Split and Rephrase: Better Evaluation and a Stronger Baseline

Roee Aharoni & Yoav Goldberg
Computer Science Department
Bar-Ilan University
Ramat-Gan, Israel
`{roee.aharoni,yoav.goldberg}@gmail.com`

Abstract

Splitting and rephrasing a complex sentence into several shorter sentences that convey the same meaning is a challenging problem in NLP. We show that while vanilla seq2seq models can reach high scores on the proposed benchmark (Narayan et al., 2017), they suffer from memorization of the training set which contains more than 89% of the unique simple sentences from the validation and test sets. To aid this, we present a new train-development-test data split and neural models augmented with a copy-mechanism, outperforming the best reported baseline by 8.68 BLEU and fostering further progress on the task.

1 Introduction

Processing long, complex sentences is challenging. This is true either for humans in various circumstances (Inui et al., 2003; Watanabe et al., 2009; De Belder and Moens, 2010) or in NLP tasks like parsing (Tomita, 1986; McDonald and Nivre, 2011; Jelínek, 2014) and machine translation (Chandrasekar et al., 1996; Pouget-Abadie et al., 2014; Koehn and Knowles, 2017). An automatic system capable of breaking a complex sentence into several simple sentences that convey the same meaning is very appealing.

A recent work by Narayan et al. (2017) introduced a dataset, evaluation method and baseline systems for the task, naming it "Split-and-Rephrase". The dataset includes 1,066,115 instances mapping a single complex sentence to a sequence of sentences that express the same meaning, together with RDF triples that describe their semantics. They considered two system setups: a text-to-text setup that does not use the accompany-

ing RDF information, and a semantics-augmented setup that does. They report a BLEU score of 48.9 for their best text-to-text system, and of 78.7 for the best RDF-aware one. We focus on the text-to-text setup, which we find to be more challenging and more natural.

We begin with vanilla SEQ2SEQ models with attention (Bahdanau et al., 2015) and reach an accuracy of 77.5 BLEU, substantially outperforming the text-to-text baseline of Narayan et al. (2017) and approaching their best RDF-aware method. However, manual inspection reveal many cases of unwanted behaviors in the resulting outputs: (1) many resulting sentences are *unsupported* by the input: they contain correct facts about relevant entities, but these facts were not mentioned in the input sentence; (2) some facts are *repeated*—the same fact is mentioned in multiple output sentences; and (3) some facts are *missing*—mentioned in the input but omitted in the output.

The model learned to *memorize entity-fact pairs* instead of learning to split and rephrase. Indeed, feeding the model with examples containing entities alone without any facts about them causes it to output perfectly phrased but unsupported facts (Table 3). Digging further, we find that 99% of the simple sentences (more than 89% of the unique ones) in the validation and test sets also appear in the training set, which—coupled with the good memorization capabilities of SEQ2SEQ models and the relatively small number of distinct simple sentences—helps to explain the high BLEU score.

To aid further research on the task, we propose a more challenging split of the data. We also establish a stronger baseline by extending the SEQ2SEQ approach with a copy mechanism, which was shown to be helpful in similar tasks (Gu et al., 2016; Merity et al., 2017; See et al., 2017). On the original split, our models outperform the

	count	unique
RDF entities	32,186	925
RDF relations	16,093	172
complex sentences	1,066,115	5,544
simple sentences	5,320,716	9,552
train complex sentences	886,857	4,438
train simple sentences	4,451,959	8,840
dev complex sentences	97,950	554
dev simple sentences	475,337	3,765
test complex sentences	81,308	554
test simple sentences	393,420	4,015
% dev simple in train	99.69%	90.9%
% test simple in train	99.09%	89.8%
% dev vocab in train	97.24%	
% test vocab in train	96.35%	

Table 1: Statistics for the WEBSPLIT dataset.

Model	BLEU	#S/C	#T/S
SOURCE	55.67	1.0	21.11
REFERENCE	–	2.52	10.93
Narayan et al. (2017)			
HYBRIDSIMPL	39.97	1.26	17.55
SEQ2SEQ	48.92	2.51	10.32
MULTISEQ2SEQ*	42.18	2.53	10.69
SPLIT-MULTISEQ2SEQ*	77.27	2.84	11.63
SPLIT-SEQ2SEQ*	78.77	2.84	9.28
This work			
SEQ2SEQ128	76.56	2.53	10.53
SEQ2SEQ256	77.48	2.57	10.56
SEQ2SEQ512	75.92	2.59	10.59

Table 2: BLEU scores, simple sentences per complex sentence (#S/C) and tokens per simple sentence (#T/S), as computed over the test set. SOURCE are the complex sentences and REFERENCE are the reference rephrasings from the test set. Models marked with * use the semantic RDF triples.

best baseline of Narayan et al. (2017) by up to 8.68 BLEU, without using the RDF triples. On the new split, the vanilla SEQ2SEQ models break completely, while the copy-augmented models perform better. In parallel to our work, an updated version of the dataset was released (v1.0), which is larger and features a train/test split protocol which is similar to our proposal. We report results on this dataset as well. The code and data to reproduce our results are available on Github.[1] We encourage future work on the split-and-rephrase task to use our new data split or the v1.0 split instead of the original one.

2 Preliminary Experiments

Task Definition In the split-and-rephrase task we are given a complex sentence C, and need to produce a sequence of simple sentences $T_1, ..., T_n$, $n \geq 2$, such that the output sentences convey all and only the information in C. As additional supervision, the split-and-rephrase dataset associates each sentence with a set of RDF triples that describe the information in the sentence. Note that the number of simple sentences to generate is not given as part of the input.

Experimental Details We focus on the task of splitting a complex sentence into several simple ones *without* access to the corresponding RDF triples in either train or test time. For evaluation we follow Narayan et al. (2017) and compute the averaged individual multi-reference BLEU score for each prediction.[2] We split each prediction to

sentences[3] and report the average number of simple sentences in each prediction, and the average number of tokens for each simple sentence. We train vanilla sequence-to-sequence models with attention (Bahdanau et al., 2015) as implemented in the OPENNMT-PY toolkit (Klein et al., 2017).[4] Our models only differ in the LSTM cell size (128, 256 and 512, respectively). See the supplementary material for training details and hyperparameters. We compare our models to the baselines proposed in Narayan et al. (2017). HYBRIDSIMPL and SEQ2SEQ are text-to-text models, while the other reported baselines additionally use the RDF information.

Results As shown in Table 2, our 3 models obtain higher BLEU scores then the SEQ2SEQ baseline, with up to 28.35 BLEU improvement, despite being single-layer models vs. the 3-layer models used in Narayan et al. (2017). A possible explanation for this discrepancy is the SEQ2SEQ baseline using a dropout rate of 0.8, while we use 0.3 and only apply it on the LSTM outputs. Our results are also better than the MULTISEQ2SEQ and SPLIT-MULTISEQ2SEQ models, which use explicit RDF information. We also present the macro-average[5] number of sim-

---

Input	Prediction
A Fortress of Grey Ice with ISBM 0-7653-0633-6 has 672 pages .	J.V. Jones authored A Fortress of Grey Ice . A Fortress of Grey Ice has 672 pages .
The address , 11 Diagonal Street is located in South Africa where the leader is Cyril Ramaphosa and some Asian South Africans live .	The address , 11 Diagonal Street is located in South Africa . The leader of South Africa is called Cyril Ramaphosa . The leader of South Africa is called Cyril Ramaphosa . The leader of South Africa is called Cyril Ramaphosa .
Alan Shepard Alan Shepard Alan Shepard	Alan Shepard is dead . Alan Shepard was a test pilot .
AFC Ajax AFC Ajax AFC Ajax	AFC Ajax 's manager is Jong Ajax . AFC Ajax N.V. own Sportpark De Toekomst .

Table 3: Predictions from a vanilla SEQ2SEQ model, illustrating unsupported facts, missing facts and repeated facts. The last two rows show inputs we composed to demonstrate that the models memorize entity-fact pairs.

ple sentences per complex sentence in the reference rephrasings (REFERENCE) showing that the SPLIT-MULTISEQ2SEQ and SPLIT-SEQ2SEQ baselines may suffer from over-splitting since the reference splits include 2.52 simple sentences on average, while the mentioned models produced 2.84 sentences.
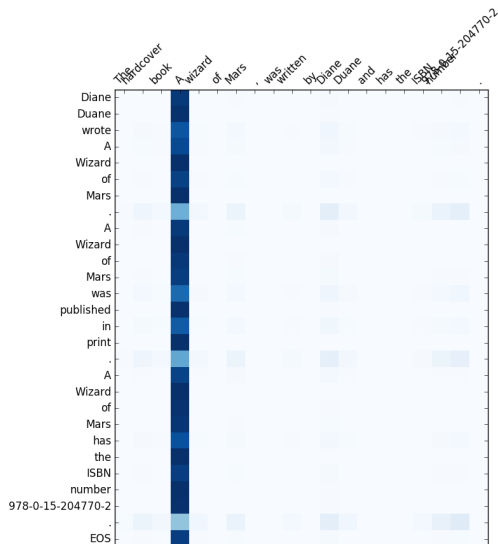

Figure 1: SEQ2SEQ512's attention weights. Horizontal: input. Vertical: predictions.

Analysis We begin analyzing the results by manually inspecting the model's predictions on the validation set. This reveals three common kinds of mistakes as demonstrated in Table 3: unsupported facts, repetitions, and missing facts. All the unsupported facts seem to be related to entities mentioned in the source sentence. Inspecting the attention weights (Figure 1) reveals a worrying trend: throughout the prediction, the model focuses heavily on the first word in of the first entity ("A wizard of Mars") while paying little attention to other cues like "hardcover", "Diane" and

references of a specific complex sentence, and then average these numbers.

"the ISBN number". This explains the abundance of "hallucinated" unsupported facts: rather than learning to split and rephrase, the model learned to identify entities, and spit out a list of facts it had memorized about them. To validate this assumption, we count the number of predicted sentences which appeared as-is in the training data. We find that 1645 out of the 1693 (97.16%) predicted sentences appear verbatim in the training set. Table 1 gives more detailed statistics on the WEBSPLIT dataset.

To further illustrate the model's recognize-and-spit strategy, we compose inputs containing an entity string which is duplicated three times, as shown in the bottom two rows of Table 3. As expected, the model predicted perfectly phrased and correct facts about the given entities, although these facts are clearly not supported by the input.

3 New Data-split

The original data-split is not suitable for measuring generalization, as it is susceptible to "cheating" by fact memorization. We construct a new train-development-test split to better reflect our expected behavior from a split-and-rephrase model. We split the data into train, development and test sets by randomly dividing the 5,554 distinct complex sentences across the sets, while using the provided RDF information to ensure that:

1. Every possible RDF relation (e.g., BORNIN, LOCATEDIN) is represented in the training set (and may appear also in the other sets).

2. Every RDF triplet (a complete fact) is represented only in one of the splits.

While the set of complex sentences is still divided roughly to 80%/10%/10% as in the original split, now there are nearly no simple sentences in

	count	unique
train complex sentences	1,039,392	4,506
train simple sentences	5,239,279	7,865
dev complex sentences	13,294	535
dev simple sentences	39,703	812
test complex sentences	13,429	503
test simple sentences	41,734	879
# dev simple in train	35 (0.09%)	
# test simple in train	1 (0%)	
% dev vocab in train	62.99%	
% test vocab in train	61.67%	
dev entities in train	26/111 (23.42%)	
test entities in train	25/120 (20.83%)	
dev relations in train	34/34 (100%)	
test relations in train	37/37 (100%)	

Table 4: Statistics for the RDF-based data split

the development and test sets that appear verbatim in the train-set. Yet, every relation appearing in the development and test sets is supported by examples in the train set. We believe this split strikes a good balance between challenge and feasibility: to succeed, a model needs to learn to identify relations in the complex sentence, link them to their arguments, and produce a rephrasing of them. However, it is not required to generalize to unseen relations. [6]

The data split and scripts for creating it are available on Github.[7] Statistics describing the data split are detailed in Table 4.

4 Copy-augmented Model

To better suit the split-and-rephrase task, we augment the SEQ2SEQ models with a copy mechanism. Such mechanisms have proven to be beneficial in similar tasks like abstractive summarization (Gu et al., 2016; See et al., 2017) and language modeling (Merity et al., 2017). We hypothesize that biasing the model towards copying will improve performance, as many of the words in the simple sentences (mostly corresponding to entities) appear in the complex sentence, as evident by the relatively high BLEU scores for the SOURCE baseline in Table 2.

Copying is modeled using a "copy switch" probability $p(z)$ computed by a sigmoid over a learned composition of the decoder state, the context vector and the last output embedding. It interpolates the $p_{softmax}$ distribution over the target vocabulary and a copy distribution p_{copy} over the source sentence tokens. p_{copy} is simply the computed attention weights. Once the above distribu-

		BLEU	#S/C	#T/S
original data split	SOURCE	55.67	1.0	21.11
	REFERENCE	–	2.52	10.93
	SPLIT-SEQ2SEQ	78.77	2.84	9.28
	SEQ2SEQ128	76.56	2.53	10.53
	SEQ2SEQ256	77.48	2.57	10.56
	SEQ2SEQ512	75.92	2.59	10.59
	COPY128	78.55	2.51	10.29
	COPY256	83.73	2.49	10.66
	COPY512	87.45	2.56	10.50
new data split	SOURCE	55.66	1.0	20.37
	REFERENCE	–	2.40	10.83
	SEQ2SEQ128	5.55	2.27	11.68
	SEQ2SEQ256	5.28	2.27	10.54
	SEQ2SEQ512	6.68	2.44	10.23
	COPY128	16.71	2.0	10.53
	COPY256	23.78	2.38	10.55
	COPY512	24.97	2.87	10.04
v1.0	SOURCE	56.1	1.0	20.4
	REFERENCE	–	2.48	10.69
	COPY512	25.47	2.29	11.74

Table 5: Results over the test sets of the original, our proposed split and the v1.0 split

tions are computed, the final probability for an output word w is:

$$p(w) = p(z = 1)p_{copy}(w) + p(z = 0)p_{softmax}(w)$$

In case w is not present in the output vocabulary, we set $p_{softmax}(w) = 0$. We refer the reader to See et al. (2017) for a detailed discussion regarding the copy mechanism.

5 Experiments and Results

Models with larger capacities may have greater representation power, but also a stronger tendency to memorize the training data. We therefore perform experiments with copy-enhanced models of varying LSTM widths (128, 256 and 512). We train the models using the negative log likelihood of $p(w)$ as the objective. Other than the copy mechanism, we keep the settings identical to those in Section 2. We train models on the original split, our proposed data split and the v1.0 split.

Results Table 5 presents the results. On the original data-split, the COPY512 model outperforms all baselines, improving over the previous best by 8.68 BLEU points. On the new data-split, as expected, the performance degrades for all models, as they are required to generalize to sentences not seen during training. The copy-augmented models perform better than the baselines in this case as well, with a larger relative gap which can be explained by the lower lexical overlap between the train and the test sets in the new split. On the v1.0 split the results are similar to those on our split,

Input	Prediction
Madeleine L'Engle who is influenced by George Macdonald wrote " A Severed Wasp " .	A Severed Wasp was written by George Macdonald . A Severed Wasp is from the United States .
The A.C. Lumezzane has 4150 members and play in the Lega Pro League .	A.C. Lumezzane 's ground is in the Lega Pro League . A.C. Lumezzane has 4150 members .
Irish English is the official language of Ireland , which is lead by Enda Kenny and home to Adare Manor .	Adare Manor is located in Ireland . English is the language of Ireland . English is the language of Ireland . (repeated x3)

Table 6: Predictions from the COPY512 model, trained on the new data split.

in spite of it being larger (1,331,515 vs. 886,857 examples), indicating that merely adding data will not solve the task.

Analysis We inspect the models' predictions for the first 20 complex sentences of the original and new validation sets in Table 7. We mark each simple sentence as being "correct" if it contains all and only relevant information, "unsupported" if it contains facts not present in the source, and "repeated" if it repeats information from a previous sentence. We also count missing facts. Figure 2 shows the attention weights of the COPY512 model for the same sentence in Figure 1. Reassuringly, the attention is now distributed more evenly over the input symbols. On the new splits, all models perform catastrophically. Table 6 shows outputs from the COPY512 model when trained on the new split. On the original split, while SEQ2SEQ128 mainly suffers from missing information, perhaps due to insufficient memorization capacity, SEQ2SEQ512 generated the most unsupported sentences, due to overfitting or memorization. The overall number of issues is clearly reduced in the copy-augmented models.
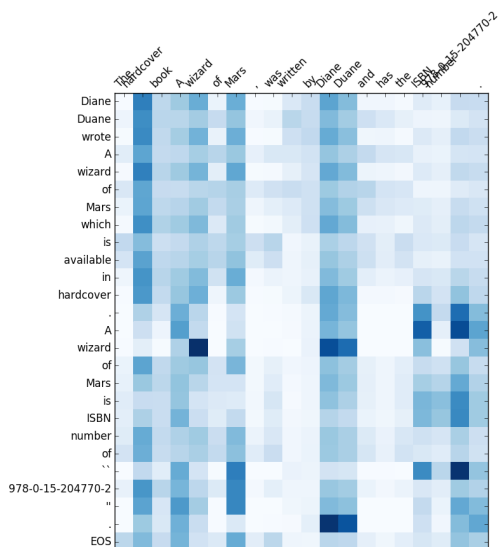

Figure 2: Attention weights from the COPY512 model for the same input as in Figure 1.

Model	unsup.	repeated	correct	missing
original split				
SEQ2SEQ128	5	4	40/49 (82%)	9
SEQ2SEQ256	2	2	42/46 (91%)	5
SEQ2SEQ512	12	2	36/49 (73%)	5
COPY128	3	4	42/49 (86%)	4
COPY256	3	2	45/50 (90%)	6
COPY512	5	0	46/51 (90%)	3
new split				
SEQ2SEQ128	37	8	0	54
SEQ2SEQ256	41	7	0	54
SEQ2SEQ512	43	5	0	54
COPY128	23	3	2/27 (7%)	52
COPY256	35	2	3/40 (7%)	49
COPY512	36	13	11/54 (20%)	43
v1.0 split				
COPY512	41	3	3/44 (7%)	51

Table 7: Results of the manual analysis, showing the number of simple sentences with unsupported facts (unsup.), repeated facts, missing facts and correct facts, for 20 complex sentences from the original and new validation sets.

6 Conclusions

We demonstrated that a SEQ2SEQ model can obtain high scores on the original split-and-rephrase task while not actually learning to split-and-rephrase. We propose a new and more challenging data-split to remedy this, and demonstrate that the cheating SEQ2SEQ models fail miserably on the new split. Augmenting the SEQ2SEQ models with a copy-mechanism improves performance on both data splits, establishing a new competitive baseline for the task. Yet, the split-and-rephrase task (on the new split) is still far from being solved. We strongly encourage future research to evaluate on our proposed split or on the recently released version 1.0 of the dataset, which is larger and also addresses the overlap issues mentioned here.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Raman Chandrasekar, Christine Doran, and Bangalore Srinivas. 1996. Motivations and methods for text simplification. In *Proceedings of the 16th conference on Computational linguistics*. Association for Computational Linguistics.

Jan De Belder and Marie-Francine Moens. 2010. Text simplification for children. In *Proceedings of the SIGIR workshop on accessible search systems*. ACM.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany. http://www.aclweb.org/anthology/P16-1154.

Kentaro Inui, Atsushi Fujita, Tetsuro Takahashi, Ryu Iida, and Tomoya Iwakura. 2003. Text simplification for reading assistance: a project note. In *Proceedings of the second international workshop on Paraphrasing-Volume 16*. Association for Computational Linguistics.

Tomáš Jelínek. 2014. Improvements to dependency parsing using automatic simplification of data. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*. Association for Computational Linguistics, Vancouver, Canada. http://aclweb.org/anthology/P17-4012.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*. Association for Computational Linguistics, Vancouver. http://www.aclweb.org/anthology/W17-3204.

Ryan McDonald and Joakim Nivre. 2011. Analyzing and integrating dependency parsers. *Computational Linguistics* 37.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. 2017. Split and rephrase. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. http://aclweb.org/anthology/D17-1064.

Jean Pouget-Abadie, Dzmitry Bahdanau, Bart van Merrienboer, Kyunghyun Cho, and Yoshua Bengio. 2014. Overcoming the curse of sentence length for neural machine translation using automatic segmentation. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Association for Computational Linguistics, Doha, Qatar. http://www.aclweb.org/anthology/W14-4009.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics. http://www.aclweb.org/anthology/P17-1099.

Masaru Tomita. 1986. Efficient parsing for natural languagea fast algorithm for practical systems. int. series in engineering and computer science.

Willian Massami Watanabe, Arnaldo Candido Junior, Vinícius Rodriguez Uzêda, Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro Pardo, and Sandra Maria Aluísio. 2009. Facilita: reading assistance for low-literacy readers. In *Proceedings of the 27th ACM international conference on Design of communication*. ACM.