inserting a word by sampling LM trigram frequency, up to average rate of expansion by other methods;
4. Fan et al.'s (Fan, Lewis, and Dauphin, 2018) seq2seq fusion model, trained on our dataset for 500k steps, with outputs pruned of repeats in the same way. This is using the default top 10 sampling with temperature 0.8, comparable to other baselines; nonetheless we found that output length and diversity were relatively significantly random.

As depicted in the table, the human subjects preferred the original sentences (obtained from the original stories) to the compressed sentences (the kernels) 71.7% of the time. This is in fact our upper bound, as the expanded sentences were actual story lines. Our method, Parabola c, has outperformed all other baselines, and has reached human level equivalence with 50% of human subjects preferring it to the original human input.

Example expansions for human input sentences, with human preference data, are given in Table 4.2. These examples are chosen to compare across methods (including one exponential method mentioned, without preference data) and illustrate user preferences, which are difficult to predict. For additional comparison, some compression kernels and original sentences of writers (from the corpus) are given in Table 4.3.

TABLE 4.1: Sampling method evaluations.

Sampling	Preference	Significant metrics
Parabola c	**0.5**	
Parabola b^2	0.483	
Greedy	0.422	
Random 0.7	0.417	Frechet $r = 0.26$
Random 0.3	0.417	
Beam search	0.413	
Fan et al. s2s	0.3	
3-gram freq.	0.1	
Kernel vs. original	0.717	

TABLE 4.2: Example expansions for human input, with approval ratios.

Input		they were creeping around the corner when they heard a horrible scream .
BS 10	1/3	and then they were rushing around the corner , when they 'd first heard a faint scream , and then turned to look at each other 's eyes .
Sampling 0.7	0/3	and now , in all these other respects , they were both rushing back around the corner , when they 'd first heard a strange scream of distress , and then very quietly .
Input		there was a princess that lived in a castle .
BS 10	1/3	but there was also a princess that still lived in a small castle .
Sampling 0.7	2/3	but there was also a romance-a queen that lived in a larger , spacious , rambling castle.
Exp. 0.7	-	now , there was a new , high-ranking and female-american soul that lived in a castle .
Input		the meaning of life and reality at its core is, it can be what you want it to be.
Parabola b^2	1/3	but the meaning of a new life and reality, at its highest core – is like this, when you would all still want it to be there.
Input		the kind found his perfect princess.
Fan et al.	1/3	the kind of family found his perfect princess, the only one.
5-acc. 1.2	3/3	the kind of girl in there have been a small, princess with her name.

TABLE 4.3: Example kernels and original sentences, with approval ratios.

Kernel		he put a hand on gabriel 's shoulder and guided him .
Original	3/3	he put a hand on gabriel 's shoulder and guided him from the kitchen and into the shadows of the yard .
Kernel		he has feeling for others outside circle of friends and attaches value to life.
Original	1/3	he has little feeling for others outside a very small circle of friends , and attaches little real value to human life .

Metrics

Automatic metrics that we tested have low Pearson's r and Spearman's ρ with evaluator preferences ($|r| \leq 0.1$). This varied across sampling methods but generally not to the point of significance ($|r| \leq 0.2$). 2328 directly comparable preferences were collected in total.

The metrics computed were:

1. discrete Frechet and cosine distances in InferSent unsupervised sentence embeddings (Conneau et al., 2017);
2. ratio of unique added unigrams and bigrams to length (Dist-1 and Dist-2 (Li et al., 2016));
3. ROUGE-1, ROUGE-2, BLEU-2, BLEU-4 (Lin, 2004; Papineni et al., 2002);
4. expansion ratio, added words, and input and output lengths (the latter three with consistent $r \approx -0.1$, low negative, as expected).

No statistically significant differences in variance or mean were detected in subranges upon plotting of embedding distances and other metrics. Using InferSent we trained reference MLPs (Conneau and Kiela, 2018) on SICK dataset entailment and similarity (Marelli et al., 2014) and SNLI dataset entailment (Bowman et al., 2015), and again r was negligible. Training the MLPs on preference data for prediction, on a 10% test set we saw $r = 0.02$. Additionally, we manually evaluated a relation on the (100-sentence) beam search results subset, as its relatively generic output may extend to the preservation of meaning. Here (Table 4.4), for the accurate preservation of entities or concepts (58% of expansions) $r = 0.21$, and for strongly contradicting or changing the meaning (20%) $r = -0.18$. While we use unsupervised sentence embeddings, we do not have more elementary perplexity-based metrics, and we leave them to future work.

TABLE 4.4: Beam search (width 10) with manually evaluated entailment and InferSent distances.

r	Preserving (58%)	Contradicting (20%)	Frechet	Cosine	Dist-1
Preference	0.21	-0.18	-0.1	0.18	-0.12
Preserving	-	-	-0.24	0.48	-0.35

Discussion

For sentence expansion, a more relevant or diverse output is not necessarily better. Presumably, each evaluator has different expectations from an expansion, and learning these is important for an authoring tool. The automatic metric results illustrate the necessity of early human evaluation. Approval data given in Table 4.2 shows that preference can be counter-intuitive, perhaps due to the diverse population of MTurk workers. In one experiment, the two baseline methods Random 0.3 and trigram frequencies were compared, and the former were preferred 68.3% of the time; less than might be expected given the latter's performance in Table 4.2.

Expansion outputs sometimes contradict, and the frequency of this is not explained purely by decoding method and the compression removing "not"s. Given the test collection methodology, input phrases might be inclined toward cliché, while in the corpus clichés are much likelier to appear in a subverted form. Conversely, conceptually dense sentences such as adages or the already published writing of a veteran author are unlikely to gain from extension (in general style). Splitting off coherence from meaning does not appear to be useful to our goal; however, grammar remains a significant factor in user evaluations in our experiments.

In Table 4.4 we have correlation of manual entailment with InferSent distances ($r = 0.48$ for cosine) on beam search. If an entailment metric is considered reliable, it is easy to resample the output until entailment occurs; this does not seem to affect human preference, however.

As usual in text models, our system allows narrowing the possible style as desired, to some degree, using network bias, priming or scenting a pre-trained model with one author's books prior to decoding. We did not evaluate by humans the generation with specific author styles; nevertheless we give an example of the possibility in Table 4.5.

TABLE 4.5: Example outputs, general and primed style.

Input	he woke up .
Style: *general*	he woke up in the brush .
Style: *Adams*	he woke up , carefully .

Chapter 5

Conclusion

Writing is not easy for humans beyond what they already know, and DNNs do not have it easy because current approximations of ideas are functionally rudimentary. Lack of coherence seen in their predictions may be partially remediable by constructed notions of local coherence (Grosz, Joshi, and Weinstein, 1995), which is not order-neutral and cannot be simply translated to neural terms, it seems. While others have tried continuously re-integrating past choices, in this work we have exploited sentential structure to work around the limitation.

5.1 Conclusion and Future Work

We have defined a task and described our experiments in sentence expansion and enhancement. We take a sentence input from humans and produce a more literary, abstractively expanded sentence as output that equals the original, by human evaluation. The task is relevant in aids for writers, where it would save time and potentially improve quality. It is relevant in virtual agents in games, text ads, and other media benefitting from adaptable content. The task and our attempts and methods for a solution are novel according to our searches. We create a parallel corpus of fiction sentences and their compressions and train seq2seq models on the reverse to perform expansion. A modification to the objective function encourages learning output features and makes training at nontrivial length possible. Simple curve-based sampling methods distribute output novelty in a controlled way. Our model outputs, while not independently superior to human inputs, are shown to achieve parity and surpass baselines. With the necessary loss modification, 11% more of total expansions are preferred; with the best performing decoding method, 20% more of total; compared to a sentence adaptation of the model of Fan et al (Fan, Lewis, and Dauphin, 2018). We also observe that common metrics of text generation do not predict user preferences for this task.

Our expansions do not compete directly with human generated expansions, but this second context for comparison across methods will be useful in proving use of an expander indeed improves quality by saving time. In our estimate an untrained writer averages one minute to expand a sentence as the system does.

Future work includes building an assistive user interface for users to choose a sampling method and edit output, at the same time learning personal preferences, used as feedback to the method. This learning needs to be done by a deeper model than MLPs used here. Further work includes testing alternative platforms or enhancements to standard seq2seq and other tools which we have based our work on, and exploring threading of a number of sentences.

Bibliography

Axelrod, Amittai, Xiaodong He, and Jianfeng Gao (2011). "Domain Adaptation via Pseudo In-Domain Data Selection". In: *EMNLP 2011*.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). "Neural Machine Translation by Jointly Learning to Align and Translate". In: *CoRR* abs/1409.0473.

Bird, Steven, Ewan Klein, and Edward Loper (2009). "Natural Language Processing with Python". In:

Bowman, Samuel R. et al. (2015). "A large annotated corpus for learning natural language inference". In: *EMNLP*.

Bowman, Samuel R. et al. (2016). "Generating Sentences from a Continuous Space". In: *CoNLL*.

Chambers, Nathanael and Daniel Jurafsky (2008). "Unsupervised Learning of Narrative Event Chains". In: *ACL*.

— (2011). "Template-Based Information Extraction without the Templates". In: *ACL*.

Chelba, Ciprian et al. (2014). "One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling". In: *INTERSPEECH*.

Cho, Kyunghyun et al. (2014). "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *EMNLP*.

Cífka, Ondrej et al. (2018). "Eval all, trust a few, do wrong to none: Comparing sentence generation models". In: *CoRR* abs/1804.07972.

Clarke, James and Mirella Lapata (2008). "Global inference for sentence compression : an integer linear programming approach". In: *J. Artif. Intell. Res. (JAIR)* 31, pp. 399–429.

Cohn, Trevor and Mirella Lapata (2008). "Sentence Compression Beyond Word Deletion". In: *COLING*.

Conneau, Alexis and Douwe Kiela (2018). "SentEval: An Evaluation Toolkit for Universal Sentence Representations". In: *arXiv preprint arXiv:1803.05449*.

Conneau, Alexis et al. (2017). "Supervised Learning of Universal Sentence Representations from Natural Language Inference Data". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 670–680. URL: https://www.aclweb.org/anthology/D17-1070.

Dauphin, Yann et al. (2017). "Language Modeling with Gated Convolutional Networks". In: *ICML*.

Fan, Angela, Mike Lewis, and Yann Dauphin (2018). "Hierarchical Neural Story Generation". In: *ACL*.

F'evry, Thibault and Jason Phang (2018). "Unsupervised Sentence Compression using Denoising Auto-Encoders". In:

Filippova, Katja and Yasemin Altun (2013). "Overcoming the Lack of Parallel Data in Sentence Compression". In: *EMNLP*.

Filippova, Katja et al. (2015). "Sentence Compression by Deletion with LSTMs". In: *EMNLP*.

Grosz, Barbara J., Aravind K. Joshi, and Scott Weinstein (1995). "Centering: A Framework for Modeling the Local Coherence of Discourse". In: *Computational Linguistics* 21, pp. 203–225.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long Short-Term Memory". In: *Neural Computation* 9, pp. 1735–1780.

Ide, Nancy et al. (2008). "MASC: the Manually Annotated Sub-Corpus of American English". In: *LREC*.

Józefowicz, Rafal et al. (2016). "Exploring the Limits of Language Modeling". In: *CoRR* abs/1602.02410.

Kiros, Jamie Ryan et al. (2015). "Skip-Thought Vectors". In: *NIPS*.

Kneser, Reinhard and Hermann Ney (1995). "Improved backing-off for M-gram language modeling". In: *ICASSP*.

Krippendorff, Klaus (1980). "Content Analysis: An Introduction to Its Methodology". In:

Li, Boyang et al. (2013). "Story Generation with Crowdsourced Plot Graphs". In: *AAAI*.

Li, Boyang et al. (2014). "Storytelling with Adjustable Narrator Styles and Sentiments". In: *ICIDS*.

Li, Jiwei et al. (2016). "A Diversity-Promoting Objective Function for Neural Conversation Models". In: *HLT-NAACL*.

Lin, Chin-Yew (2004). "ROUGE: A Package For Automatic Evaluation Of Summaries". In:

Manning, Christopher D. et al. (2014). "The Stanford CoreNLP Natural Language Processing Toolkit". In: *Association for Computational Linguistics (ACL) System Demonstrations*, pp. 55–60. URL: http://www.aclweb.org/anthology/P/P14/P14-5010.

Marelli, Marco et al. (2014). "A SICK cure for the evaluation of compositional distributional semantic models". In: *LREC*.

Mikolov, Tomas et al. (2013). "Distributed Representations of Words and Phrases and their Compositionality". In: *CoRR* abs/1310.4546.

Napoles, Courtney, Matthew Gormley, and Benjamin Van Durme (2012). "Annotated English Gigaword". In: *Linguistic Data Consortium*.

Nomoto, Tadashi (2009). "A Comparison of Model Free versus Model Intensive Approaches to Sentence Compression". In: *EMNLP*.

Over, Paul, Hoa Dang, and Donna K. Harman (2007). "DUC in context". In: *Inf. Process. Manage.* 43, pp. 1506–1520.

Papineni, Kishore et al. (2002). "Bleu: a Method for Automatic Evaluation of Machine Translation". In: *ACL*.

Prabhumoye, Shrimai et al. (2018). "Style Transfer Through Back-Translation". In: *ACL*.

Propp, V (1928). *Morphology of the folktale*. Russian. Leningrad: Academia.

Ranzato, Marc'Aurelio et al. (2016). "Sequence Level Training with Recurrent Neural Networks". In: *CoRR* abs/1511.06732.

Riedl, Mark O. and Robert Michael Young (2010). "Narrative Planning: Balancing Plot and Character". In: *J. Artif. Intell. Res. (JAIR)* 39, pp. 217–268.

Rush, Alexander M., Sumit Chopra, and Jason Weston (2015). "A Neural Attention Model for Abstractive Sentence Summarization". In: *EMNLP*.

Schmidhuber, Jürgen (2015). "Deep Learning in Neural Networks: An Overview". In: *Neural networks : the official journal of the International Neural Network Society* 61, pp. 85–117.

Semeniuta, Stanislau, Aliaksei Severyn, and Sylvain Gelly (2018). "On Accurate Evaluation of GANs for Language Generation". In: *CoRR* abs/1806.04936.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (2016). "Neural Machine Translation of Rare Words with Subword Units". In: *CoRR* abs/1508.07909.

Sriram, Anuroop et al. (2018). "Cold Fusion: Training Seq2Seq Models Together with Language Models". In: *Interspeech*.

Su, Jinyue et al. (2018). "Incorporating Discriminator in Sentence Generation: a Gibbs Sampling Method". In: *CoRR* abs/1802.08970.

Sutskever, Ilya, James Martens, and Geoffrey E. Hinton (2011). "Generating Text with Recurrent Neural Networks". In: *ICML*.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le (2014). "Sequence to Sequence Learning with Neural Networks". In: *NIPS*.

Theis, Lucas, Aäron van den Oord, and Matthias Bethge (2015). "A note on the evaluation of generative models". In: *CoRR* abs/1511.01844.

Toutanova, Kristina et al. (2016). "A Dataset and Evaluation Metrics for Abstractive Compression of Sentences and Short Paragraphs". In: *EMNLP*.

Wang, Di et al. (2017). "Steering Output Style and Topic in Neural Response Generation". In: *EMNLP*.

Young, R. Michael and Johanna D. Moore (1994). "DPOCL: A Principled Approach to Discourse Planning". In: *CoRR* abs/cmp-lg/9406020.

Zhao, Tiancheng, Ran Zhao, and Maxine Eskénazi (2017). "Learning Discourse-level Diversity for Neural Dialog Models using Conditional Variational Autoencoders". In: *ACL*.