Learning To Split and Rephrase From Wikipedia Edit History

Jan A. Botha[*]   **Manaal Faruqui**[*]   **John Alex**   **Jason Baldridge**   **Dipanjan Das**

`{jabot,mfaruqui,jpalex,jasonbaldridge,dipanjand}@google.com`

Google AI Language

Abstract

Split and rephrase is the task of breaking down a sentence into shorter ones that together convey the same meaning. We extract a rich new dataset for this task by mining Wikipedia's edit history: WikiSplit contains one million naturally occurring sentence rewrites, providing sixty times more distinct split examples and a ninety times larger vocabulary than the WebSplit corpus introduced by Narayan et al. (2017) as a benchmark for this task. Incorporating WikiSplit as training data produces a model with qualitatively better predictions that score 32 BLEU points above the prior best result on the WebSplit benchmark.

1 Introduction

A complex sentence can typically be rewritten into multiple simpler ones that together retain the same meaning. Performing this *split-and-rephrase* task is one of the main operations in text simplification, alongside paraphrasing and dropping less salient content (Siddharthan, 2006; Zhu et al., 2010; Woodsend and Lapata, 2011, i.a.). The area of automatic text simplification has received a lot of attention (Siddharthan, 2014; Shardlow, 2014), yet still holds many open challenges (Xu et al., 2015). Splitting sentences in this way could also benefit systems where predictive quality degrades with sentence length, as observed in, e.g., relation extraction (Zhang et al., 2017) and translation (Koehn and Knowles, 2017). And the schema-free nature of the task may allow for future supervision in the form of crowd-sourced rather than expensive expert annotation (He et al., 2015).

Narayan et al. (2017) introduce the WebSplit corpus for the split-and-rephrase task and report results for several models on it. Aharoni and Goldberg (2018) improve WebSplit by reducing overlap in the data splits, and demonstrate that neural

---

[*]Both authors contributed equally.

A classic leaf symptom is water-soaked lesions between the veins which appear as angular leaf-spots where the lesion edge and vein meet.

A classic leaf symptom is the appearance of angular, water-soaked lesions between the veins. The angular appearance results where the lesion edge and vein meet.

Figure 1: A split-and-rephrase example extracted from a Wikipedia edit, where the top sentence had been edited into two new sentences by removing some words (yellow) and adding others (blue).

encoder-decoder models (Bahdanau et al., 2014) perform poorly, even when enhanced with a copy mechanism (Gu et al., 2016; See et al., 2017).

One limitation of the WebSplit examples themselves is that they contain fairly unnatural linguistic expression using a small vocabulary. We introduce new training data mined from Wikipedia edit histories that have some noise, but which have a rich and varied vocabulary over naturally expressed sentences and their extracted splits. Figure 1 gives an example of how a Wikipedia editor rewrote a single sentence into two simpler ones. We create WikiSplit, a set of one million such examples mined from English Wikipedia, and show that models trained with this resource produce dramatically better output for split and rephrase.

Our primary contributions are:

- A scalable, language agnostic method for extracting split-and-rephrase rewrites from Wikipedia edits.
- Public release of the English WikiSplit dataset, containing one million rewrites: `http://goo.gl/language/wiki-split`
- By incorporating WikiSplit into training, we more than double (30.5 to 62.4) the BLEU score obtained on WebSplit by Aharoni and Goldberg (2018).

Correct

Street Rod is the first in a series of two games released for the PC and Commodore 64 in 1989.   Street Rod is the first in a series of two games. It was released for the PC and Commodore 64 in 1989.
He played all 60 minutes in the game and rushed for 114 yards, more yardage than all the Four Horsemen combined.   He played all 60 minutes in the game. He rushed for 114 yards, more yardage than all the Four Horsemen combined.

Unsupported

When the police see Torco's injuries, they send Ace to a clinic to be euthanized, but he escapes and the clinic worker covers up his incompetence.   When the police see Torco's injuries to his neck, they believe it is a result of Ace biting him. They send Ace to a clinic to be euthanized, but he escapes and the clinic worker covers up his incompetence.

Missing

The avenue was extended to Gyldenløvesgade by Copenhagen Municipality in 1927-28 and its name was changed to Rosenørns Allé after Ernst Emil Rosenørn (1810-1894).   The avenue was extended to Gyldenløvesgade by Copenhagen Municipality in 1927-28. The street was named after Ernst Emil Rosenørn (1810-1894).

Table 1: Examples of correct and noisy sentence splits extracted from Wikipedia edits. Noise from unsupported or missing statements is visualized with the same coloring as in Figure 1.

2 The WikiSplit Corpus

WebSplit provides a basis for measuring progress on splitting and rephrasing sentences. However, its small size, inherent repetitiveness, and synthetic nature limit its broader applicability. In particular, we see it as a viable benchmark for evaluating models, but not for training them. To that end, we introduce the WikiSplit corpus and detail its construction next.

2.1 Mining Wikipedia Edits

Wikipedia maintains snapshots of entire documents at different timestamps, which makes it possible to reconstruct edit histories for documents. This has been exploited for many NLP tasks, including sentence compression (Yamangil and Nelken, 2008), text simplification (Yatskar et al., 2010; Woodsend and Lapata, 2011; Tonelli et al., 2016) and modeling semantic edit intentions (Yang et al., 2017).

To construct the WikiSplit corpus, we identify edits that involve sentences being split. A list of sentences for each snapshot is obtained by stripping HTML tags and Wikipedia markup and running a sentence break detector (Gillick, 2009). Temporally adjacent snapshots of a Wikipedia page are then compared to check for sentences that have undergone a split like that shown in Figure 1. We search for splits in both temporal directions.

Given all candidate examples extracted this way, we use a high-precision heuristic to retain only high quality splits. To extract a full sentence C and its candidate split into $S = (S_1, S_2)$, we

Thresh. δ	Correct	Unsupp.	Miss.	Size
0.1	161	35	6	1.4m
0.2	168	35	4	1.0m
0.3	169	31	4	0.5m

Table 2: Quality vs corpus size trade-off when setting the similarity threshold. The counts are for a random sample of 100 split-and-rephrase examples extracted using our method (i.e., 200 simple sentences). Keys: **Unsupp**orted; **Miss**ing

require that C and S_1 have the same trigram prefix, C and S_2 have the same trigram suffix, and S_1 and S_2 have different trigram suffixes. To filter out misaligned pairs, we use BLEU scores (Papineni et al., 2002) to ensure similarity between the original and the split versions.

Specifically, we discard pairs where BLEU(C, S_1) or BLEU(C, S_2) is less than δ (an empirically chosen threshold). If multiple candidates remain for a given sentence C, we retain $\arg\max_S (\text{BLEU}(C, S_1) + \text{BLEU}(C, S_2))$.[1]

2.2 Corpus Statistics and Quality

Our extraction heuristic is imperfect, so we manually assess corpus quality using the same categorization schema proposed by Aharoni and Goldberg (2018); see Table 1 for examples of *correct*, *unsupported* and *missing* sentences in splits extracted from Wikipedia. We do this for 100 randomly selected examples using three different

---

[1]We attempted to mitigate other noise inherent in Wikipedia by removing items that 1) repeated a token more than three times in a row; 2) contained a token longer than 25 characters; 3) were suggestive of profane vandalism.

	WebSplit		WikiSplit	
	Count	Unique	Count	Unique
C	1.3m	17k	1.0m	1.0m
S'	6.1m	28k	2.0m	1.9m
t	344k	7k	33.1m	633k

Table 3: Training corpus statistics in terms of complex sentences (C), simple sentences ($S'=\cup_i S_i$) and tokens (t, appearing across unique complex sentences). WikiSplit provides much greater diversity and scale.

thresholds of δ. As shown in Table 2, δ=0.2 provides the best trade-off between quality and size.

Out of the 100 complex sentences in the sample, only 4 contained information that was not completely covered by the simple sentences. In our corpus, every complex sentence is split into two simpler sentences, so the sample contains 200 simple sentences. Out of these we found 168 (84%) to be correct, while 35 (18%) contained unsupported facts. Thus, for the overall sample of 100 split-and-rephrase examples, 68% are perfect while 32% contain some noise (either unsupported facts or missing information). We stress that our main goal is to use data extracted this way as training data and accept that its use for evaluation is an imperfect signal with some inherent noise and bias (by construction).

After extraction and filtering, we obtain over one million examples of sentence splits from around 18 million English documents. We randomly reserved 5000 examples each for tuning, validation and testing, producing 989,944 unique complex training sentences, compared to the 16,938 of WebSplit (cf. Table 3).

2.3 Comparison to WebSplit

Narayan et al. (2017) derived the WebSplit corpus by matching up sentences in the WebNLG corpus (Gardent et al., 2017) according to partitions of their underlying meaning representations (RDF triples). The WebNLG corpus itself was created by having crowd workers write sentential realizations of one or more RDF triples. The resulting language is often unnatural, for example, *"Akeem Dent once played for the Houston Texans team which is based in Houston in Texas."*[2]

Repetition arises because the same sentence fragment may appear in many different examples.

This is to be expected given that WebSplit's small vocabulary of 7k words must account for the 344k tokens that make up the distinct complex sentences themselves.[3]

This is compounded in that each sentence contains a named entity by construction. In contrast, our large new WikiSplit dataset offers more natural and diverse text (see examples in Table 1), having a vocabulary of 633k items covering the 33m tokens in its distinct complex sentences.

The task represented by our WikiSplit dataset is *a priori* both harder and easier than that of the WebSplit dataset – harder because of the greater diversity and sparsity, but potentially easier due to the uniform use of a single split.

Of the two datasets, WebSplit is better suited for evaluation: its construction method guarantees cleaner data than is achieved by our extraction heuristic, and it provides multiple reference decompositions for each complex sentence, which tends to improve the correlation of automatic metrics with human judgment in related text generation tasks (Toutanova et al., 2016).

3 Experiments

In order to understand how WikiSplit can inform the split-and-rephrase task, we vary the composition of the training set when training a fixed model architecture. We compare three training configurations: WEBSPLIT only, WIKISPLIT only, and BOTH, which is simply their concatenation.

Text-to-text training instances are defined as all the unique pairs of (C, S), where C is a complex sentence and S is its simplification into multiple simple sentences (Narayan et al., 2017; Aharoni and Goldberg, 2018). For training, we delimit the simple sentences with a special symbol. We depart from the prior work by only using a subset of the WebSplit training set: we take a fixed sub-sample such that each distinct C is paired with a single S, randomly selected from the multiple possibilities in the dataset. This scheme produced superior performance in preliminary experiments.

As a quality measure, we report multi-reference *corpus-level* BLEU[4] (Papineni et al., 2002), but

---

[2]Given RDF triple: {(H_Txns, city, Texas), (Akeem_Dent, formerTeam, H_Txns), (H_Txns, city, Houston)}.

[3]We use **WebSplit v1.0** throughout, which is the scaled-up re-release by Narayan et al. (2017) at `http://github.com/shashiongithub/Split-and-Rephrase`, commit `a9a288c`. Preliminary experiments showed the same trends on the smaller v0.1 corpus, as resplit by Aharoni and Goldberg (2018).

[4]Using NLTK v3.2.2, with case sensitive scoring.

↓train/eval→	WebSplit 1.0	WikiSplit
SOURCE	58.0	73.4
SPLITHALF	54.9	71.7
WEBSPLIT	35.3	4.2
WIKISPLIT	59.4	76.0
BOTH	**61.4**	**76.1**

Table 4: Corpus-level BLEU scores on the validation sets for the same model architecture trained on different data.

	BLEU	sBLEU	#S/C	#T/S
Reference		–	2.5	10.9
SOURCE	58.7	56.1	1.0	20.5
SPLITHALF	55.7	53.0	2.0	10.8
AG18	30.5	25.5	2.3	11.8
WEBSPLIT	34.2	30.5	2.0	8.8
WIKISPLIT	60.4	58.0	2.0	11.2
BOTH	**62.4**	**60.1**	2.0	11.0

Table 5: Results on the WebSplit v1.0 test set when varying the training data while holding model architecture fixed: corpus-level BLEU, sentence-level BLEU (to match past work), simple sentences per complex sentence, and tokens per simple sentence (micro-average). AG18 is the previous best model by Aharoni and Goldberg (2018), which used the full WebSplit training set, whereas we downsampled it.

include sentence-level BLEU (sBLEU) for direct comparison to past work.[5] We also report length-based statistics to quantify splitting.

We use the same sequence-to-sequence architecture that produced the top result for Aharoni and Goldberg (2018), "Copy512", which is a one-layer, bi-directional LSTM (cell size 512) with attention (Bahdanau et al., 2014) and a copying mechanism (See et al., 2017) that dynamically interpolates the standard word distribution with a distribution over the words in the input sentence. Training details are as described in the Appendix of Aharoni and Goldberg (2018) using the OpenNMT-py framework (Klein et al., 2017).[6]

3.1 Results

We compare to the SOURCE baseline, which is the previously reported method of taking the unmodified input sentence as prediction, and we add SPLITHALF, the natural baseline of deterministically splitting a complex sentence into two equal-length token sequences and appending a period to the first one.

Table 4 compares our three training configurations on the validation sets of both WebSplit and WikiSplit. The WEBSPLIT model scores 35.3 BLEU on the WebSplit validation set but fails to generalize beyond its narrow domain, as evidenced by reaching only 4.2 BLEU on the WikiSplit validation set.

The example predictions in Table 7 illustrate how this model tends to drop content ("Alfred Warden", "mouth", "Hamburg"), hallucinate common elements from its training set ("food", "ingredient", "publisher") and generally fails to produce coherent sentences.

In contrast, the WIKISPLIT model achieves 59.4 BLEU on the WebSplit validation set, *without observing any in-domain data*. It also outperforms the two deterministic baselines on both validation sets by a non-trivial BLEU margin. This indicates that the WikiSplit training data enable better generalization than when using WebSplit by itself. Reintroducing the downsampled, in-domain training data (BOTH) further improves performance on the WebSplit evaluation.

These gains in BLEU from using WikiSplit carry over to the blind manual evaluation we performed on a random sample of model predictions on the WebSplit validation set. As shown in Table 6, the BOTH model produced the most accurate output (95% correct simple sentences), with the lowest incidence of missed or unsupported statements. Our manual evaluation includes the corresponding outputs from Aharoni and Goldberg (2018) (AG18), which were 22% accurate.

The examples in Table 7 demonstrate that the WIKISPLIT and BOTH models produce much more coherent output which faithfully rephrases the input. In Example 1, the combined model (BOTH) produces three fluent sentences, overcoming the strong bias toward two-sentence output inherent in the majority of its training examples.

We relate our approach to prior work on Web-Split v1.0 by reporting scores on its test set in Table 5. Our best performance in BLEU is again obtained by combining the proposed WikiSplit dataset with the downsampled WebSplit, yielding a 32 point improvement over the prior best result.

---

[5]Past work on WebSplit (Narayan et al., 2017; Aharoni and Goldberg, 2018) reported macro-averaged *sentence-level* BLEU, calculated without smoothing precision values of zero. We found this ill-defined case occurred often for low-quality output.

[6]`github.com/OpenNMT/OpenNMT-py`, 0ecec8b

Training	Unsupported	Missing	Repeated	Correct
AG18	82	45	12	26/119 (22%)
WEBSPLIT	58	47	13	32/100 (32%)
WIKISPLIT	8	5	0	91/100 (91%)
BOTH	**4**	**4**	**0**	**95/100 (95%)**

Table 6: Manual evaluation results, as counts over the simple sentences predicted by each model for a random sample of 50 inputs from WebSplit 1.0 validation set.

Example 1 from WebSplit validation set

Input	*Alan Bean , born in Wheeler , Texas , was a crew member of Apollo 12 where Alfred Worden served as the backup pilot .*
AG18	Alan Bean is a member of the crew of Apollo 12 which was operated by backup pilot . Alan Shepard was born in Texas .
WEBSPLIT	Alan Bean served as a crew member of Apollo 12 . Alan Bean was born in Wheeler .
WIKISPLIT	Alan Bean , born in Wheeler , Texas . He was a crew member of Apollo 12 where Alfred Worden served as the backup pilot .
BOTH	Alan Bean was born in Wheeler , Texas . Alan Bean was a crew member of Apollo 12 . Alfred Worden served as the backup pilot .

Example 2 from WikiSplit validation set

Input	*He is a short , dark grayish - green monster with a beak - like mouth and a cap made out of the upper part of a creature 's skull .*
WEBSPLIT	He is a food found in grayish . grayish monster is an ingredient in grayish .
WIKISPLIT	He is a short , dark grayish - green monster with a beak - like mouth . A cap made out of the upper part of a creature 's skull .
BOTH	He is a short , dark grayish - green monster with a beak - like mouth . His cap is made out of the upper part of a creature 's skull .

Example 3 from WikiSplit validation set

Input	*Der beglückte Florindo is an opera composed by Handel in Hamburg in 1708 as part of a double opera , the other part being Die verwandelte Daphne .*
WEBSPLIT	Handel opera is the publisher of the opera opera . Handel is the capital of 1708 .
WIKISPLIT	Der beglückte Florindo is an opera composed by Handel in Hamburg in 1708 . It was part of a double opera , the other part being Die verwandelte Daphne .
BOTH	Der beglückte Florindo is an opera composed by Handel in Hamburg in 1708 as part of a double opera . The other part being Die verwandelte Daphne .

Table 7: Example model predictions for items from each validation set. AG18 gives the output of the Copy512-model of Aharoni and Goldberg (2018), while the other outputs are from our models trained on the corresponding data.

4 Conclusion and Outlook

Our results demonstrate a large, positive impact on the split-and-rephrase task when training on large, diverse data that contains some noise. This suggests that future improvements may come from finding other such sources of data as much as from modeling. The new WikiSplit dataset is intended as training data, but for further progress on the split-and-rephrase task, we ideally need evaluation data also derived from naturally occurring sentences, and an evaluation metric that is more sensitive to the particularities of the task.

5 Acknowledgments

References

Roee Aharoni and Yoav Goldberg. 2018. Split and Rephrase: Better Evaluation and a Stronger Baseline. In *Proc. of ACL*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proc. of EMNLP*.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating Training Corpora for NLG Micro-Planners. In *Proc. of ACL*.

Dan Gillick. 2009. Sentence Boundary Detection and the Problem with the U.S. In *Proc. of NAACL*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proc. of ACL*.

Luheng He, Mike Lewis, and Luke Zettlemoyer. 2015. Question-Answer Driven Semantic Role Labeling: Using Natural Language to Annotate Natural Language. In *Proc. of EMNLP*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proc. of ACL, System Demonstrations*.

Philipp Koehn and Rebecca Knowles. 2017. Six Challenges for Neural Machine Translation. In *Proc. of the First Workshop on Neural Machine Translation*.

Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. 2017. Split and Rephrase. In *Proc. of EMNLP*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proc. of ACL*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proc. of ACL*.

Matthew Shardlow. 2014. A Survey of Automated Text Simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.

Advaith Siddharthan. 2006. Syntactic Simplification and Text Cohesion. *Research on Language and Computation*, 4(1):77–109.

Advaith Siddharthan. 2014. A survey of research on text simplification. *International Journal of Applied Linguistics*, 165(2):259–298.

Sara Tonelli, Alessio Palmero Aprosio, and Francesca Saltori. 2016. SIMPITIKI: a Simplification corpus for Italian. In *Proc. of CLiC-it*.

Kristina Toutanova, Chris Brockett, Ke M. Tran, and Saleema Amershi. 2016. A Dataset and Evaluation Metrics for Abstractive Compression of Sentences and Short Paragraphs. In *Proc. of EMNLP*.

Kristian Woodsend and Mirella Lapata. 2011. Learning to Simplify Sentences with Quasi-Synchronous Grammar and Integer Programming. In *Proc. of EMNLP*.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in Current Text Simplification Research: New Data Can Help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Elif Yamangil and Rani Nelken. 2008. Mining Wikipedia Revision Histories for Improving Sentence Compression. In *Proc. of ACL*.

Diyi Yang, Aaron Halfaker, Robert Kraut, and Eduard Hovy. 2017. Identifying Semantic Edit Intentions from Revisions in Wikipedia. In *Proc. of EMNLP*.

Mark Yatskar, Bo Pang, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2010. For the sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia. In *Proc. of NAACL*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware Attention and Supervised Data Improve Slot Filling. In *Proc. of EMNLP*.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A Monolingual Tree-based Translation Model for Sentence Simplification. In *Proc. of Coling*.