Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 5517–5524
Marseille, 20-25 June 2022

One Document, Many Revisions: A Dataset for Classification and Description of Edit Intents

Dheeraj Rajagopal[*], **Xuchao Zhang**[♣], **Michael Gamon**[†],
Sujay Kumar Jauhar[†], **Diyi Yang**[♠], **Eduard Hovy**[*]
[*]Carnegie Mellon University [†]Microsoft Research
[♠]Georgia Institute of Technology [♣]NEC Labs America
{dheeraj, hovy}@cs.cmu.edu
xuczhang@nec-labs.com
{mgamon,sjauhar}@microsoft.com
diyi.yang@cc.gatech.edu

Abstract

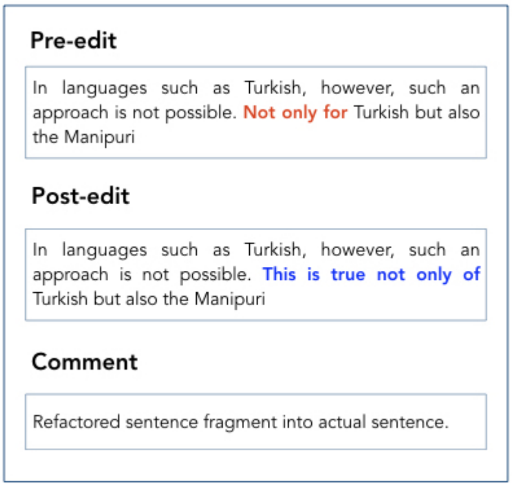
Document authoring involves a lengthy revision process, marked by individual edits that are frequently linked to comments. Modeling the relationship between edits and comments leads to a better understanding of document evolution, potentially benefiting applications such as content summarization, and task triaging. Prior work on understanding revisions has primarily focused on classifying edit intents, but falling short of a deeper understanding of the nature of these edits. In this paper, we present explore the challenge of describing an edit at two levels: identifying the edit intent, and describing the edit using free-form text. We begin by defining a taxonomy of general edit intents and introduce a new dataset of full revision histories of Wikipedia pages, annotated with each revision's edit intent. Using this dataset, we train a classifier that achieves a 90% accuracy in identifying edit intent. We use this classifier to train a distantly-supervised model that generates a high-level description of a revision in free-form text. Our experimental results show that incorporating edit intent information aids in generating better edit descriptions. We establish a set of baselines for the edit description task, achieving a best score of 28 ROUGE, thus demonstrating the effectiveness of our layered approach to edit understanding.

Keywords: Edit Description, Wiki Edits, Comment Generation

1. Introduction

A typical written document undergoes multiple revisions before reaching a final version. When multiple authors do these revisions, it can be difficult to track them and understand how the document evolved. A method to automatically identify the edit intent and the action would be of use. Further, such a method would support an eventual system that could automatically decide which sentences to edit and how to edit them.

Revisions have been shown to provide insights into understanding the relationship between authors and document evolution Yang et al. [2016], article trustworthiness Zeng et al. [2006], neutrality or bias point-of-view detection Recasens et al. [2013], and author participation Halfaker et al. [2013]. Wikipedia is a large-scale community-maintained resource that uses revisions and comments to build a successful encyclopedic resource. When an author edits a page for an *edit* or *revision*, she/he usually leaves a *comment*, which is free-form text description of the edit that typically characterizes it's intent. A typical edit example from Wikipedia is shown in figure 1. Prior work on revision-intents primarily develop classification systems based on edit intents Faigley and Witte [1981], Yang et al. [2016], Daxenberger and Gurevych [2012]. However, capturing the intent alone does not adequately characterize the nature of the edit. In this paper, we address the challenge of understanding edit-intents and its associated comments, by identifying the edit-intention, and generating the comment that described the change.


Pre-edit

In languages such as Turkish, however, such an approach is not possible. Not only for Turkish but also the Manipuri

Post-edit

In languages such as Turkish, however, such an approach is not possible. This is true not only of Turkish but also the Manipuri

Comment

Refactored sentence fragment into actual sentence.

Figure 1: Example of an edit in Wikipedia and its corresponding comment. The comment is a description of the intent or substance of an edit, left by its author.

At the first level, we classify the intent of the edit. Previous datasets of edit-intents like Daxenberger [2016] and Yang et al. [2017] have focused primarily on Wikipedia specific intent-categories. We begin by defining a taxonomy of 7 edit intents based on Yang et al. [2017] with the aim of generalizing edit intent categories to other genres of written documents. We present a publicly available dataset that contains the first 100 revisions of 147 wikipedia pages (with about 9300 datapoints in total), with its associated comment and rich meta-data, potentially used for tasks beyond edit understanding.

We use this dataset to train a classifier to predict the intent of an edit operation with about 90% accuracy. We use this classifier as distant supervision to collect a much larger dataset (10× the original data) of revisions with *fuzzy intent labels*. At the next level, we use this large-scale data for training a generation model, that attempts to describe the edit in free-form, given the pre-edit and post-edit version of a document. We show that these intent labels improve the quality of the edit description, and our best model achieves a score 28 ROUGE-L score, leaving immense room for further research in this area. To summarize,

(i) We define an edit-intent taxonomy, and present a labeled corpus with wikipedia edits over a sequence of contiguous revisions, labeled with their corresponding edit intentions.

(ii) We also present a novel *edit-description* task, that aims to describe an edit in free-form text. We show that our model that incorporates the intent learnt from a distantly supervised model learns to generate better edit descriptions.

2. Related Work

Revision Intent Classification : Faigley and Witte [1981] present the first set of edit intents and propose a taxonomy for edits. At the top level, the edits are broadly classified into either surface level meaning-preserving changes or text-based meaning-altering changes. Further work has attempted to unify these changes into fine-grained categories Bronner and Monz [2012]. Pfeil et al. [2006] focus on spelling, grammar and markup changes across multiple languages with a limited dataset of only 500 revisions. Jones [2008] analyze the differences in revisions of multiple authors in collaborative writing. Daxenberger and Gurevych [2012] study the intent of individual sequence changes rather than revisions that tend to have more than one change, with the intent categories limited to syntactic changes to text. Our taxonomy and motivation aligns closely with Yang et al. [2017].

Intent Classification for Downstream Applications : Edit Intent classification has been studied from the perspective of many downstream applications where intents are defined on the basis of specific tasks including sentence compression Yamangil and Nelken [2008], grammatical error correction Yamangil and Nelken [2008], extracting entailment rules Cabrio et al. [2012], studying vandalism Chin et al. [2010], understanding article quality Kittur and Kraut [2008] and studying argumentative writing Zhang et al. [2017].

Generation for Summarization : Abstractive document summarization systems use sequence-to-sequence models to generate abstractive summaries of text Rush et al. [2015], Chopra et al. [2016], See et al. [2017]. Multi-pass extractive-abstractive summarization systems first select content and then to build a summary Nallapati et al. [2016], Cohan et al. [2018], Pasunuru and Bansal [2018]. Our modeling decisions for the difference attention module also follows content selection and summarization steps, but varies in the fact that we only generate the description of an edit, compared to summarization which warrants modeling more complex interactions. To the best of our knowledge, there is no prior work on generating the description the edit between a pre-edit and post-edit document in free-form text.

3. Dataset

Our edit-intent categories are adapted from Yang et al. [2017], by collapsing similar categories and collecting all Wikipedia-specific edits to one category called *Other*. Although an edit might be composed of multiple edit intentions, we focus only on the primary intention, to facilitate modeling simplicity. Our edit-intent categories and description are shown in table 1.

Using this as our taxonomy, we construct a dataset of Wikipedia revision histories based on a Wikipedia data dump from October 2018[1]. From this dump we randomly sample 147 unique Wikipedia pages and pick the first N_{rev} revisions (in this paper, we set $N_{rev} = 100$). For preprocessing, we filter out comments that were empty and comments that mention only names of authors, leading us to a total of about 9300 datapoints. We pick consecutive revisions so that we can use the same dataset as a resource to study other aspects of document evolution. While we do not ourselves leverage the sequential nature of revisions in this paper, we are making the dataset freely available to the research community at `https://tinyurl.com/editsumm`. The *comment* field associated with each revision is a free-form description of the edit. We use crowdsourcing to collect intent labels for all revisions that has atleast two tokens in the comment field.

For the crowd-sourced annotation task[2], the goal of each HIT is to assign a label to a specific pre-edit and post-edit pair. To this end we ask Turkers to select the most appropriate class for a Wikipedia revision by presenting them with the following information: the text that underwent change in the revision and some surrounding context from the page, the associated comment left by the editor, Table 1 for reference of classes and their explanations, and the link to the full Wikipedia diff page[3] for additional context.

Each sample was annotated by 3 independent annotators, and samples with a majority class agreement were

[1] https://dumps.wikimedia.org/wikidatawiki/20181001/

[2] using the amazon MTurk platform

[3] example: https://goo.gl/6zGmUk

Class	Explanation
Add Supporting Evidence	Adding, removing or replacing supporting evidence (links or citations)
Fact Update	Updating facts in document (numbers, dates, etc.)
Point-of-view Change	Removing bias, rewriting using neutral tone
Add New Information	Adding new textual content to page, image, info-box or table
Remove Existing Information	Removing irrelevant or redundant information
Word-Smithing	Rephrasing or rearranging text, improving grammar, spelling, and punctuation
Other	Wikipedia specific changes like Wiki-formatting, and comments to other editors

Table 1: Classes of Edit Intents and their Explanations

considered a valid data sample (examples shown in table 4). Each turker was paid \$0.06 for an annotation. Out of approximately 12000 samples for annotation, we got a majority class for about 77 % of the samples (in the rest of the samples, all three annotators picked unique class labels), giving us a total of $\sim$ 9300 *(pre-edit, post-edit, comment, edit-intent)* quadruples. This is about twice the size of previous dataset of edit intent dataset Yang et al. [2017]. Figure 2 shows a screenshot of the turker interface for annotation of the intent class.

Our data had an inter-annotator agreement of $\alpha = 0.49$ krippendorf's alpha score Krippendorff [2011]. We observe that it took 30 seconds on average to complete one annotation task. Each selected turker had a minimum of 1000 HITs with atleast 95% success ratio. The label distribution from the collected data is shown in table 2.

Class	Percentage(%)
Provide Supporting Evidence	45.9
Word-Smithing	19.0
Add New Information	23.2
Fact Update	3.7
Point-of-view Change	0.7
Remove Existing Information	4.7
Other	2.6

Table 2: Distribution of edit intents. **The top 3 classes account for 88% of overall edits**. True to the nature of wikipedia, establishing evidence for statements was the primary reason for an edit in the document.

We additionally store a number of meta-data fields (shown in table 3) associated with each revision. Together with our annotations they comprise a rich dataset of the sequential evolution of Wikipedia pages, capable of supporting many important modeling challenges, such as author contribution over time, and document evolution from the perspective of edit-intention.

Table 4 shows example data samples from the dataset [4]

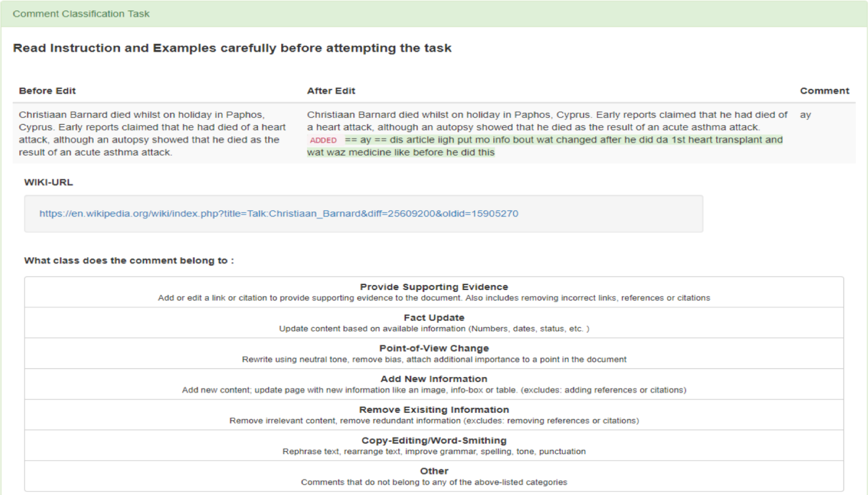
Field	Description
Title	Title of the page
Bef_Rev	Content of page before revision
Aft_Rev	Content of page after revision
Comment	Author's comment
Edit Intent	**Intent of the Edit (crowd-sourced)**
Time Stamp	Time information of revision
Author	Author ID
URL	URL corresponding to the revision
Mod_Bef	Edited block before revision
Mod_Aft	Edited block after revision
Minor	Major/Minor revision

Table 3: Dataset fields for each revision

4. Problem Formulation

Edit Representation: An atomic edit Faruqui et al. [2018] represents a single change in a document. In our work, we model each *edit* as a set of atomic edits that are made by author during one revision. Given a pre-edit $D_s = \{ws_j\}_{j=1}^{L_s}$ and post-edit $D_t = \{wt_j\}_{j=1}^{L_t}$ of a document, where ws_j, wt_j, L_s, L_t denote word in pre-edit, word in post-edit, number of words in pre-edit and post-edit respectively. We extract a representation of this edit pair es (source) and et (target), based on individual atomic edits in the sequence D_s and D_j. For every atomic edit (insertion, deletion), we extract a context window using a hyperparameter w_s, which represents the number of words extracted before and after an atomic edit as context. Each input sample consists of a pre-edit (source) and post-edit (target) pair $x_i = (es_i, et_i)$. Both es_i and et_i are sequences of words respectively: $es_i = \{s_j\}_{j=1}^{n_s}$ and $et_i = \{t_j\}_{j=1}^{n_t}$.

[4] The entire dataset is available in the supplementary material



Comment Classification Task

Read Instruction and Examples carefully before attempting the task

Before Edit	After Edit	Comment
Christiaan Barnard died whilst on holiday in Paphos, Cyprus. Early reports claimed that he had died of a heart attack, although an autopsy showed that he died as the result of an acute asthma attack.	Christiaan Barnard died whilst on holiday in Paphos, Cyprus. Early reports claimed that he had died of a heart attack, although an autopsy showed that he died as the result of an acute asthma attack. ADDED == ay == dis article iigh put mo info bout wat changed after he did da 1st heart transplant and wat waz medicine like before he did this	ay

WIKI-URL

https://en.wikipedia.org/wiki/index.php?title=Talk:Christiaan_Barnard&diff=25609200&oldid=15905270

What class does the comment belong to :

Provide Supporting Evidence
Add or edit a link or citation to provide supporting evidence to the document. Also includes removing incorrect links, references or citations

Fact Update
Update content based on available information (Numbers, dates, status, etc.)

Point-of-View Change
Rewrite using neutral tone, remove bias, attach additional importance to a point in the document

Add New Information
Add new content; update page with new information like an image, info-box or table. (excludes: adding references or citations)

Remove Exisiting Information
Remove irrelevant content, remove redundant information (excludes: removing references or citations)

Copy-Editing/Word-Smithing
Rephrase text, rearrange text, improve grammar, spelling, tone, punctuation

Other
Comments that do not belong to any of the above-listed categories

Figure 2: Mturk Interface for data annotation

Pre-Edit	Post-Edit	Intent Class
Banda Neira, or Naira, the island with the administrative capital and a small airfield (as well as **accomodation** for visitors).	Banda Neira, or Naira, the island with the administrative capital and a small airfield (as well as **accommodation** for visitors).	Word-Smithing
At that stage B should probably have died, but it continued to see use as late as the 1990s on Honeywell mainframes, and on certain [[embedded systems]], **mostly because these,poor deprived (and depraved) systems did not have anything better.**	**B continued to see** use as late as the 1990s on Honeywell mainframes, and on certain [[embedded systems]] **for a variety of reasons, including limited hardware in the small systems and extensive libraries, tools, licensing cost issues and simply being good enough for the job on others.**	Point-of-View Change

Table 4: Examples from the dataset with three fields - Pre-Edit, Post-Edit and Edit-Intent.

Here s_j denotes a source word and t_j denotes a target word and n_s and n_t are the number of words in source and target sequences respectively.

Edit-Intent Classification: In this task we attempt to classify the intent of a revision into one of the 7 classes in Table 1. More formally, given n independent samples of pre- and post-edit pairs $X = \{x_1, x_2, ...x_n\}$ and their corresponding training labels $p = \{p_1, p_2, ...p_n\}$, we attempt to learn a classifier F_i, that maximizes the likelihood of predicting a correct intent label. Here $p_i \in \{I_k\}_{k=1}^{N_p}$, where I_k denotes the edit-intent label and N_p is number of edit-intent labels.

Distant Supervision: Neural language models are typically trained on large volume of data Devlin et al. [2018], Peters et al. [2018]. We formulate generating the description of an edit as a conditional language modeling task, where the generated tokens are conditioned on the linguistic nature and intent of the edit. Since large-scale intent annotation efforts are expensive, we extract (pre-edit, post-edit, comment) triples from Wikipedia and annotate them with a fuzzy intent label using a distant supervision Mintz et al. [2009] approach. Formally, given small set of labeled data (X, y) and a classifier C, the goal is to generate large scale *weakly labeled* data $(\hat{X}, \hat{y})$. We use our edit-intent classifier F_i to generate these intent labels for $\hat{X}$.

Edit-Comment Generation: In this task, we attempt to generate a free-form text summary of the intent of an edit. Our training data $Z = \{z_1, z_2, ..z_m\}$ consists of individual samples $z_i = (es_i, et_i, \hat{p}_i)$, where es_i and et_i are the source and target sequences respectively, and $\hat{p}_i$ is the fuzzy intent class over the input pair (es_i, et_i). Each associated edit description (author comment), is represented as a sequence $C_i = \{c_j\}_{j=1}^{n_c}$, where n_c represents the number of words in C_i. The goal of the task is to build a model that is capable of generating an output sequence $\hat{C} = P(c_j | es_i, et_i, \hat{p}_i)$ given the unseen input z_{test}, at test time.

5. Model

The model section describes the model for edit-predicate classification and edit-comment generation. We use a hierarchical model with multiple layers - comprising of an input layer, contextual embedding layer, global difference attention layer, local attention layer, and an output layer. The architecture of the model is shown in figure 3.

Input Layer : In the input layer, we map words of both the source and target to dense embeddings. Given embedding dimension d, the input layer outputs two

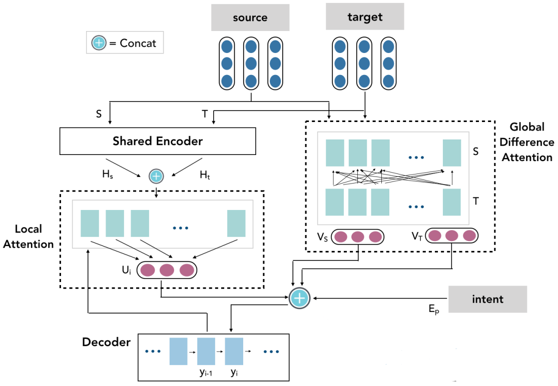


Figure 3: Overall architecture of the model that generates the edit descriptions (Inputs are displayed in gray boxes)

matrices, a representation for source $S \in \mathbb{R}^{d \times n_s}$ and target $T \in \mathbb{R}^{d \times n_t}$. To incorporate the fuzzy intent $\hat{p}_i$, we project the intent to a d_p dimensional embedding E_p, where $E_p \in \mathbb{R}^{d_p}$.

Contextual Embedding Layer : We use a bidirectional Long Short-Term Memory Network (BiLSTM) encoder Hochreiter and Schmidhuber [1997] over the sequence of word embeddings to encode both source and target. This layer computes a source encoding BiLSTM$(S) = H_s \in \mathbb{R}^{2d \times n_s}$ and a target encoding BiLSTM$(T) = H_t \in \mathbb{R}^{2d \times n_t}$, where H_s and H_t denote the concatenation of hidden representations of source and target respectively and d the dimension of hidden layer. It is worth noting that the parameters are shared while computing H_s and H_t.

Global Difference Attention : In this layer, we find an alignment between subphrases of source and target embeddings S and T. We use decomposable attention Parikh et al. [2016] as our global difference attention function. Let $S = (a_1, a_2, ..., a_{l_a})$ and $T = (b_1, b_2, ..., b_{l_b})$, where each $a_i, b_j \in \mathbb{R}^d$. The unnormalized attention weights are given by $e_{ij} = F(a_i, b_j)$, where F is a feed-forward network with ReLU activations. The attention weights are then normalized using

$$\beta_j = \sum_{j=1}^{l_b} \frac{exp(e_{ij})}{\sum_{k=1}^{l_b} exp(e_{ik})} b_j$$
$$\alpha_j = \sum_{i=1}^{l_a} \frac{exp(e_{ij})}{\sum_{k=1}^{l_a} exp(e_{kj})} a_i$$

We then find

$$v_{1,i} = G([a_i, \beta_i]) \quad \forall i \in [1, ..., l_a]$$
$$v_{2,j} = G([b_j, \alpha_j]) \quad \forall j \in [1, ..., l_b]$$

where G is the dot-product similarity. We then find the final aggregated representations $v_s = \sum_{i=1}^{l_a} v_{1,i}$ and $v_t = \sum_{j=1}^{l_b} v_{2,j}$

Local Attention : In the local attention module, we model the interactions between source and target encoded representations and the output directly. The local attention module uses the attention proposed by Bahdanau et al. [2014], where we compute a context vector for each decoder step u_i, where $u_i = \sum_{i=1}^{T_d} \alpha_{ij} h_j$, the local attention weight is computed by

$$\alpha_{ij} = \frac{e_{ij}}{\sum_{k=1}^{T_d} e_{ik}}$$

h_i represents a hidden layer in $[H_s; H_t]$, concatenation of H_s and H_t.

Output Layer : The aggregated vector from all the modules is given by $O_i = [u_i, v_s, v_j, E_p]$, where $[,]$ denotes concatenation. To generate the comment, we define the training problem as a conditional model, $p(y_{i+1}|z, y_c; \theta)$. We estimate the model parameters θ to minimize the negative log-likelihood.

$$NLL(\theta) = -\sum_{n=1}^{N} log\ p(y^{(j)}|z^{(j)}; \theta),$$
$$= -\sum_{n=1}^{N} log\ p(y_{i+1}^{(j)}|\ O_i)$$

To generate the free-form edit summaries, we use the Viterbi algorithm with greedy decoding to find the optimal $y^* = \text{argmax}_{y \in Y} \sum_{i=0}^{N-1} g(O_i)$, where $g(\cdot)$ indicates softmax$(W_g * O_i)$, where $W_g \in \mathbb{R}^{d_o \times V}$ where V denotes the decoder vocabulary size [5].

6. Experiments

6.1. Experimental Setup

In all of our models, the word embeddings are initialized using 300 dimensional GloVE vectors Pennington et al. [2014]. We use a single TitanX(Pascal) GPU, setting the batch size to 16. We optimize model parameters on training, using the Adam optimizer Kingma and

[5] the code for training this model is available in the supplementary material

Ba [2014] with a learning rate of 0.0004. We evaluate both edit-intent and action-predicate classification on accuracy. In the case of comment generation, we measure on ROUGE metric.

Edit-Intent Classification : We use the data described in 3 as the basis of our experiments. We use a train-test split of 80:20 with 10-fold cross validation.

Edit-Comment Generation : We use the same input from the above dataset of about 90000 *(pre-edit, post-edit, intent, comment)* quadruples with a 80:10:10 train:validation:test split. In this task, our goal is to generate the comment sequence C. We perform our experiments for different context window sizes.

6.2. Baselines

Since our dataset is new, we do not have any previous state-of-the-art methods to directly compare against. So, we resort to baseline classifiers that are suitably defined for each of the tasks, with their corresponding complexity in mind. For edit-intent classification, our neural-network based baselines include a MLP Classifier and the decomposable attention model proposed by Parikh et al. [2016], with the final softmax adapted to predict the intent of an edit. We also compare the performances of several non-neural network based approaches, since we anticipate that neural network models are prone to overfitting in small-scale data. Since the comment generation task does not have a prior established baseline, we resort to doing a detailed ablation-based model study. The baseline is a sequence-to-sequence BiLSTM network without attention, similar to the baseline model for action predicate classification.

7. Results and Discussion

Edit Intent Classification : Our edit intent classification results are presented in Table 5 and the classwise accuracies for teh best performing model is shown in Table 6.

Classifier	Accuracy
Random Baseline	0.14
Majority Class	0.46
KNN*	0.58 ± 0.32
SVM (linear)*	0.58 ± 0.07
Logistic Regression*	**0.89 ± 0.14**

Table 5: Edit intent classification Results, with their corresponding 95% confidence interval estimates with 10-fold cross validation. * - denotes that these classifiers use BoW features

We also show class-wise precision, recall and F1 scores for the best performing classifier (Logistic Regression) in the test set to emphasize that the classifier was not biased towards predicting only majority classes.

Edit Comment Generation :

Class	Precision	Recall	F1-score
Provide Evidence	0.90	0.94	0.92
Fact Update	0.92	0.93	0.92
Point of View Change	0.80	0.55	0.65
Add New Information	0.88	0.88	0.88
Remove Information	0.88	0.79	0.83
Word-Smithing	0.85	0.81	0.83
Other	0.85	0.88	0.86

Table 6: Class-wise Precision, Recall , F1-score for Logistic Regression

Following the evaluation metrics from the summarization literature, all our models are evaluated using the ROUGE metric. We report ROUGE-1, ROUGE-2 and ROUGE-L scores. The results for comment generation are shown in Table 7. We follow a baseline setup with ablation analysis for our experiments. Although noisily generated by distant supervision, we notice that the adding intent information provides a significant improvement to the generation results. As stand-alone modules, predicted intent had a significant effect on the generation scores. And overall, global edit attention improves the scores by about 14 points ROUGE.

Pretrained Language Model performance :

To understand pretrained language model performance on this task, we use the transformer based BART Lewis et al. [2020] sequence to sequence model. The results of the BART model is shown in table 8.

It is important to note that BART was pretrained on written text over the C4 commoncrawl corpus[6]. And adapting it to wikipedia comment generation task did not directly lead to performance gains. Our experiments show that pretrained BART does not naturally adapt for our task, opening up exciting research avenues to improved adaptation of pretrained models in real-world text settings[7].

8. Conclusion

In this paper, we present a novel dataset that helps study the task of describing an edit by identifying the intent, and using it to generate an edit description. We leverage on this dataset, with which we build a classifier to identify the edit intent. We also show that a distantly-trained system using fuzzy labels shows gains for the edit description task. We also show the results for several baselines for the edit description task by modeling the interaction between pre-edit and post-edit texts.

[6] https://commoncrawl.org

[7] the code for training BART model is available in the supplementary material

Model	ROUGE-1	ROUGE-2	ROUGE-L
Seq2Seq + LA	13.0	5.6	12.5
Seq2Seq + LA + PI	14.5	6.6	14.1
Seq2Seq + LA + PI + GA	**28.0**	**17.0**	**27.9**

Table 7: Edit summary generation results. All ROUGE metrics reported using pyrouge package. LA=local attention, PI=predicted intent, GA=global difference attention.

Model	ROUGE-1	ROUGE-2	ROUGE-L
BART	10.4	6.1	9.7

Table 8: Edit summary generation ROUGE scores for BART.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Amit Bronner and Christof Monz. User edits classification using document revision histories. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 356–366. Association for Computational Linguistics, 2012.

Elena Cabrio, Bernardo Magnini, and Angelina Ivanova. Extracting context-rich entailment rules from wikipedia revision history. In *Proceedings of the 3rd Workshop on the People's Web Meets NLP: Collaboratively Constructed Semantic Resources and their Applications to NLP*, pages 34–43. Association for Computational Linguistics, 2012.

Si-Chi Chin, W Nick Street, Padmini Srinivasan, and David Eichmann. Detecting wikipedia vandalism with active learning and statistical language models. In *Proceedings of the 4th workshop on Information credibility*, pages 3–10. ACM, 2010.

Sumit Chopra, Michael Auli, and Alexander M Rush. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, 2016.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685*, 2018.

Johannes Daxenberger. *The Writing Process in Online Mass Collaboration: NLP-Supported Approaches to Analyzing Collaborative Revision and User Interaction*. PhD thesis, Technische Universität, 2016.

Johannes Daxenberger and Iryna Gurevych. A corpus-based study of edit categories in featured and non-featured wikipedia articles. *Proceedings of COLING 2012*, pages 711–726, 2012.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Lester Faigley and Stephen Witte. Analyzing revision. *College composition and communication*, 32 (4):400–414, 1981.

Manaal Faruqui, Ellie Pavlick, Ian Tenney, and Dipanjan Das. WikiAtomicEdits: A Multilingual Corpus of Wikipedia Edits for Modeling Language and Discourse. In *Proc. of EMNLP*, 2018.

Aaron Halfaker, R Stuart Geiger, Jonathan T Morgan, and John Riedl. The rise and decline of an open collaboration system: How wikipedia's reaction to popularity is causing its decline. *American Behavioral Scientist*, 57(5):664–688, 2013.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

John Jones. Patterns of revision in online writing: A study of wikipedia's featured articles. *Written Communication*, 25(2):262–289, 2008.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Aniket Kittur and Robert E Kraut. Harnessing the wisdom of crowds in wikipedia: quality through coordination. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 37–46. ACM, 2008.

Klaus Krippendorff. Computing krippendorff's alpha-reliability. 2011.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th*

Annual Meeting of the Association for Computational Linguistics, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://www.aclweb.org/anthology/2020.acl-main.703.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics, 2009.

Ramesh Nallapati, Bowen Zhou, and Mingbo Ma. Classify or select: Neural architectures for extractive document summarization. *arXiv preprint arXiv:1611.04244*, 2016.

Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*, 2016.

Ramakanth Pasunuru and Mohit Bansal. Multi-reward reinforced summarization with saliency and entailment. *arXiv preprint arXiv:1804.06451*, 2018.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

Ulrike Pfeil, Panayiotis Zaphiris, and Chee Siang Ang. Cultural differences in collaborative authoring of wikipedia. *Journal of Computer-Mediated Communication*, 12(1):88–113, 2006.

Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. Linguistic models for analyzing and detecting biased language. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1650–1659, 2013.

Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*, 2015.

Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*, 2017.

Elif Yamangil and Rani Nelken. Mining wikipedia revision histories for improving sentence compression. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 137–140. Association for Computational Linguistics, 2008.

Diyi Yang, Aaron Halfaker, Robert E Kraut, and Eduard H Hovy. Who did what: Editor role identification in wikipedia. In *ICWSM*, pages 446–455, 2016.

Diyi Yang, Aaron Halfaker, Robert Kraut, and Eduard Hovy. Identifying semantic edit intentions from revisions in wikipedia. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2000–2010, 2017.

Honglei Zeng, Maher A Alhossaini, Li Ding, Richard Fikes, and Deborah L McGuinness. Computing trust from revision history. Technical report, Stanford Univ Ca Knowledge Systems LAB, 2006.

Fan Zhang, Homa B Hashemi, Rebecca Hwa, and Diane Litman. A corpus of annotated revisions for studying argumentative writing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1568–1578, 2017.