Generating Fake But Realistic Headlines Using Deep Neural Networks

Ashish Dandekar[1], Remmy A. M. Zen[2], Stéphane Bressan[3]

[1] National University of Singapore, Singapore
ashishdandekar@u.nus.edu
[2] Universitas Indonesia, Jakarta, Indonesia
remmy.augusta@ui.ac.id
[3] National University of Singapore, Singapore
steph@nus.edu.sg

Abstract. Social media platforms such as Twitter and Facebook implement filters to detect fake news as they foresee their transition from social media platform to primary sources of news. The robustness of such filters lies in the variety and the quality of the data used to train them. There is, therefore, a need for a tool that automatically generates fake but realistic news.
In this paper, we propose a deep learning model that automatically generates news headlines. The model is trained with a corpus of existing headlines from different topics. Once trained, the model generates a fake but realistic headline given a seed and a topic. For example, given the seed "Kim Jong Un" and the topic "Business", the model generates the headline "kim jong un says climate change is already making money".
In order to better capture and leverage the syntactic structure of the headlines for the task of synthetic headline generation, we extend the architecture - Contextual Long Short Term Memory, proposed by Ghosh et al. - to also learn a part-of-speech model. We empirically and comparatively evaluate the performance of the proposed model on a real corpora of headlines. We compare our proposed approach and its variants using Long Short Term Memory and Gated Recurrent Units as the building blocks. We evaluate and compare the topical coherence of the generated headlines using a state-of-the-art classifier. We, also, evaluate the quality of the generated headline using a machine translation quality metric and its novelty using a metric we propose for this purpose. We show that the proposed model is practical and competitively efficient and effective.

Keywords: Deep Learning, Natural Language Generation, Text Classification

1 Introduction

In the Digital News Report 2016[1], Reuters Institute for the Study of Journalism claims that 51% of the people in their study indicate the use of social media

[1] http://www.digitalnewsreport.org/survey/2016/overview-key-findings-2016/

platforms as their primary source of news. This transition of social media platforms to news sources further accentuates the issue of the trustworthiness of the news which is published on the social media platforms. In order to address this, social media platform like Facebook has already started working with five fact-checking organizations to implement a filter which can flag fake news on the platform[2].

Starting from the traditional problem of spam filtering to a more sophisticated problem of anomaly detection, machine learning techniques provide a toolbox to solve such a spectrum of problems. Machine learning techniques require a good quality training data for the filters to be robust and effective. To train fake news filters, they need a large amount of fake but realistic news. Fake news, which are generated by a juxtaposition of a couple of news without any context, do not lead to robust filtering. Therefore, there is a need of a tool which automatically generates a large amount of good quality fake but realistic news.

In this paper, we propose a deep learning model that automatically generates news headlines given a seed and the context. For instance, for a seed "obama says that", typical news headlines generated under *technology* context reads "obama says that google is having new surface pro with retina display design" whereas the headline generated under *business* context reads "obama says that facebook is going to drop on q1 profit". For the same seed with *medicine* and *entertainment* as the topics, typical generated headlines are "obama says that study says west africa ebola outbreak has killed million" and "obama says that he was called out of kim kardashian kanye west wedding" respectively.

We expect that the news headlines generated by the model should not only adhere to the provided context but also to conform to the structure of the sentence. In order to catch the attention of the readers, news headlines follow the structure which deviates from the conventional grammar to a certain extent. We extend the architecture of Contextual Long Short Term Memory (CLSTM), proposed by Ghosh et al. [9], to learn the part-of-speech model for news headlines. We compare Recurrent Neural Networks (RNNs) variants towards the effectiveness of generating news headlines. We qualitatively and quantitatively compare the topical coherence and the syntactic quality of the generated headlines and show that the proposed model is competitively efficient and effective.

Section 2 presents the related work. Section 3 delineates the proposed model along with some prerequisites in the neural network. We present experiments and evaluation in Section 4. Section 5 concludes the work by discussing the insights and the work underway.

2 Related Work

In the last four-five years, with the advancement in the computing powers, neural networks have taken a rebirth. Neural networks with multiple hidden layers, dubbed as "Deep Neural Networks", have been applied in many fields starting

[2] https://www.theguardian.com/technology/2016/dec/15/facebook-flag-fake-news-fact-check

from classical fields like multimedia and text analysis [11, 28, 18, 29] to more applied fields [32, 7]. Different categories of neural networks have been shown to be effective and specific to different kinds of tasks. For instance, Restricted Boltzmann Machines are widely used for unsupervised learning as well as for dimensionality reduction [13] whereas Convolutional Neural Networks are widely used for image classification task [18].

Recurrent Neural Networks [28] (RNNs) are used learn the patterns in the sequence data due to their ability to capture interdependence among the observations [12, 10]. In [5], Chung et al. show that the extensions of RNN, namely Long Short Term Memory (LSTM) [14] and Gated Recurrent Unit (GRU) [3], are more effective than simple RNNs at capturing longer trends in the sequence data. However, they do not conclude which of these gated recurrent model is better than the other. Readers are advised to refer to [22] for an extensive survey of RNNs and their successors.

Recurrent neural networks and their extensions are widely used by researchers in the domain of text analysis and language modeling. Sutskever et al. [29] have used multiplicative RNN to generate text. In [10], Graves has used LSTM to generate text data as well as images with cursive script corresponding to the input text. Autoencoder [13] is a class of neural networks which researchers have widely used for finding latent patterns in the data. Li et al. [19] have used LSTM-autoencoder to generates text preserving the multi-sentence structure in the paragraphs. They give entire paragraph as the input to the system that outputs the text which is both semantically and syntactically closer to the input paragraph. Tomas et al. [24, 25] have proposed RNN based language models which have shown to outperform classical probabilistic language models. In [26], Tomas et al. provide a context along with the text as an input to RNN and later predict the next word given the context of preceding text. They use LDA [2] to find topics in the text and propose a technique to compute topical features of the input which are fed to RNN along with the input. Ghosh et al. [9] have extended idea in [26] by using LSTM instead of RNN. They use the language model at the level of a a word as well as at the level of a sentence and perform experiments to predict next word as well as next sentence given the input concatenated with the topic. There have been evidences of LSTM outperforming GRU for the task of language modeling [16, 15]. Nevertheless, we compare our proposed model using both of these gated recurrent building blocks. We use the simple RNN as our baseline for the comparison.

Despite these applications of deep neural networks on the textual data, there are few caveats in these applications. For instance, although in [9] authors develop CLSTM which is able to generate text, they evaluate its predictive properties purely using objective metric like perplexity. The model is not truly evaluated to see how effective it is towards generating the data. In this paper, our aim is to use deep neural networks to generate the text and hence evaluate the quality of synthetically generated text against its topical coherence as well as grammatical coherence.

3 Methodology

3.1 Background: Recurrent Neural Network

Recurrent Neural Network (RNN) is an adaptation of the standard feedforward neural network wherein connections between hidden layers form a loop. Simple RNN architecture consists of an input layer (x), a hidden layer (h), and an output layer (y). Unlike the standard feedforward networks, the hidden layer of RNN receives an additional input from the previous hidden layer. These recurrent connections give RNN the power to learn sequential patterns in the input. We use the many-to-many variant of RNN architecture which outputs n-gram given the previous n-gram as the input. For instance, given {*(hello, how, are)*} trigram as the input, RNN outputs {*(how, are, you)*} as the preceding trigram.

Bengio et al. [1] show that learning the long-term dependencies using gradient descent becomes difficult because the gradients eventually either vanish or explode. The gated recurrent models, LSTM [14] and GRU [3], alleviate these problems by adding gates and memory cells (in the case of LSTM) in the hidden layer to control the information flow. LSTM introduces three gates namely forget gate (f), input gate (i), and output gate (o). Forget gate filters the amount of information to retain from the previous step, whereas input and output gate defines the amount of information to store in the memory cell and the amount of information to transfer to the next step, respectively. Equation 1 shows the formula to calculate the forget gate activations at a certain step t. For given layers or gates m and n, W_{mn} denotes the weight matrix and b_m is the bias vector for the respective gate. h is the activation vector for the hidden state and $\sigma(\cdot)$ denotes the sigmoid function. Readers are advised to refer to [14] for the complete formulae of each gate and layer in LSTM.

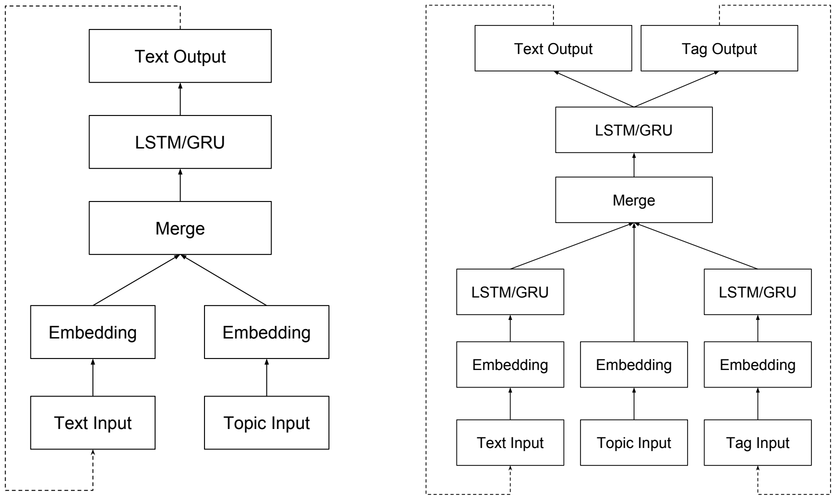
$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \tag{1}$$

GRU simplifies LSTM by merging the memory cell and the hidden state, so there is only one output in GRU. It uses two gates which are update and reset gate. Update gate unifies the input gate and the forget gate in LSTM to control the amount of information from the previous hidden state. The reset gate combines the input with the previous hidden state to generate the current hidden state.

3.2 Proposed Syntacto-Contextual Architecture

Simple RNNs predict the next word solely based on the word dependencies which are learnt during the training phase. Given a certain text as a seed, the seed may give rise to different texts depending on the context. Refer to the Section 1 for an illustration. [9] extends the standard LSTM to Contextual Long Short Term Memory (CLSTM) model which accepts the context as an input along with the text. For example, an input pair {*(where, is, your), (technology)*} generates an output like {*(is, your, phone)*}. CLSTM is a special case of the architecture shown in Figure 1a using LSTM as the gated recurrent model.

In order to use the model for the purpose of text generation, contextual information is not sufficient to obtain a good quality output. A good quality text is coherent not only in terms of its semantics but also in terms of its syntax. By providing the syntactic information along with the text, we extend the contextual model to Syntacto-Contextual (SC) models. Figure 1b shows the general architecture of the proposed model. We encode the patterns in the syntactic meta information and input text using the gated recurrent units and, later, merge them with the context. The proposed model not only outputs text but also corresponding syntactic information. For instance, an input {(where, is, your), (adverb, verb, pronoun), (technology)} generates output like {(is, your, phone), (verb, pronoun, noun)}.



(a) Contextual Architecture [9]   (b) Syntacto-Contextual Architecture

Fig. 1: Contextual and Syntacto-Contextual Architectures

Mathematically, the addition of context and syntactic information amounts to learning a few extra weight parameters. Specifically, in case of LSTM , Equation 1 will be modified to Equation 2 and Equation 3, for CLSTM and SCLSTM respectively. In Equation 2 and Equation 3, p represents topic embedding and s represents embedding of the syntactic information.

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f + \mathbf{W}_{\mathbf{pf}}\mathbf{p}_{\mathbf{t}}) \tag{2}$$

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f + \mathbf{W}_{\mathbf{pf}}\mathbf{p}_{\mathbf{t}} + \mathbf{W}_{\mathbf{sf}}\mathbf{s}_{\mathbf{t}}) \tag{3}$$

For the current study, we annotate the text input with the part-of-speech tags using Penn Treebank tagset [23]. We learn the parameters of the model using stochastic gradient descent by minimizing the loss for both output text and output tags. We, also, work on a variation of the contextual architecture which does not accept topic as an input and uses conventional RNN instead of LSTM. This model is treated as the baseline against which all of the models will be compared.

For each of the model, we embed the input in a vector space. We merge the inputs by column wise concatenation of the vectors. We perform experiments using both LSTM and GRU as the gated recurrent units. The output layer is a softmax layer that represents the probability of each word or tag. We sample from that probability to get the next word and tag output.

4 Experimentation and Results

We conduct a comparative study on five different models using a real-world News Aggregator Dataset. In the beginning of this section, we present the details of the datset and the experimental setup for the study. We, further, describe various quality metrics which we use to evaluate the effectiveness of the models. We perform quantitative analysis using these metric and present our results. We complete the evaluation by presenting micro-analysis for a sample of generated news headlines to show the qualitative improvement observed in the task of news headline generation.

4.1 Dataset

We use the News Aggregator dataset[3] consisting of the news headlines collected by the news aggregator from 11,242 online news hostnames, such as *time.com*, *forbes.com*, *reuters.com*, etc. between 10 March 2014 to 10 August 2014. The dataset contains 422,937 news articles divided into four categories, namely business, technology, entertainment, and health. We randomly select 45000 news headlines, which contain more than three words, from each category because we give trigram as the input to the models. We preprocess the data in two steps. Firstly, we remove all non alpha-numeric characters from the news titles. Secondly, we convert all the text into lower case. After the preprocessing, the data contains 4,274,380 unique trigrams and 39,461 unique words.

4.2 Experimental Setup

All programs are run on Linux machine with quad core 2.40GHz Intel® Core i7™processor with 64GB memory. The machine is equipped with two Nvidia GTX 1080 GPUs. Python® 2.7.6 is used as the scripting language. We use a high-level neural network Python library, *Keras* [4] which runs on top of *Theano* [30]. We use categorical cross entropy as our loss function and use ADAM [17] as an optimizer to automatically adjust the learning rate.

We conduct experiments and comparatively evaluate five models. We refer to those models as, **baseline** - a simple RNN model, **CLSTM** - contextual

[3] https://archive.ics.uci.edu/ml/datasets/News+Aggregator

architecture with LSTM as the gated recurrent model, **CGRU** - contextual architecture with GRU as the gated recurrent model, **SCLSTM** - syntacto-contextual architecture with LSTM as the gated recurrent model, **SCGRU** - syntacto-contextual architecture with GRU as the gated recurrent model, in the rest of the evaluation. All inputs are embedded into a 200-dimensional vector space. We use recurrent layers each with 512 hidden units with 0.5 dropout rate to prevent overfitting. To control the randomness of the prediction, we set the temperature parameter in our output softmax layer to 0.4. We use the batch size of 32 to train the model until the validation error stops decreasing.

4.3 Evaluation Metrics

In this section, we present different evaluation metrics that we use for the quantitative analysis. Along with purely objective quantitative metrics such as perplexity, machine translation quality metric, and topical precision, we use metrics like grammatical correctness, n-gram repetition for a finer effectiveness analysis. Additionally, we devise a novelty metric to qualitatively analyse the current use case of news headline generation.

Perplexity is commonly used as the performance measure [9, 10, 16, 15] to evaluate the predictive power of a language model. Given N test data with w_t as the target outputs, the perplexity is calculated by using Equation 4, where $p^i_{w_t}$ is the probability of the target output of sample i. A good language model assigns a higher probability to the word that actually occurs in the test data. Thus, a *language model with lower perplexity is a better model.*

$$perplexity = 2^{-\frac{1}{N}\sum_{i=1}^{N}\log p^i_{w_t}} \tag{4}$$

As it happens, the exponent in the Equation 4 is the approximation of cross-entropy[4], which is the loss function we minimize to train the model, given a sequence of fixed length.

Although the task under consideration of the presented work is not of a word or a topic prediction, we simply use perplexity as a purely objective baseline metric. We complement it by using various application specific measures in order to evaluate the effectiveness of the quality of the generated text.

Topical coherence refers to the extent to which the generated text adheres to the desired topic. In order to evaluate the topical coherence, one requires a faithful classifier which predicts the topic of generated text. We treat the topics predicted by the classifier as the ground truth to quantitatively evaluate the topical coherence. The proposed method generates a news headline given a seed and a topic of the news. People have widely used Multinomial naive Bayes classifier to deal with text data due to independence among the words given a certain class[5]. We train a Multinomial naive Bayes classifier with Laplace smoothing on the news dataset consisting of 45000 news from each of the four categories. We hold out 20% of the data for validation. By proper tuning of the

[4] http://cs224d.stanford.edu/lecture_notes/notes4.pdf

[5] https://www.kaggle.com/uciml/news-aggregator-dataset

smoothing parameter, we achieve 89% validation accuracy on the news dataset. We do not use this metric for the baseline model.

Taking existing text as a reference, a **quality** metric evaluates the effectiveness of the generated text in correspondence to the reference. Such a metric measures the *closeness* of the generated text to the reference text. Metrics such as BLEU [27], Rouge [8], NIST [21] are widely used to evaluate the quality of machine translation. All of these metrics use "gold standard", which is either the original text or the text written by the domain experts, to check the quality of the generated text. We use BLEU as the metric to evaluate the quality of generated text. For a generated news headline, we calculate its BLEU score by taking all the sentences in the respective topic from the dataset as the reference. Interested readers should refer to [33] for a detailed qualitative and quantitative interpretation of BLEU scores.

With the motivation of the current work presented in the Section 1, we want the generated text from our model to be as *novel* as possible. So as to have a robust fake news filter, the fake news, which is used to train the model, should not be a juxtaposition of few existing news headlines. More the patterns it learns from the training data to generate a single headline, more novel is the generated headline. We define **novelty** of the generated output as the number of unique patterns the model learns from the training data in order to generate that output. We realize this metric by calculating longest common sentence common to the generated headline and each of the headline in the dataset. Each of these sentences stands as a pattern that the model has learned to generate the text. *Novelty* of a generated headline is taken as the number of unique longest common sentences.

The good quality generated text should be both *novel* and grammatically correct. **Grammatical correctness** refers the judgment on whether the generated text adheres to the set of grammatical rules defined by a certain language. Researchers either employ experts for evaluation or use advanced grammatical evaluation tools which require the gold standard reference for the evaluation [6]. We use an open-source grammar and spell checker software called LanguageTool[6] to check the grammatical correctness of our generated headlines. LanguageTool uses NLP based 1516 English grammar rules to detect syntactical errors. Aside from NLP based rules, it used English specific spelling rules to detect spelling errors in the text. To evaluate grammatical correctness, we calculate the percentage of grammatically correct sentences as predicted by the LanguageTool.

We find that LanguageTool only recognizes word repetition as an error. Consider a generated headline *beverly hills hotel for the first in the first in the world* as an example. In this headline, there is a trigram repetition - *the first in* - that passes LanguageTool grammatical test. Such headlines are not said to be good quality headlines. We add new rules with a regular expression to detect such repetitions. We count **n-gram repetitions** within a sentence for values of n greater than two.

[6] API and Java package available at https://languagetool.org

4.4 Results

To generate the output, we need an initial trigram as a seed. We randomly pick the initial seed from the set of news headlines from the specified topic. We use windowing technique to generate the next output. We remove the first word and append the output to the back of the seed to generate the next output. The process stop when specified sentence length is generated. We generate 100 sentences for each topic in which each sentence contains 3 seed words and 10 generated words.

Quantitative Evaluation Table 1 summarizes the quantitative evaluation of all the models using metrics described in Section 4.3. Scores in bold numbers denote the best value for each metric. We can see that for Contextual architecture, GRU is a better gated recurrent model. Conversely, LSTM is better for Syntacto-Contextual architecture.

For Syntacto-Contextual architecture, we only consider the perplexity of the text output to make a fair comparison with the Contextual architecture. We analyze that our Syntacto-Contextual architecture has a higher perplexity score because the model jointly minimizes both text and syntactical output losses. On the other hand, the baseline model has a low perplexity score because it simply predicts the next trigram with control on neither the context nor the syntax.

A high score on classification precision substantiates that all of these models generate headlines which are coherent with the topic label with which they are generated. We observe that all of the models achieve a competitive BLEU score. Although Contextual architecture performs slightly better in terms of BLEU score, Syntacto-Contextual architecture achieves a higher novelty score. In the qualitative evaluation, we present a more detailed comparative analysis of BLEU scores and novelty scores.

We observe that the news headlines generated by Syntacto-Contextual architecture are more grammatically correct than other models. Figure 2 shows the histogram of n-gram repetitions in the generated news headline. We see that the Syntacto-Contextual architecture gives rise to news headlines with less number of n-gram repetitions.

Lastly, we have empirically evaluated, but not presented here, the time taken by different models for one epoch. CLSTM takes 2000 seconds for one epoch whereas SCLSTM takes 2131 seconds for one epoch. Despite the Syntacto-Contextual architecture being a more complex architecture than Contextual architecture, it shows that it is competitively efficient.

Qualitative Evaluation. Table 2 presents the samples of generated news from CLSTM proposed by [9] and SCLSTM, which outweighs the rest of the models in the quantitative analysis.

In Table 1, we see that the Contextual architecture models receive a higher BLEU score than the proposed architecture models. BLEU score is calculated using n-gram precisions with the news headlines as the reference. It is not always

Table 1: Quantitative Evaluation.

	Baseline	**CLSTM[9]**	**CGRU**	**SCLSTM**	**SCGRU**
Perplexity	108.383	119.10	**92.22**	146.93	175.83
Topical Coherence (%)	-	84.25	77.25	**94.75**	87.50
Quality (BLEU)	0.613	0.637	**0.655**	0.633	0.625
Novelty	21.605	24.67	25.21	**26.57**	25.65
Grammatical Correctness (%)	28.25	49.75	50.75	**75.25**	69.00
n-gram Repetitions	11	30	12	**5**	8

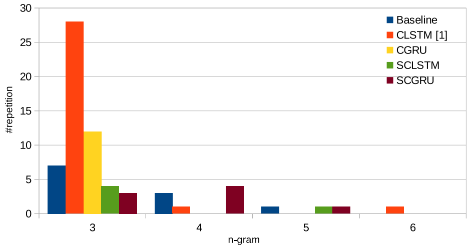


Fig. 2: n-gram repetition analysis

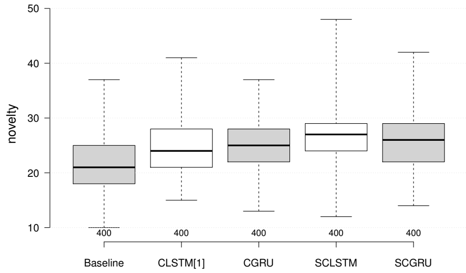


Fig. 3: Boxplot for novelty metric

necessary that the higher BLEU score leads towards a good quality text generation. Qualitative analysis of generated headlines shows that the higher BLEU score, in the most cases, is the result of the juxtaposition of the existing news headlines. For instance, consider a headline generated by CLSTM model as an example - “justin bieber apologizes for racist joke in new york city to take on” - which receives a BLEU score of 0.92. When we search for the same news in the dataset, we find that this generated news is a combination of two patterns from the following two headlines, “justin bieber apologizes for racist joke” and “uber temporarily cuts fares in new york city to take on city cabs”. Whereas the headline generated by SCLSTM with the same seed is quite a novel headline. In the training dataset there is mention of neither Justin Bieber joking on Twitter nor joke for gay fans. Similar observation can be made with the news related to Fukushima. In the training data set there is no news headline which links Fukushima with climate change. Additionally, there is no training data which links higher growth risk to climate change as well. Thus, we observe that the headlines generated using SCLSTM are qualitatively better than CLSTM.

All of the models presented in the work are probabilistic models. Text generation being a probabilistic event, on the one hand it is possible that contextual architecture generates a good quality headline at a certain occasion. For instance, we see that CLSTM also generates some good quality news headlines such as “the fault in our stars trailer for the hunger games mockingjay part teaser”. On the other hand, it is possible that Syntacto-Contextual architecture generates some news headline with poor quality or repetitions, such as “obama warns google apple to make android support support for mobile for mobile”. In order

Table 2: Generated News Headlines

Category	**CLSTM** [9]	**SCLSTM**
Medicine	first case chikungunya virus in saudi arabia reports new mers cov in the	first case chikungunya virus found in florida state health care system draws attention
	mosquitoes test positive for west africa in guinea in june to be the	mosquitoes test positive for west nile virus found in storage room at home
	ticks and lyme disease in the us in the us are the best	ticks and lyme disease rates double in autism risk in china in west
Business	us accuses china and russia to pay billion in billion in us in	us accuses china of using internet explorer bug leaves users to change passwords
	wake of massive data breach of possible to buy buy stake in us	wake of massive recall of million vehicles for ignition switch problem in china
	japan fukushima nuclear plant in kansas for first time in the first time	japan fukushima nuclear plant linked to higher growth risk of climate change ipcc
Entertainment	justin bieber apologizes for racist joke in new york city to take on	justin bieber apologizes for racist joke joke on twitter for gay fans have
	the fault in our stars trailer for the hunger games mockingjay part teaser	the fault in our stars star chris hemsworth and elsa pataky reveal his
	giant practical spaceship interiors for joint venture in mexico production of star wars	giant practical spaceship hits the hollywood walk of fame induction ceremony ceremony in
Technology	first android wear watches and google be to be available in the uk	first android wear watch google play edition is now available on xbox one
	samsung sues newspaper for anti vaccine and other devices may be the best	samsung sues newspaper over facebook experiment on users with new profile feature is
	obama warns google glass to be forgotten on the us government issues recall	obama warns google apple to make android support support for mobile for mobile

to qualitatively analyse the novelty of generated sentence, we need to observe how likely such events occur. Figure 3 shows the boxplot of novelty numbers we calculate for each of 400 generated news headlines using different models. As discussed earlier, we want our model to generate novel news headlines. So, we prefer higher novelty scores. Although the mean novelty of all of the models lie around 24, we see that SCLSTM is more likely to generate the novel headlines. Additionally, we observe that contextual and Syntacto-Contextual architectures performs better than the baseline model.

As mentioned in the quantitative evaluation, Contextual architecture gives rise to news headlines with a large number of n-gram repetitions. In an extreme case, CLSTM model generates the following headline, "lorillard *inc nyse wmt wal mart stores* inc nyse wmt wal mart stores", that contains 6-gram repetition. The news headline generated by CLSTM - "samsung sues newspaper for anti vaccine and other devices may be the best"- exemplifies the smaller topical coherence observed for the Contextual architecture models.

In order to garner the opinion of real-world users, we use CrowdFlower[7] to conduct a crowdsource based study. In this study, we generate two news headlines

[7] https://www.crowdflower.com/

using CLSTM and SCLSTM using the same seed and ask the workers to choose a more realistic headline between two. We generate such a pair of headlines for 200 different seeds. Each pair is evaluated by three workers and majority vote is used to choose the right answer. At the end of the study, 66% workers agree that SCLSTM generates more realistic headlines than CLSTM.

5 Discussion and Future work

In [9], Ghosh et al. proposed a deep learning model to predict the next word or sentence given the context of the input text. In this work, we adapted and extended their model towards automatic generation of news headlines. The contribution of the proposed work is two-fold. Firstly, in order to generate news headlines which are not only topically coherent but also syntactically sensible, we proposed an architecture that learns part-of-speech model along with the context of the textual input. Secondly, we performed thorough qualitative and quantitative analysis to assess the quality of the generated news headlines using existing metrics as well as a novelty metric proposed for the current application. We comparatively evaluated the proposed models with [9] and a baseline. To this end, we show that the proposed approach is competitively better and generates good quality news headlines given a seed and the topic of the interest.

Through this work, we direct our methodology for data-driven text generation towards a "constraint and generate" paradigm from a more brute-force way of "generate and test". Quality assessment of the generated data using generative model remains an open problem in the literature [31]. We use the measure of quality, which in our case is the grammatical correctness, as an additional constraint for the model in order to generate the good quality data. The usage of POS tags as the syntactic element is mere a special case in this application. We can think of more sophisticated meta information to enrich the quality of text generation. Ontological categories can be an alternative option.

Acknowledgement

This work is partially supported by the National Research Foundation, Prime Ministers Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme and by the National University of Singapore under a grant from Singapore Ministry of Education for research project number T1 251RES1607. We kindly acknowledged UCI for maintaining repository of publicly available Machine Learning dataset [20].

References

1. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks 5(2), 157–166 (1994)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of machine Learning research 3(Jan), 993–1022 (2003)

3. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
4. Chollet, F.: Keras. https://github.com/fchollet/keras (2015)
5. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
6. Dahlmeier, D., Ng, H.T.: Better evaluation for grammatical error correction. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 568–572. Association for Computational Linguistics (2012)
7. Deng, L., Yu, D., et al.: Deep learning: methods and applications. Foundations and Trends® in Signal Processing 7(3–4), 197–387 (2014)
8. Doddington, G.: Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In: Proceedings of the second international conference on Human Language Technology Research. pp. 138–145. Morgan Kaufmann Publishers Inc. (2002)
9. Ghosh, S., Vinyals, O., Strope, B., Roy, S., Dean, T., Heck, L.: Contextual lstm (clstm) models for large scale nlp tasks. arXiv preprint arXiv:1602.06291 (2016)
10. Graves, A.: Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850 (2013)
11. Graves, A., Jaitly, N., Mohamed, A.r.: Hybrid speech recognition with deep bidirectional lstm. In: Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on. pp. 273–278. IEEE (2013)
12. Graves, A., Mohamed, A.r., Hinton, G.: Speech recognition with deep recurrent neural networks. In: Acoustics, speech and signal processing (icassp), 2013 ieee international conference on. pp. 6645–6649. IEEE (2013)
13. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. science 313(5786), 504–507 (2006)
14. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
15. Irie, K., Tüske, Z., Alkhouli, T., Schlüter, R., Ney, H.: Lstm, gru, highway and a bit of attention: an empirical overview for language modeling in speech recognition. in INTERSPEECH (2016)
16. Jozefowicz, R., Zaremba, W., Sutskever, I.: An empirical exploration of recurrent network architectures. In: in the Proceedings of The 32nd ICML. pp. 2342–2350 (2015)
17. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
19. Li, J., Luong, M.T., Jurafsky, D.: A hierarchical neural autoencoder for paragraphs and documents. arXiv preprint arXiv:1506.01057 (2015)
20. Lichman, M.: UCI machine learning repository (2013), http://archive.ics.uci.edu/ml
21. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Proceedings of the ACL-04 Workshop. vol. 8 (2004)
22. Lipton, Z.C., Berkowitz, J., Elkan, C.: A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019 (2015)
23. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of english: The penn treebank. Computational linguistics 19(2), 313–330 (1993)

24. Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., Khudanpur, S.: Recurrent neural network based language model. In: Interspeech. vol. 2, p. 3 (2010)
25. Mikolov, T., Kombrink, S., Burget, L., Černockỳ, J., Khudanpur, S.: Extensions of recurrent neural network language model. In: Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. pp. 5528–5531. IEEE (2011)
26. Mikolov, T., Zweig, G.: Context dependent recurrent neural network language model. SLT 12, 234–239 (2012)
27. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting on association for computational linguistics. pp. 311–318. Association for Computational Linguistics (2002)
28. Sutskever, I.: Training recurrent neural networks. Ph.D. thesis, University of Toronto (2013)
29. Sutskever, I., Martens, J., Hinton, G.E.: Generating text with recurrent neural networks. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11). pp. 1017–1024 (2011)
30. Theano Development Team: Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints abs/1605.02688 (May 2016), http://arxiv.org/abs/1605.02688
31. Theis, L., Oord, A.v.d., Bethge, M.: A note on the evaluation of generative models. arXiv preprint arXiv:1511.01844 (2015)
32. Wang, H., Wang, N., Yeung, D.Y.: Collaborative deep learning for recommender systems. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1235–1244. ACM (2015)
33. Zhang, Y., Vogel, S., Waibel, A.: Interpreting bleu/nist scores: How much improvement do we need to have a better system? In: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC). pp. 2051–2054