Proofread Sentence Generation as Multi-Task Learning with Editing Operation Prediction

Yuta Hitomi [1] **Hideaki Tamori** [1] **Naoaki Okazaki** [2] **Kentaro Inui** [3]
[1] Media Lab, The Asahi Shimbun Company
[2] Tokyo Institute of Technology
[3] Tohoku University, RIKEN Center Advanced Intelligence Project
{hitomi-y1, tamori-h}@asahi.com, okazaki@c.titech.ac.jp
inui@ecei.tohoku.ac.jp

Abstract

This paper explores the idea of *robot editors*, automated proofreaders that enable journalists to improve the quality of their articles. We propose a novel neural model of multi-task learning that both generates proofread sentences and predicts the editing operations required to rewrite the source sentences and create the proofread ones. The model is trained using logs of the revisions made professional editors revising draft newspaper articles written by journalists. Experiments demonstrate the effectiveness of our multi-task learning approach and the potential value of using revision logs for this task.

1 Introduction

There is growing research interest in automatic sentence generation (Vinyals et al., 2015; Rush et al., 2015; Sordoni et al., 2015). Coincidentally (or inevitably), media companies have increasingly attempted to create *robot journalists* that can automatically generate content, mostly using data from limited domains (e.g., earthquakes, sports and stockmarkets) (Clerwall, 2014; Carlson, 2015; Dorr, 2016). In this paper, we explore the idea of *robot editors*, i.e., automated proofreaders that enable journalists to improve the quality of their articles.

The most closely related field to the topic of this paper is grammatical error correction (Dale and Kilgarriff, 2011; Dale et al., 2012; Ng et al., 2013, 2014). However, this task handles only grammatical errors, whereas proofreading encompasses a variety of tasks:grammatical error correction (GEC), spell checks, simplification, fact checking, standardization, compression, paraphrasing, etc. Another task, the automated evaluation of scientific writing shared task (Daudaravicius et al., 2016), has the goal of automatically evaluating scientific writing. The focus of this shared task was the binary classification problem of detecting sentences that need improvement. Although the corpus used contained qualitative improvements, the shared task did not tackle high-quality sentence generation.

This paper investigates the task of proofread sentence generation (PSG) using logs of the revisions made by professional editors to draft newspaper articles written by journalists. The goal of this research is to explore a computational model for improving text quality. To this end, we propose a novel multi-task learning approach that both generates proofread sentences and predicts the editing operations involved in rewriting the source sentences to create the proofread ones.

The contributions of this research are three-folds: (i) This is the first study to explore an encoderdecoder architecture for PSG. (ii) We show that our proposed multi-task learning method can outperform a state-of-the-art baseline method for GEC. (iii) We also examine the benefits and issues of using revision logs for PSG.

2 Method

Given a source sentence (sequence of words) $x_1, \cdots, x_m$, this study addresses the task of generating the proofread sentence $y_1, \cdots, y_n$, where m and n denote the number of words in the source and proofread sentences, respectively. Usually, proofreading does not change the content of the input text significantly, and changes only small parts of the text. Thus, detecting source sentence spans that require revision is an important PSG sub-task. This paper explores a multi-task learning approach that both generates proofread sentences and predicts the editing operations required.

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 436–441,
Taipei, Taiwan, November 27 – December 1, 2017 ©2017 AFNLP

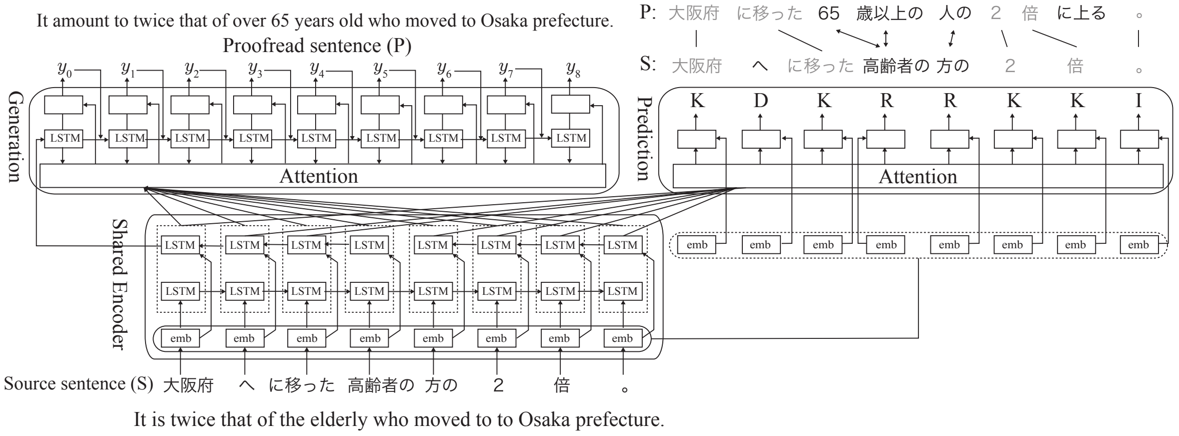
It amount to twice that of over 65 years old who moved to Osaka prefecture.
Proofread sentence (P)
y_0 y_1 y_2 y_3 y_4 y_5 y_6 y_7 y_8
Generation
LSTM
Attention
P: 大阪府 に移った 65 歳以上の 人の 2 倍 に上る 。
S: 大阪府 へ に移った 高齢者の 方の 2 倍 。
Prediction
K D K R R K K I
Attention
emb
Shared Encoder
LSTM
emb
Source sentence (S) 大阪府 へ に移った 高齢者の 方の 2 倍 。
It is twice that of the elderly who moved to to Osaka prefecture.

Figure 1: Overview of the neural network model for generating proofread sentences and predicting editing operations via multi-task learning. Boxes labeled 'emb' denote word embeddings.

Inspired by work on multi-task learning in neural networks, we implement multi-task learning as an end-to-end neural network with a shared source sentence encoder (Figure 1). The network generates proofread sentences and predicts editing operations.Although these two tasks solve the same problem, we believe that these two neural network models focus on different aspects of the data. The generation model considers the source sentence as a whole (although it may also include an attention mechanism), whereas the prediction model looks at the local contexts of words in order to correct functional word usage, incorrect spellings, and so on. This is why we have designed the proposed method using a multi-task learning approach.

2.1 Generating proofread sentences using an encoderdecoder model with attention

Following recent work on GEC (Yuan and Briscoe, 2016), we use an encoder-decoder model with global attention (Luong et al., 2015) to generate proofread sentences. We use bi-directional Long Short-Term Memory (LSTM) to encode the source sentences. LSTMs recurrently compute the memory and hidden vectors at time step $s \in \{1, \cdots, m\}$ using those at time step $s-1$ or $s+1$ and the word x_s in the source sentence, as follows:

$$[\overrightarrow{\boldsymbol{h}}_s; \overrightarrow{\boldsymbol{c}}_s] = \overrightarrow{\mathrm{LSTM}}(x_s, [\overrightarrow{\boldsymbol{h}}_{s-1}; \overrightarrow{\boldsymbol{c}}_{s-1}]), \quad (1)$$

$$[\overleftarrow{\boldsymbol{h}}_s; \overleftarrow{\boldsymbol{c}}_s] = \overleftarrow{\mathrm{LSTM}}(x_s, [\overleftarrow{\boldsymbol{h}}_{s+1}; \overleftarrow{\boldsymbol{c}}_{s+1}]). \quad (2)$$

Here, $\boldsymbol{c}_s$ and $\boldsymbol{h}_s$ represent the memory and hidden vectors, respectively, at time step s. The encoder output is a concatenation of the hidden vectors:

$$\widetilde{\boldsymbol{h}}_{\boldsymbol{s}} = [\overrightarrow{\boldsymbol{h}}_s; \overleftarrow{\boldsymbol{h}}_s]. \quad (3)$$

The decoder computes the memory and hidden vectors at time step $t \in \{1, \cdots, n\}$ using those at time step $t-1$ and the (predicted) word w_t as follows:

$$[\boldsymbol{h}_t; \boldsymbol{c}_t] = \overrightarrow{\mathrm{LSTM}}(w_t, [\boldsymbol{h}_{t-1}; \boldsymbol{c}_{t-1}]). \quad (4)$$

Here, we set $\boldsymbol{h}_0 = \overleftarrow{\boldsymbol{h}}_1$ and $\boldsymbol{c}_0 = \boldsymbol{0}$[1]. For $t > 1$, we feed the y_{t-1} predicted at the previous time step back as the input word w_t of the decoder.

The decoder predicts the word y_t at time step t using a softmax layer on top of the vector $\bar{\boldsymbol{h}}_{\boldsymbol{t}}$ with an integrated attention mechanism:

$$\begin{aligned}
\log p(y|x) &= \sum_{n=1}^{t} \log p(y_t|y_{<t}, \boldsymbol{x}), && (5)\\
p(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x}) &= \mathrm{softmax}(\boldsymbol{W_o}\bar{\boldsymbol{h}}_{\boldsymbol{t}}), && (6)\\
\bar{\boldsymbol{h}}_{\boldsymbol{t}} &= \tanh(\boldsymbol{W_r}[\boldsymbol{v_t}; \boldsymbol{h_t}]), && (7)\\
\boldsymbol{v_t} &= \sum_s \alpha_t(s)\widetilde{\boldsymbol{h}}_{\boldsymbol{s}}, && (8)\\
\alpha_t(s) &= \frac{\exp(\boldsymbol{h}_t^{\intercal}\boldsymbol{W_a}\widetilde{\boldsymbol{h}}_{\boldsymbol{s}})}{\sum_{s'}\exp(\boldsymbol{h}_t^{\intercal}\boldsymbol{W_a}\widetilde{\boldsymbol{h}}_{\boldsymbol{s'}})}. && (9)
\end{aligned}$$

Here, $\boldsymbol{v}_t$ represents a vector computed by the attention mechanism at time step t, and $\alpha_t(s)$ is an attention score computed at decoding time step t by looking at the source word at encoding time step s.

[1] The reason for using $\overleftarrow{\boldsymbol{h}}_1$ instead of $\widetilde{\boldsymbol{h}}_{\boldsymbol{1}}$ is to speed up training by reducing the dimensionality of the vectors in the decoder.

2.2 Editing-operation prediction as a sequential labeling task

In addition to performing sentence revision, we propose to simultaneously undertake an additional task: *editing-operation prediction*. Formally, given a source sentence $x_1, \cdots, x_m$, this task predicts the sequence of editing operations $z_1, \cdots, z_m$ required to obtain the revised sentence $y_1, \cdots, y_n$, where each editing operation z_i keeps (K), deletes (D), inserts (I), or replaces (R) the word x_i. This task resembles grammatical error detection.

As explained in Section 3.1, the revision logs do not provide supervised training data for the editing operations, only pairs of source and proofread sentences. We therefore created pseudo supervised training data by running the `diff` program on the word sequences of the source and proofread sentences. There are cases where multiple sequences of primitive editing-operations can be derived from a pair of sentences. The performance may be improved if the best operation sequence. However, we leave this direction out of scope of this paper. To obtain an editing-operation sequence that was the same length as the source sentence (m), we labeled I labels to the words immediately following the positions where the actual insert operations were required.

We reuse the vector $\widetilde{\boldsymbol{h}}_{\boldsymbol{s}}$ for word x_s in the source sentence (Equation 3) to predict z_s:

$$\log p(z|x) = \sum_{s=1}^{m} \log p(z_s|z_{<s}, \boldsymbol{x}), \qquad (10)$$

$$p(z_s|\boldsymbol{z}_{<s}, \boldsymbol{x}) = \mathrm{softmax}(\boldsymbol{W_l} \tanh(\widetilde{\boldsymbol{h}}_{\boldsymbol{i}})). \qquad (11)$$

Liu and Liu (2016) proposed a grammatical error detection method that considers intra-attention. We also explored this approach by replacing the softmax function in Equation 11 with the following one:

$$\begin{aligned} p(z_s|\boldsymbol{z}_{<s}, \boldsymbol{x}) &= \mathrm{softmax}(\boldsymbol{W_l}\bar{\boldsymbol{u}}_t) && (12)\\ \bar{\boldsymbol{u}}_t &= \tanh(\boldsymbol{W_s}[\boldsymbol{x_t}; \boldsymbol{u_t}]) && (13)\\ \boldsymbol{u}_s &= \sum_i \beta_s(i)\widetilde{\boldsymbol{h}}_{\boldsymbol{s}} && (14)\\ \beta_s(i) &= \frac{\exp(\widetilde{\boldsymbol{h}}_{\boldsymbol{s}}^{\intercal}\widetilde{\boldsymbol{h}}_{\boldsymbol{i}})}{\sum_{i'} \exp(\widetilde{\boldsymbol{h}}_{\boldsymbol{s}}^{\intercal}\widetilde{\boldsymbol{h}}_{\boldsymbol{i'}})} && (15) \end{aligned}$$

Here, $\boldsymbol{u}_s$ represents a vector computed by the attention mechanism at time step s, and $\beta_s(i)$ is an attention score computed at time step s by looking at the source word at time step $i \in \{1, \cdots, m\}$.

Dataset	# changed	# unchanged	# total
Train	710,540	1,317,260	2,027,800
Validation	63,062	96,938	160,000
Test	458	642	1,100

Table 1: The Numbers of instances in each dataset.

2.3 Training

Given a training dataset $\mathcal{D}$, we minimize the following multi-task learning loss function:

$$-\sum_{(x,y,z)\in\mathcal{D}} \{\log p(y|x) + \log p(z|x)\}. \qquad (16)$$

We also consider a loss function that weights the sub-task loss (Zhang et al., 2014):

$$-\sum_{(x,y,z)\in\mathcal{D}} \{\log p(y|x) + \lambda \log p(z|x)\}. \qquad (17)$$

Here, λ ($0 \leq \lambda \leq 1$) denotes the weight given to the editing-operation prediction errors, learned through gradient descent. In contrast to conventional multi-task learning, which maximizes performance for all tasks, our primary goal is to optimize the main task which is why we have created a loss function that weights the sub-task differently.

3 Experiments

3.1 Dataset

We used a corpus of Japanese newspaper articles where professional editors at a media company had rewritten draft articles (written by journalists) to create proofread (published) (Tamori et al., 2017). The dataset consists of 2,209,249 sentence pairs in total: 810,227 pairs were changed during the revision process, and 1,399,022 pairs were left unchanged. To focus on revisions within sentence boundaries, we excluded revisions involving sentence splitting or merging. This dataset reflects the real work done by the company to improve the quality of its newspaper articles. The revisions came in a variety of forms, from syntactic changes, such as GEC and spelling normalization to content-level changes, such as elaboration and fact checking. Table 1 shows the number of instances in the training, validation, and test sets.

We split the sentences into words using SentencePiece[2] to efficiently reduce the vocabulary

[2] https://github.com/google/sentencepiece SentencePiece is an unsupervised text tokenizer.

Dataset	Model	GLEU	Prec	Recall	$F_{0.5}(M^2)$	WER	BLEU
All pairs	Gen	70.14	*	*	*	24.90	74.02
	Gen + Pred	70.47	*	*	*	24.23	**75.37**
	Gen + Pred Attn	**70.68**	*	*	*	**23.90**	74.48
	Gen + Pred Attn + W	70.57	*	*	*	37.24	54.76
Changed pairs	Gen	67.74	22.27	6.12	14.23	36.74	64.23
	Gen + Pred	68.10	23.14	5.59	13.37	35.67	**66.29**
	Gen + Pred Attn	**68.63**	24.89	6.28	14.84	**35.55**	65.31
	Gen + Pred Attn + W	67.01	**36.59**	**13.30**	**26.59**	48.91	45.82
Unchanged pairs	Gen	86.72	*	*	*	16.47	82.69
	Gen + Pred	87.34	*	*	*	**15.51**	**83.33**
	Gen + Pred Attn	**87.44**	*	*	*	16.17	82.51
	Gen + Pred Attn + W	87.27	*	*	*	28.94	62.66

Table 2: Performance of the proposed and baseline methods. The asterisks '*' indicate performance values that are unavailable because the precision and recall for unchanged pairs are always 0 and 100.

size. We used proofread sentences (published newspaper articles) to train SentencePiece. Vocabulary sizes for the input and output layers were 32,661 and 32,630, respectively. Our model can be trained without unknown words.

3.2 Experimental setup

The batch size was set to 100 and, improve computational efficiency, each batch consisted of sentences of the same length. The dimensionality of the distributed representations (word embeddings and hidden states) to was 300. The model parameters were trained using Adam. Following Jozefowicz et al. (2015), forget gate bias was initialized to 1.0, and the other gate biases were initialized to 0. In addition, we used dropout (at a rate of 0.2) for the LSTMs. Breadth-first search was used for decoding, with a beam width of 10 (Yuan, 2017).

Six measures were utilized to evaluate the performance of the PSG model: GLEU (Napoles et al., 2015)[3], precision, recall, M^2 score (Dahlmeier and Ng, 2012), Word Error Rate (WER) (Jurafsky and Martin, 2008), and BLEU (Papineni et al., 2002). These measures are often used in GEC and machine translation research. Note that the precision, recall, and M^2 score measures excluded words appearing in both the source and proofread sentences from evaluation (Dahlmeier and Ng, 2012).

3.3 Results

Table 2 shows the performance of the proposed method according to the above metrics. Two variants of the proposed method were used: "Gen + Pred" used Equation 11 (without the attention mechanism for predicting edit operations), whereas "Gen + Pred Attn" used Equation 12 (with the attention mechanism), and "Gen + Pred Attn + W" used Equations 12 and 17 (weighting the sub-task losses). For comparison, we also report the performance of a baseline method "Gen" used only an encoder-decoder model (described in Section 2.1) without multi-task learning.

The multi-task learning models outperformed the baseline encoder-decoder model for all metrics. The use of intra-attention for predicting editing-operations was also effective, except for the BLEU metric and the WER for unchanged pairs. Unlike BLEU, GLEU includes a mechanism for penalizing incorrect 'reluctant' revisions (copying words from the source sentences) and rewarding correct 'aggressive' revision (adding words that do not appear in the source sentences). We can therefore infer that "Gen + Pred with Attn" model was more aggressive in changing words in the source sentences than "Gen + Pred" model.

The table also shows the models' performances for changed/unchanged sentence pairs. As expected, the performance metrics for unchanged pairs are higher than those for changed pairs. The low recall values for changed pairs indicate that it is difficult to predict words that do not appear in the source sentences.

However, we can see that weighting sub-task losses improved precision, recall, and M^2 score performance. We believe that active proofreading performance improved because of not over-

[3] The hyperparameter λ of GLEU was set to 1.

learning the sub-task. That said, the other metrics decreased because the model made changes in many places where no editing was required.

3.4 Discussion

In this section, we discuss issues with the presented method and dataset. The presented method was not good at inserting words, particularly when the editor had appended new information to the sentence, as in the following example.

- **Source**: Soon it will have been six months since the law was established.
- **Proofread**: *On the 19th,* it will have been six months since the law was established *last September.*

It is difficult for a computational model to insert the phrases "on the 19th" and "last September". We found that 132 of the 366 changed pairs (28.8%) in the test set included new information. Although this kind of editing improves the quality of the article, it is unrealistic to handle this situation using an encoder-decoder model. It may be useful to separate these pairs from the dataset in future.

We also observed instances in which editors merged pieces of information from multiple sentences, particularly so as to yield more concise sentences, as in the following example.

- **Source**: A meeting where people who heve lost their families to cancer discuss ... .There are ten members in their 30s to 70s *who have lost their families to cancer.*
- **Proofread**: A meeting where people who have lost their families to cancer discuss ... .There are ten members in their 30s to 70s.

As a result, it may be worth exploring not only discarding instances where additional information has been added to the proofread sentences but also building a model that considers other sentences appearing near the source sentence.

We also noticed cases where the sentences output by the model provide good revisions but did not match to the reference sentences, such as the following.

- **Source**: The reserve players tackle it frantically *so they won't* miss the chance.
- **Proofread**: The reserve players tackle it frantically *so as not to* miss *the* opportunity.
- **System output**: The reserve players tackle it frantically *so as not to* miss *the chance*.

As has often been discussed in the literature on machine translation, summarization, and GEC, establishing a good evaluation metric is an ongoing research issue.

4 Conclusion

In this paper, we have presented a novel multi-task learning approach for combined PGS and editing-operation prediction. Experimental results show that our approach was able to outperform the baseline for all metrics. The experiments also show that newspaper article revision logs can provide promising supervised training data for the model. We plan to continue exploring ways of creating good-quality articles in the future.

Acknowledgments

We thank the anonymous reviewers for their suggestions. We also thank Takafumi Ochiai and Yosuke Fukada for their contribution to storing the dataset.

References

Matt Carlson. The robotic reporter. *Digital Journalism*, 3(3):416–431, 2015.

Christer Clerwall. Enter the robot journalist. *Journalism Practice*, 8(5):519–531, 2014.

Daniel Dahlmeier and Hwee Tou Ng. 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 (NAACL 2012)*, pages 568–572, 2012.

Robert Dale and Adam Kilgarriff. Helping our own: The HOO 2011 pilot shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation (ENLG 2011)*, pages 242–249, 2011.

Robert Dale, Ilya Anisimoff, and George Narroway. HOO 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 54–62, 2012.

Vidas Daudaravicius, Rafael E Banchs, Elena Volodina, and Courtney Napoles. A report on the automatic evaluation of scientific writing shared task. In *Proceedings of the 11th Workshop on Innovative*

Use of NLP for Building Educational Applications, pages 53–62, 2016.

Konstantin Nicholas Dorr. Mapping the field of algorithmic journalism. *Digital Journalism*, 4(6):700–722, 2016.

Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2342–2350, 2015.

Daniel Jurafsky and James H. Martin. *Speech and Language Processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall, 2008.

Zhuoran Liu and Yang Liu. Exploiting unlabeled data for neural grammatical error detection. *CoRR*, abs/1611.08987, 2016.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 1412–1421, 2015.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015)*, pages 588–593, 2015.

Hwee Tou Ng, Yuanbin Wu, and Christian Hadiwinoto. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task (CoNLL Shared Task 2013)*, pages 1–12, 2013.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task (CoNLL Shared Task 2014)*, pages 1–14, 2014.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 311–318, 2002.

Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 379–389, 2015.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

Hideaki Tamori, Yuta Hitomi, Naoaki Okazaki, and Kentaro Inui. Analyzing the revision logs of a japanese newspaper for article quality assessment. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pages 46–50, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*, 2015.

Zheng Yuan. Grammatical error correction in non-native English. Technical Report 904, University of Cambridge, Computer Laboratory, 2017.

Zheng Yuan and Ted Briscoe. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*, pages 380–386, 2016.

Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *European Conference on Computer Vision*, pages 94–108. Springer, 2014.