Dataset	#Train	#Dev	#Test	Ref len	Doc len
CNN	90266	1220	1093	37	540
DM	196961	12148	10397	61	593
NYT	137778	17222	17223	88	727

Table 1: Statistics of the CNN, Daily Mail, and NYT (see text) datasets. CNN features the shortest reference summaries overall, and this is where we find compression is most effective.

A Experimental Setup

Data Preprocessing We preprocess the datasets with the scripts provided by See et al. (2017), which uses Stanford CoreNLP tokenization Manning et al. (2014). We use the non-anonymized version of the CNN/DM as in previous summarization work. For the New York Times Corpus, we filter out the examples with abstracts shorter than 50 words following the criteria in (Durrett et al., 2016), yielding the NYT dataset. The statistics of the datasets are listed in Table 1. During sentence selection, we always select 3 sentences for CNN/DM and 5 sentences for NYT, which gave the best performance. For our syntactic analysis, all datasets are parsed with the constituency parser in Stanford CoreNLP (Manning et al., 2014).

Implementation Details We use the same pretrained word embeddings used in (Narayan et al., 2018). The size of the sentence and document representation vectors is 200. For the compression module, we use ELMo as the contextualized encoder without fine-tuning the parameter and project the vectors back to 200 dimensions after the ELMo layer. Dropout is applied after word embedding layers and LSTM layers at a rate of 0.2. We use the Adam optimizer (Kingma and Ba, 2014) with the initial learning rate at 0.001. The model converges after 2 epochs of training. In initial experiments, we also found ELMo to be useful for sentence selection as well. However, to simplify comparisons with past work and due to scaling issues, we use it for compression only. We use ROUGE (Lin, 2004) for evaluation.[1] During oracle construction, we use simplified unigram and bigram F_1 scores as a faster approximation to the full ROUGE.

[1]Command line parameters: "-c 95 -m -n 2"

Node Type	Len	% of comps	Oracle comp %
PP	5.7	39%	67%
JJ	1.0	19%	84%
SBAR	12.1	11%	59%
ADVP	1.3	7%	91%

Table 2: Statistics of compression options in CNN. We show the top four constituency types that are compressible, along with the average length, the fraction of available compressions it accounts for, and how frequently the oracle says to compress these constituents.

B Turk Instructions

Figure 1 shows the interface for Amazon turk human evaluation.

C Type Analysis

In Table 2, we show the statistics of the compression options in CNN. PP attachment and adjectives are the top 2 compression options and according to the oracle, more than half of PP and almost all of the adjectives are compressable without hurting the ROUGE.

References

Greg Durrett, Taylor Berg-Kirkpatrick, and Dan Klein. 2016. Learning-Based Single-Document Summarization with Compression and Anaphoricity Constraints. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1998–2008. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759. Association for Computational Linguistics.

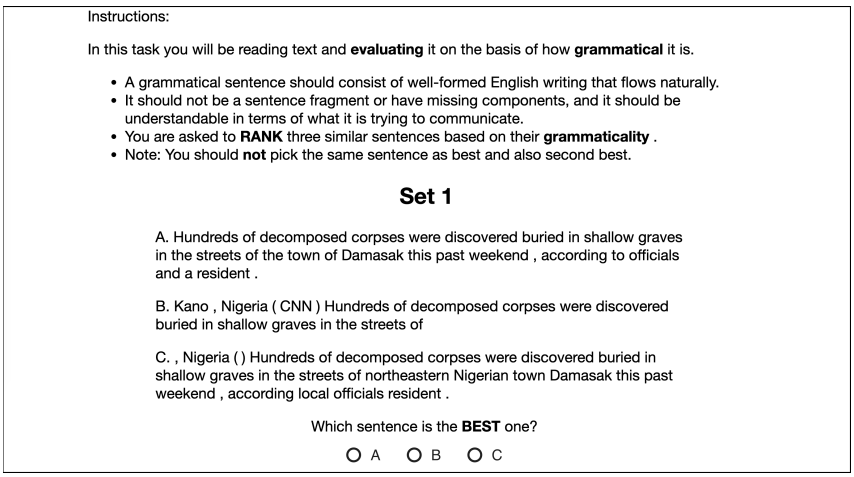

Figure 1: The interface for Amazon turk human evaluation. All of the examples are fully shuffled.

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)*, pages 1073–1083. Association for Computational Linguistics.