Learning to Represent Edits

My path working with edits, in a nutshell

Edison Marrese-Taylor

07/06/2023

National Institute of Advanced Industrial Science and Technology
The University of Tokyo

Table of contents

About Me

About me: Timeline

https://epochx.github.io/

- Industrial Engineer from the University of Chile
- Went to Japan in 2013 as a graduate MEXT scholar
- Did my Ph.D. at the University of Tokyo, under the supervision of Yutaka Matsuo http://ymatsuo.com/, after graduation I stayed there as a postdoc
- On April 2021 I became researcher at AIST https://www.airc.aist.go.jp/en/kirt/, and continued as a visiting assistant professor at University of Tokyo

About me (2)

- **Research**
  - Interested in multi-modality, specifically on video-and-language (I'm intentionally leaving this topic for a future talk)
  - Learning to understanding and represent source code and natural language edits (Loyola et al., 2017, 2018; Marrese-Taylor et al., 2019, 2020)
- **Misc**: Broad interest in affect in text, including emotion (Marrese-Taylor and Matsuo, 2017; Balazs et al., 2018) and irony detection (Ilić et al., 2018)
- **Committee Member**: NAACL, EMNLP, ACL, INLG, AAAI
- **Education**: I have been teaching an undergrad class on Introduction to Machine Learning from 2018 to 2022, Co-guiding 1 (+2) Ph.D., 4 (+2) Master's and 2 Interns at the University of Tokyo.

Understanding Source Code Changes on GitHub

with Pablo Loyola and Yutaka Matsuo

Motivation

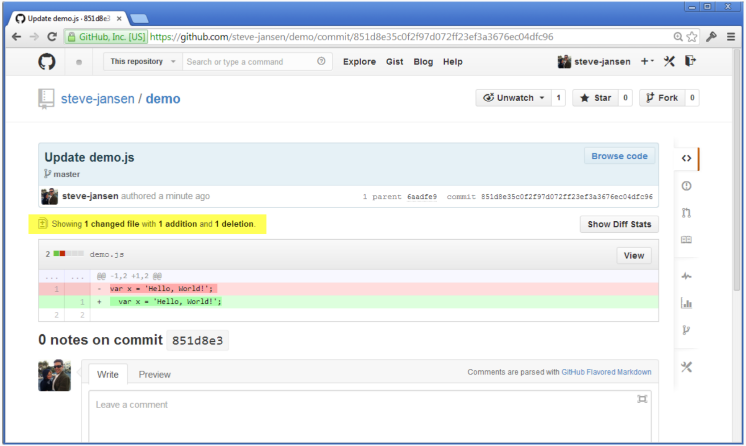
Source code inherently reflects human intent

- It encodes the way we command a machine to perform a task
- It is expected that it follows distributional regularities that a proper natural language manifests
- Allows an indirect way of communication between developers

Automatic code summarization methods

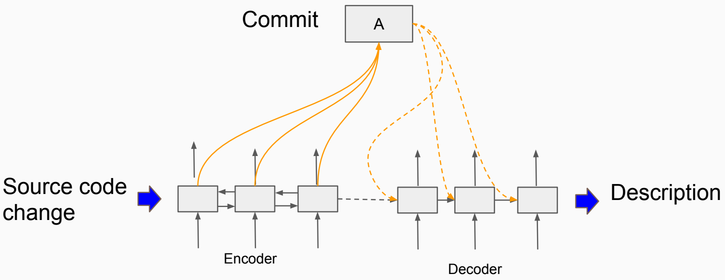
- Can help provide relevant insights to developers, but is static.
- Software development can be seen as a sequence of incremental changes
- Source code changes are critical for understanding program evolution so **how can we extend it to encode code changes into natural language representations?**

Idea: Code Commits in GitHub



Proposed approach

- Encoder-Decoder with a global attention mechanism is used to learn more expressive portions of the sequences (Loyola et al., 2017).
- During testing we use **beam search** to approximate the most likely message.
- Evaluation based on BLEU-4 the standard metric to evaluate machine translation models.



Experiments and Results

- Data collected from 4 programming languages, ranging 12 active large scale programs.
  Atomicity assumption: one file-change per commit
- Baseline: MOSES treating the problem as a phrase-based translation task.

Dataset	*atomic*			*full*	
	Val. acc	**BLEU**	**Moses**	**Val. acc**	**BLEU**
Theano	36.81%	9.5	7.1	39.88%	10.9
keras	45.76%	13.7	7.8	59.30%	8.8
youtube-dl	50.84%	16.4	17.5	53.65%	17.7
node	52.46%	7.8	7.7	53.70%	7.2
angular	44.39%	13.9	11.7	45.06%	15.3
react	49.44%	11.4	10.7	48.61%	12.1
opencv	50.77%	11.2	9.0	49.00%	8.4
CNTK	48.88%	17.9	11.8	44.85%	9.3
bitcoin	50.04%	17.9	13.0	55.03%	15.1
CoreNLP	63.20%	28.5	10.1	62.25%	26.7
elasticsearch	36.53%	11.8	5.2	35.98%	6.4
guava	65.52%	29.8	19.5	67.15%	34.3

Experiments and Results (2)

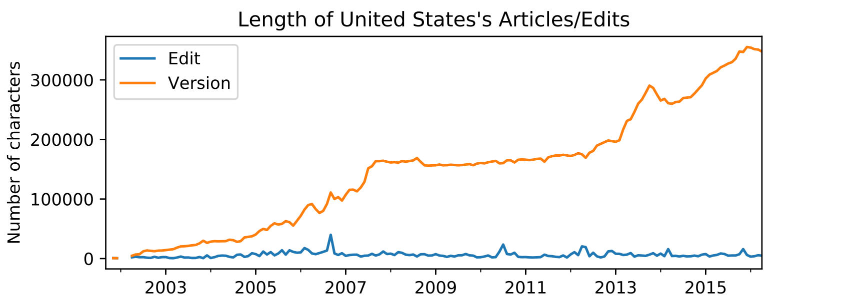
	Reference	Generated
keras	Fix image resizing in preprocessing/image Fix test flakes	Fixed image preprocessing . Fix flaky test
Theano	fix crash in the new warning message . remove var not used . Better error msg	Better warning message . remove not used code . better error message .
youtube-dl	[crunchyroll] Fix uploader and upload date extraction [extractor/common] Improve base url construction [mixcloud] Use unicode_literals	[crunchyroll] Fix uploader extraction [extractor/common] Improve extraction [common] Use unicode_literals
opencv	fixed gcc compilation remove unused variables in OCL_PERF_TEST_P ( )	fixed compile under linux remove unused variable in the module

Article Quality Assessment on Wikipedia

with Pablo Loyola and Yutaka Matsuo

Motivation

- Assessing the quality of Wikipedia articles is critical for maintaining its reputation and credibility.
- Existing approaches for quality assessment are:
    - Static (no time dependency is considered).
    - Work at the document-level.
    - Based on a set of predefined hand-crafted features (ORES).
- **Problem:** Article size grows over time, hard to scale.



Proposed Approach (1)

Idea: A model that receives as input only the edit and **returns a measure of article quality**. As edits are usually accompanied by a short description, we explore whether learning to generate this description could help improve quality assessment.

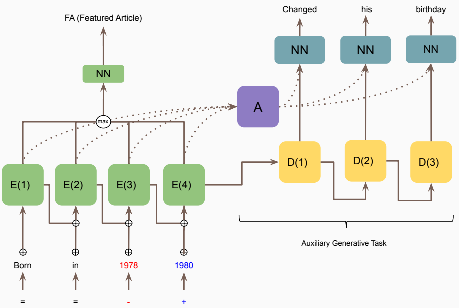
We tokenize each sentence and then use a standard *diff* algorithm to compare each sequence pair, and build an *edit-sentence* based on the alignment, containing **added**, **deleted** and **unchanged** tokens, with the token-level labels $+$, $-$ and $=$. For example:

$+$	**Errare humanum est**, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.	-	**Ut enim ad minim veniam**, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

$$\underset{+}{\text{Errare}}\ \underset{+}{\text{humanum}}\ \underset{+}{\text{est}}\ \underset{-}{\text{Ut}}\ \underset{-}{\text{enim}}\ \underset{-}{\text{ad}}\ \underset{-}{\text{minim}}\ \underset{-}{\text{veniam}}\ \underset{=}{,}\ \underset{=}{\text{quis}}\ \underset{=}{\text{nostrud}}\ \underset{=}{\text{exercitation}}\ \underset{=}{\text{ullamco}}\ \underset{=}{\text{laboris}}\ \underset{=}{\text{nisi}}$$

$$\underset{=}{\text{ut}}\ \underset{=}{\text{aliquip}}\ \underset{=}{\text{ex}}\ \underset{=}{\text{ea}}\ \underset{=}{\text{commodo}}\ \underset{=}{\text{consequat}}\ \underset{=}{.}$$

Proposed Approach (2)



- Quality assessment is a **multi-class classification** with labels Stub $\leq$ Start $\leq$ C $\leq$ B $\leq$ GA $\leq$ FA (Warncke-Wang et al., 2013).
- We incorporate edit messages by adding an **auxiliary generative task**, modeled using seq2seq. This loss is added to the classification cross entropy using a weight based on parameter λ.

Experiments

- **Data**: we use some of the most edited articles for English and German Wikipedia and obtain quality data using the ORES API (Warncke-Wang et al., 2013), as a silver standard.
- **Evaluation**: For the classification we used accuracy on the validation set for hyper-parameter tuning and evaluation, and also measured macro-averaged F1-Score. The generative task is evaluated using BLEU.

Model	F1-Score	Accuracy	BLEU
Regular	0.47	0.74	-
+ *edit-sentence*	0.56	**0.80**	-
+ *diff* tags	0.62	0.78	-
+ Generation $\lambda = 0.2$	0.28	0.61	0.25
+ Generation $\lambda = 0.5$	0.33	0.68	0.24
+ Generation $\lambda = 0.8$	0.41	0.77	0.25
+ Generation $\lambda = 0.9$	**0.65**	0.77	0.22
Only Generation ($\lambda = 0$)	-	-	0.23

Results: Summary

Dataset Model		Test F1-Score	Test Accuracy	Test BLEU
Barack Obama	C	0.62	**0.91**	-
	C+G	**0.66**	0.88	0.20
Donald Trump	C	**0.47**	**0.78**	-
	C+G	**0.47**	0.77	0.20
Guns n' Roses	C	0.18	**0.84**	-
	C+G	**0.30**	0.81	0.20
Xbox 360	C	0.30	0.61	-
	C+G	**0.32**	**0.63**	0.31
Chicago	C	0.38	**0.72**	-
	C+G	**0.39**	0.71	0.29
Pink Floyd	C	0.35	0.80	-
	C+G	**0.37**	**0.80**	0.35
Manchester United F.	C	0.17	0.72	-
	C+G	**0.39**	**0.77**	0.43
Wikiclass	C	0.40	0.40	-

Variational Inference for Learning Representations of Natural Language Edits

with Machel Reid and Yutaka Matsuo

Motivation

- Editing documents has become a pervasive component of many human activities (Miltner et al., 2019).
- **Is it possible to automatically extract rules from these common edits?**.
    - Yes! Learning distributed representations of edits (Yin et al., 2019)
- **Can we do better?**

A Generative Model for Edits

Proposal

A task based on self-supervision to learn edit representations, where:

- $x_-^{(i)}$ is the original version of an object
- $x_+^{(i)}$ its form after a change has been applied

Then, we assume the following generative process to obtain $x_+^{(i)}$ from $x_-^{(i)}$:

$$p(\mathbf{x}_+|\mathbf{x}_-) = \int_{\mathbf{z}} p(\mathbf{x}_+, z|\mathbf{x}_-)d_z = \int_{\mathbf{z}} p(\mathbf{x}_+|z, \mathbf{x}_-)p(z)d_z \tag{1}$$

Where $\mathbf{x}_+$ and $\mathbf{x}_-$ are observed random variables associated to $x_+^{(i)}$ and $x_-^{(i)}$ respectively, and z **represents a continuous latent variable that models the edit process**.

To evaluate models, we propose Performance Evaluation of Edit Representations (PEER).

PEER: Motivation

Intrinsic evaluation

→ No external data

- Gold-standard performance of the editor (token-level accuracy).
- Visual inspection of the semantic similarity of neighbors in latent space.
- Clustering and visual inspection of clusters.

Extrinsic Evaluation

→ External data required

- Visual inspection of the 2D-projected edit space on edits for a certain label.
- One-shot performance of the editor on similar edits.
- Ability to capture other properties of the edit (one or many labels associated).

We propose to resort to automatic and more standard evaluations, using BLEU-4, as well as GLEU (Napoles et al., 2015) and a set of downstream tasks.

PEER: Tasks

Three downstream tasks, each associated to a large(r) unlabeled dataset for self-supervised training (intrinsic evaluation) and a small(er) annotated dataset with labels for extrinsic evaluation.

End Task	Training Dataset (unlabeled)	Evaluation Dataset (labeled)
Edit-level article quality classification	WikiAtomicEdits (Faruqui et al., 2018a), WikiEditsMix	WikiEditsMix (4 edit-level quality labels)
MT post-edit type classification	QT21 En-De (Specia et al., 2017)	QT21 En-De MQM (6 post-edit type labels)
Grammar Error Correction difficulty classification	Lang 8 (Bryant et al., 2019)	WI + Locness (3 difficulty CEFR levels)

Results: Intrinsic Evaluation

Train. Data	Model	Valid BLEU	Valid GLEU	Test BLEU	Test GLEU
WikiAtomicSample	Guu	0.63	0.60	0.28	0.26
	Yin	0.81	0.79	0.81	0.79
	EVE	**0.84**	**0.82**	**0.84**	**0.82**
WikiEditsMix	Guu	0.56	0.53	0.54	0.52
	Yin	**0.65**	**0.65**	**0.65**	**0.65**
	EVE	0.58	0.61	0.55	0.57
Lang 8	Guu	0.53	0.43	0.51	0.41
	Yin	0.65	0.58	0.65	0.58
	EVE	**0.68**	**0.61**	**0.68**	**0.60**
QT21 De-En	Guu	0.47	0.37	0.32	0.30
	Yin	**0.57**	**0.49**	**0.57**	**0.49**
	EVE	0.53	0.45	0.54	0.46

Results: Extrinsic Evaluation

Train. Data	Model	Eval. Data	Accuracy Train	Accuracy Valid	Accuracy Test
WikiAtomicSample	Guu	WikiEditsMix	0.738	0.740	0.743
	Yin		0.671	0.672	0.668
	EVE		**0.782**	**0.780**	**0.774**
WikiEditsMix	Guu		**0.670**	**0.668**	**0.666**
	Yin		0.604	0.597	0.600
	EVE		0.637	0.642	0.638
Lang 8	Guu	WI + Locness	0.924	0.856	0.856
	Yin		0.836	0.831	0.831
	EVE		**0.971**	**0.958**	**0.958**
QT21 De-En	Guu	QT21 De-En MQM	0.925	0.896	0.933
	Yin		0.972	0.952	0.964
	EVE		**0.999**	**0.992**	**0.992**

Edit Aware Representation Learning via Levenshtein Prediction

with Machel Reid and Alfredo Solano

Introduction

- Most edit representation learning approaches are based on auto-encoding Yin et al. (2019); Marrese-Taylor et al. (2021), using “self-supervised learning“, others produced representations indirectly focusing on edit-centric downstream tasks (Sarkar et al., 2019; Marrese-Taylor et al., 2019)
- **Would a “neural Levenshtein algorithm” be conducive to improved downstream performance on edit-based tasks?**
  - We look at using the **Levenshtein algorithm as a form of supervision** to encourage a model to learn to convert a given input sequence into a desired output sequence

Levenshtein Prediction

- x_- original version of an object (a sequence of tokens).
- x_+ form after a change has been applied (also a sequence of tokens).

We tokenize (x_-, x_+), then use the Levenshtein algorithm to identify the text spans that have changed. Let $x_-^{i:j}$ be the sub-span on x_- that goes from positions i to j"

1. When a span has been inserted at $x_-^{i:j}$, such that it appears in $x_+^{k:j}$, we label the tokens in the latter as w^+, and also label token x_-^{i-1}, as $+$.
2. If $x_-^{i:j}$ has been replaced by the span $x_+^{k:l}$, we label the tokens on the respective spans as $\Leftrightarrow$ and $w^{\Leftrightarrow}$.
3. If the span $x_-^{i:j}$ has been removed from the sequence, we label each token as $-$.
4. Tokens that have not been involved in the edit are label with an empty tag, denoted as $=$.

Levenshtein Prediction (2)

As a result of our post-processing, each token in both x_- and x_+ is mapped to a single Levenshtein operation label: $\Leftrightarrow$, $w^{\Leftrightarrow}$, $+$ or w^+.

For example:

- **Input sequence (x_-):** "My name is John"
- **Output sequence (x_+):** "My last name is Wayne"

Becomes (using white-space tokenization):

[CLS]	My	name	is	John	[SEP]	My	last	name	is	Wayne
$=$	$+$	$=$	$=$	$\Leftrightarrow$	$=$	$=$	w^+	$=$	$=$	$w^{\Leftrightarrow}$

Thus, the end goal of our task is to predict these token-level Levenshtein operations relevant to transform x_- into x_+.

Data: Pre-training

We leverage large available corpora containing natural language edits:

- WikiEditsMix (Marrese-Taylor et al., 2021)
- WikiAtomicEdits (Faruqui et al., 2018b)

Dataset	Edits	Avg. Len
WikiAtomicEdits		
Insertions	13.7M	24.5
Deletions	9.3M	25.1
WikiEditsMix	114K	61.6

We use WikiEditsMix for ablation experiments regarding our proposed $\mathcal{L}_{x_\Delta}$ and $\mathcal{L}_{MLM}$ losses. To evaluate the pre-training phase, we utilize the **overall and per-token F1-score**.

Data: Downstream Tasks

- **Paraphrasing Detection**: we measure the ability of our edit encoder to model structure, context, and word order information, by means of using PAWS (Yang et al., 2019),
- **Edit-level Article Quality Estimation**: multi-class classification to predict the quality labels on WikiEditsMix (Marrese-Taylor et al., 2021). Concretely, the task is edit-level quality prediction with 4 labels: *spam, vandalism, attack OK*, each corresponding to a different quality of the edit.
- **Classification of Grammatical Errors**: since grammatical errors consist of many different types, we follow previous work (Marrese-Taylor et al., 2021) and use the GEC difficulty level annotations in the WI + Locness (Bryant et al., 2019) dataset.

We use **accuracy** for PAWS, and **F1-score** for the other datasets. Zero-shot and fine-tuning settings.

Baselines

- Encoder proposed by Yin et al. (2019), but we omit the copy mechanism proposed in the paper in order to make our results comparable.
- EVE (Marrese-Taylor et al., 2021), which also uses an auto-encoding loss for training, but does so in variational inference framework.
- The approach by Guu et al. (2018), but skip their sampling procedure.
- RoBERTa-base (Liu et al., 2019), as our task requires the model to capture structure, context, and word order information, we initialize our model with the RoBERTa-base weights, which we also adopt as a baseline for downstream experiments.

Results: Ablation Experiments

Results of our ablation experiments on WikiEditsMix:

Model	WikiEditsMix (F1-score)						PAWS		WikiEdits		GEC	
	$+$	w^{+}	$\Leftrightarrow$	$w^{\Leftrightarrow}$	$-$	All	ZS	Ft	ZS	Ft	ZS	Ft
$\mathcal{L}_{lev}$	**89.4**	**96.1**	90.6	88.6	93.7	**91.8**	56.8	94.9	56.8	78.1	**49.5**	52.4
$\mathcal{L}_{lev} + \mathcal{L}_{x_\Delta}$	87.8	95.6	89.9	**88.7**	93.5	91.2	**63.8**	94.9	56.7	78.2	48.6	**53.4**
$\mathcal{L}_{lev} + \mathcal{L}_{MLM}$	80.0	94.7	**93.8**	86.3	**95.6**	90.2	60.7	**95.0**	**64.8**	**78.4**	48.8	53.1

Comparison with state-of-the-art

Model		PAWS	WikiEditsMix	GEC
Zero-shot	RoBERTa	58.1	63.2	**50.7**
	EARL_{Mix}	**63.8**	56.7	48.6
	$\text{EARL}_{\text{Ins+Del}}$	62.2	**57.0**	47.6
Fine-tuning	RoBERTa	94.5	**78.9**	54.0
	Guu (2018)	-	74.3	85.6
	Yin (2019)	-	66.8	83.1
	EVE (2021)	-	77.4	**95.8**
	EARL_{Mix}	**94.9**	78.2	53.4
	$\text{EARL}_{\text{Ins+Del}}$	94.5	78.3	54.5

EARL_{Mix} and $\text{EARL}_{\text{Ins+Del}}$ indicate models that have been pre-trained on WikiEditsMix and WikiAtomicEdits (Insertions+Deletions), respectively

Conclusions

- Results on GEC poor because of domain of pre-training is too different to our data, which comes from Wikipedia. Pre-training on a GEC dataset should help (Marrese-Taylor et al., 2021).
- $\mathcal{L}_{MLM}$ generally helps the models attain better performance on the downstream, and that $\mathcal{L}_{x_\Delta}$ sometimes helps as well, specially for PAWS
- Our continued-training loss does not make the model forget the original pre-training, keeping performance on MNLI
- Impact of more data does not seem that important (results on INSERTIONS vs WIKIEDITSMIX are similar.
- Results could be due to pre-training/fine-tuning domain similarity rather than due to the effectiveness of $\mathcal{L}_{lev}$.

References

Jorge Balazs, Edison Marrese-Taylor, and Yutaka Matsuo. 2018. IIIDYT at IEST 2018: Implicit Emotion Classification With Deep Contextualized Word Representations. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Association for Computational Linguistics, Brussels, Belgium, pages 50–56.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 Shared Task on Grammatical Error Correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, Florence, Italy, pages 52–75. https://doi.org/10.18653/v1/W19-4406.

Manaal Faruqui, Ellie Pavlick, Ian Tenney, and Dipanjan Das. 2018a. WikiAtomicEdits: A Multilingual Corpus of Wikipedia Edits for Modeling Language and Discourse. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, pages 305–315. http://aclweb.org/anthology/D18-1028.

Manaal Faruqui, Ellie Pavlick, Ian Tenney, and Dipanjan Das. 2018b. WikiAtomicEdits: A multilingual corpus of Wikipedia edits for modeling language and discourse. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, pages 305–315. https://doi.org/10.18653/v1/D18-1028.

Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating Sentences by Editing Prototypes. *Transactions of the Association for Computational Linguistics* 6:437–450. https://doi.org/10.1162/tacl$_a0$0030.

Suzana Ilić, Edison Marrese-Taylor, Jorge Balazs, and Yutaka Matsuo. 2018. Deep contextualized word representations for detecting sarcasm and irony. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Association for Computational Linguistics, Brussels, Belgium, pages 2–7. http://aclweb.org/anthology/W18-6202.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]* http://arxiv.org/abs/1907.11692.

Pablo Loyola, Edison Marrese-Taylor, Jorge Balazs, Yutaka Matsuo, and Fumiko Satoh. 2018. Content Aware Source Code Change Description Generation. In *Proceedings of the 11th International Conference on Natural Language Generation*. Association for Computational Linguistics, Tilburg University, The Netherlands, pages 119–128.

Pablo Loyola, Edison Marrese-Taylor, and Yutaka Matsuo. 2017. A Neural Architecture for Generating Natural Language Descriptions from Source Code Changes. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 287–292. https://doi.org/10.18653/v1/P17-2045.

Edison Marrese-Taylor, Pablo Loyola, and Yutaka Matsuo. 2019. An Edit-centric Approach for Wikipedia Article Quality Assessment. In *Proceedings of the 5th Workshop on Noisy User-Generated Text (W-NUT 2019)*. Association for Computational Linguistics, Hong Kong, China, pages 381–386. https://doi.org/10.18653/v1/D19-5550.

Edison Marrese-Taylor and Yutaka Matsuo. 2017. EmoAtt at EmoInt-2017: Inner attention sentence embedding for Emotion Intensity. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Association for Computational Linguistics, Copenhagen, Denmark, pages 233–237.

Edison Marrese-Taylor, Machel Reid, and Yutaka Matsuo. 2020. Variational Inference for Learning Representations of Natural Language Edits. *arXiv:2004.09143 [cs]* .

Edison Marrese-Taylor, Machel Reid, and Yutaka Matsuo. 2021. Variational Inference for Learning Representations of Natural Language Edits. *Proceedings of the AAAI Conference on Artificial Intelligence* 35(15):13552–13560. https://ojs.aaai.org/index.php/AAAI/article/view/17598.

Anders Miltner, Sumit Gulwani, Vu Le, Alan Leung, Arjun Radhakrishna, Gustavo Soares, Ashish Tiwari, and Abhishek Udupa. 2019. On the fly synthesis of edit suggestions. *Proceedings of the ACM on Programming Languages* 3(OOPSLA):143:1–143:29. https://doi.org/10.1145/3360569.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. 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 (Volume 2: Short Papers)*. Association for Computational Linguistics, Beijing, China, pages 588–593. https://doi.org/10.3115/v1/P15-2097.

Soumya Sarkar, Bhanu Prakash Reddy, Sandipan Sikdar, and Animesh Mukherjee. 2019. StRE: Self Attentive Edit Quality Prediction in Wikipedia. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, pages 3962–3972. https://doi.org/10.18653/v1/P19-1387.

L. Specia, K. Harris, A. Burchardt, M. Turchi, M. Negri, and I. Skadina. 2017. Translation Quality and Productivity: A Study on Rich Morphology Languages. pages 55–71. https://cris.fbk.eu/handle/11582/313118.XiFXEeGRVGM.

Morten Warncke-Wang, Dan Cosley, and John Riedl. 2013. Tell Me More: An Actionable Quality Model for Wikipedia. In *Proceedings of the 9th International Symposium on Open Collaboration*. ACM, New York, NY, USA, WikiSym '13, pages 8:1–8:10. https://doi.org/10.1145/2491055.2491063.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, pages 3687–3692. https://doi.org/10.18653/v1/D19-1382.

Pengcheng Yin, Graham Neubig, Miltiadis Allamanis, Marc Brockschmidt, and Alexander L. Gaunt. 2019. Learning to Represent Edits. In *Proceedings of the 7th International Conference on Learning Representations*. https://openreview.net/forum?id=BJl6AjC5F7.