Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 6638–6644
Marseille, 11–16 May 2020

Abstractive Document Summarization without Parallel Data

Nikola I. Nikolov and **Richard H.R. Hahnloser**
Institute of Neuroinformatics, University of Zürich and ETH Zürich, Switzerland
{niniko, rich}@ini.ethz.ch

Abstract

Abstractive summarization typically relies on large collections of paired articles and summaries. However, in many cases, parallel data is scarce and costly to obtain. We develop an abstractive summarization system that relies only on large collections of example summaries and non-matching articles. Our approach consists of an unsupervised sentence extractor that selects salient sentences to include in the final summary, as well as a sentence abstractor that is trained on pseudo-parallel and synthetic data, that paraphrases each of the extracted sentences. We perform an extensive evaluation of our method: on the CNN/DailyMail benchmark, on which we compare our approach to fully supervised baselines, as well as on the novel task of automatically generating a press release from a scientific journal article, which is well suited for our system. We show promising performance on both tasks, without relying on any article-summary pairs.

Keywords: automatic summarization, abstractive summarization, text rewriting, low-resource settings

1. Introduction

Text summarization aims to produce a shorter, informative version of an input text. While extractive summarization only selects important sentences from the input, abstractive summarization generates content without explicitly re-using whole sentences (Nenkova et al., 2011) resulting summaries that are more fluent. In recent years, a number of successful approaches have been proposed for both extractive (Nallapati et al., 2017; Narayan et al., 2018) and abstractive (See et al., 2017; Chen and Bansal, 2018) summarization paradigms. State-of-the-art abstractive approaches are supervised, relying on large collections of paired articles and summaries. However, competitive performance of abstractive systems remains a challenge when the availability of parallel data is limited, such as in low-resource domains or for languages other than English.

Even when parallel data is severely limited, we may have access to example summaries and large collections of articles on similar topics. Examples are blog posts or scientific press releases, for which the original articles may be unavailable or behind a paywall.

In this paper, we develop a system[1] for abstractive document summarization (Figure 1) that only relies on example summaries and non-matching articles, bypassing the need for large-scale parallel corpora. Our system consists of two components: First, an unsupervised **sentence extractor** selects salient sentences. Second, each extracted sentence is paraphrased using a **sentence abstractor**. The abstractor is trained on pseudo-parallel data extracted from raw corpora, as well as on additional synthetic data generated through backtranslation.

We evaluate our approach on two summarization tasks. First, on the CNN/DailyMail news article summarization dataset, we compare the performance of our method based on non-parallel data to fully supervised models based on parallel data (Section 5.1.). Second, we test our system on the novel task of automatically generating a press release from a scientific journal article (Section 6.). This task is well suited for our system because only a small number of aligned document pairs can be extracted from available repositories of scientific journal articles and press releases. On both tasks, our method achieves promising results without relying on any parallel article-summary pairs.

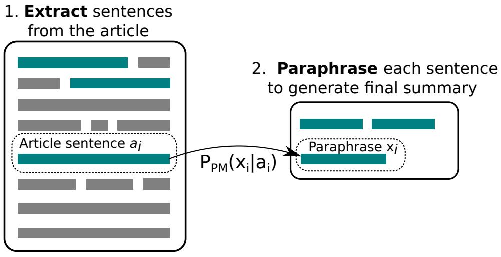
1. **Extract** sentences from the article
2. **Paraphrase** each sentence to generate final summary
Article sentence a_i
Paraphrase x_i
$P_{PM}(x_i|a_i)$

Figure 1: Overview of our approach to abstractive document summarization without parallel data: given an input article $\boldsymbol{a} = a_1, ..., a_N$ consisting of N sentences, (1) we fist select salient sentences a_i (in blue) using an unsupervised extractive summarization algorithm; we then (2) generate paraphrases x_i using a sentence paraphrasing model P_{PM}, trained on pseudo-parallel and synthetic data.

2. Background

2.1. Supervised Summarization

In recent years, there have been large advances in supervised **abstractive summarization**, for headline generation (Rush et al., 2015; Nallapati et al., 2017) as well as for generation of multi-sentence summaries (See et al., 2017). State-of-the-art approaches are typically trained to generate summaries either in a fully end-to-end fashion (See et al., 2017), processing the entire article at once, or hierarchically, first extracting content and then paraphrasing it sentence-by-sentence (Chen and Bansal, 2018).

Both approaches rely on large collections of article-summary pairs such as the annotated Gigaword (Napoles et al., 2012) or the CNN/DailyMail dataset (Nallapati et al., 2016). The heavy reliance on manually curated resources prohibits the use of abstractive summarization in domains other than news articles, or languages other than English, where parallel data may not be as abundantly available. In

[1] https://github.com/ninikolov/low_resource_summarization

such areas, extractive summarization often remains the preferred choice.

2.2. Unsupervised Summarization

Unsupervised summarization has a long history within the extractive summarization paradigm. Given an input article consisting of N sentences $\boldsymbol{a} = \{a_1, ..., a_N\}$, the goal of **extractive summarization** is to select the K most salient sentences as the output summary, without employing any paraphrasing or fusion. A typical approach is to weigh each sentence either with respect to the document as a whole (Radev et al., 2004) or through an adjacency-based measure of sentence importance (Erkan and Radev, 2004).

There is less work on unsupervised abstractive summarization. (Chu and Liu, 2019) propose a review summarization system based on autoencoders. Their focus is, however, on multi-document rather than on single-document summarization as it is in our case. (Dohare et al., 2018) develop a pipeline for semantic abstractive summarization which works by constructing a graph from the article and then generating a summary from the most informative part of the graph. (Isonuma et al., 2019) propose an abstractive summarization framework based on learning the discourse structure of input articles, and generating a single-sentence summary using a language model trained to reconstruct sentences from example reviews.

Our work focuses on multi-sentence abstractive summarization using large-scale non-parallel resources such as collections of summaries *without* matching articles. Recently, several methods have been proposed to reduce the need for parallel data either through harvesting pseudo-parallel data from raw corpora (Nikolov and Hahnloser, 2019) or by synthesizing data using backtranslation (Sennrich et al., 2016). Such methods have been shown to be viable for a number of tasks such as unsupervised machine translation (Lample et al., 2018), sentence compression (Fevry and Phang, 2018), and style transfer (Lample et al., 2019). To the best of our knowledge, our work is the first to extend such methods to single-document summarization to generate multi-sentence abstractive summaries in a data-driven fashion.

3. Approach

Our system (see Figure 1) consists of two components: an **extractor** (Section 3.1.) that picks salient sentences to include in the final summary and an **abstractor** (Section 3.2.) that subsequently paraphrases each of the extracted sentences, rewriting them to meet the target summary style.

Our approach is similar to (Chen and Bansal, 2018), except that they use parallel data to train their extractors and abstractors. In contrast, during training, we only assume access to M example summaries $\boldsymbol{S} = \{\boldsymbol{s}_0, .., \boldsymbol{s}_M\}$ without matching articles. During testing, given an input article consisting of N sentences $\boldsymbol{a} = \{a_0, ..., a_N\}$, our system is capable of generating a multi-sentence abstractive summary consisting of K sentences (where K is a hyperparameter).

3.1. Sentence Extractor

The extractor selects the K most salient article sentences to include in the summary. We consider two unsupervised variants for the extractor:

LEAD picks the first K sentences from the article and returns them as the summary. For many datasets, such as CNN/DailyMail, LEAD is a simple but tough baseline to beat, especially using abstractive methods (See et al., 2017). Because LEAD may not be the optimal choice for other domains or datasets, we experiment with another unsupervised extractive approach.

LEXRANK (Erkan and Radev, 2004) represents the input as a highly connected graph in which vertices represent sentences and edges between sentences are assigned weights equal to their term-frequency inverse document frequency (TF-IDF) similarity, provided that the TF-IDF similarity is higher than a predefined threshold t. The centrality of a sentence is then computed using the PageRank algorithm.

3.2. Sentence Abstractor

The sentence abstractor (P_{PM}) is trained to generate a paraphrase x_i for every article sentence a_i, rewriting it to meet the target sentence style of the summaries. We implement P_{PM} as an LSTM encoder-decoder with an attention mechanism (Bahdanau et al., 2014). Instead of training the abstractor on parallel examples of sentences from articles and summaries, we train it on a synthetic dataset created in two steps (summarized in Figure 2):

Pseudo-parallel dataset. The first step is to obtain an initial set of pseudo-parallel article-summary sentence pairs. Because we assume access to *some* example summaries, our approach is to align summary sentences to an external corpus of comparable articles. Here, we apply the large-scale alignment method from (Nikolov and Hahnloser, 2019), which hierarchically aligns documents followed by sentences in the two datasets (see Figure 2, step 1). The alignment is implemented using nearest neighbor search: first on document and then on sentence embeddings.

Backtranslated pairs. We use the initial pseudo-parallel dataset to train a backtranslation model $P_{\text{BT}}(x_i|s_i)$, following (Sennrich et al., 2016). The model learns to synthesize ”fake” article sentences given a summary sentence (see Figure 2, 2). We use P_{BT} to generate multiple synthetic article sentences x_{ij} for each summary sentence s_i available, taking the $j = 1, \ldots, J$ top hypotheses predicted by beam search[2].

To train our final sentence paraphrasing model $P_{\text{PM}}(s_i|x_i)$, we combine all pseudo-parallel and backtranslated pairs into a single dataset of article-summary sentence pairs.

4. Experimental Set-up

Implementation details. P_{PM} and P_{BT} are both implemented as bidirectional LSTM encoder-decoder models with 256 hidden units, embedding dimension 128, and an

[2] We also experimented with sampling (Edunov et al., 2018) but found it to be too noisy in the current setting.

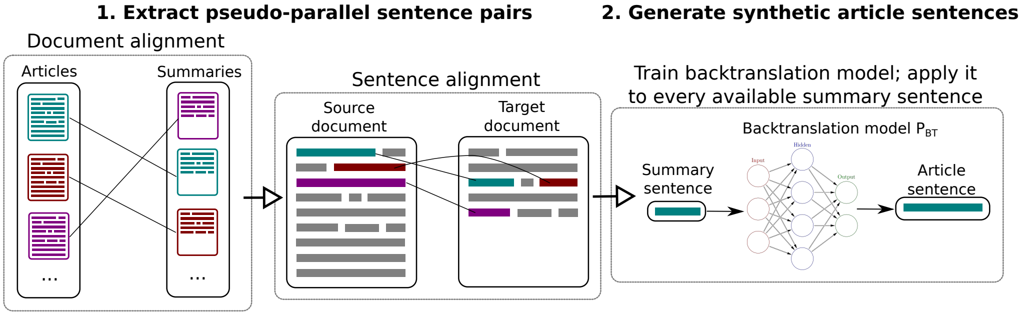
1. Extract pseudo-parallel sentence pairs
2. Generate synthetic article sentences
Document alignment
Articles
Summaries
...
...
Sentence alignment
Source document
Target document
Train backtranslation model; apply it to every available summary sentence
Backtranslation model P_{BT}
Summary sentence
Article sentence
Input
Hidden
Output

Figure 2: Overview of our pipeline for constructing the training dataset of our sentence abstractor P_{PM}. (1), given a dataset of summaries and articles on similar topics, we extract pseudo-parallel document and sentence pairs using large-scale alignment (Nikolov and Hahnloser, 2019). (2), we use the pseudo-parallel pairs to train a backtranslation model P_{BT}, which we use to synthesize article sentences given a summary sentence.

Approach	R-1	R-2	R-L	MET	#
ORACLE	47.33	26.43	43.69	30.76	132
Unsupervised extractive baselines					
LEAD	38.78	17.57	35.49	23.67	119
LEXRANK	34.49	14.1	31.32	21.27	133
Supervised abstractive baselines (Trained on parallel data)					
LSTM	35.61	15.04	32.7	16.24	58
EXT-ABS†	38.38	16.12	36.04	19.39	
EXT-ABS-RL†	40.88	17.8	38.54	20.38	73
Abstractive summarization without parallel data (this work)					
LEAD + ABS$_{\text{PP+SYN-5}}$	32.98	11.13	30.88	13.51	50
LEXRANK + ABS$_{\text{PP+SYN-5}}$	30.87	9.42	28.82	12.51	52

Table 1: Metric results on the CNN/Daily Mail test set. **R-1/2/L** are the **ROUGE-1/2/L** F1 scores; **MET** is **METEOR**, while **#** is the average number of tokens in the summaries. † are from (Chen and Bansal, 2018).

(1) ABS$_{\text{PP-0.63}}$: cnn is the first time in three years . the other contestants told the price of the price . the game show will be hosted by the tv game show . the game of the game is the first of the show .
(2) ABS$_{\text{PP+SYN-5}}$: a tv legend has returned to the first time in eight years . contestants told the price of " the price is right " bob barker hosted the tv game show for 35 years . the game is the first of the show 's " lucky seven "
(3) ABS$_{\text{PAR}}$: a tv legend returned to doing what he does best . contestants told to " come on down ! " on april 1 edition . he hosted the tv game show for 35 years before stepping down in 2007 . barker handled the first price-guessing game of the show , the classic " lucky seven "

Table 2: Example outputs on the CNN/DM dataset (LEAD extractor): (1)/(2) are trained on pseudo-parallel/synthetic data, while the abstractor in (3) is trained on parallel data.

Approach (# pairs)	R-1	R-2	R-L	#
ABS$_{\text{PP-0.60}}$ (2M)	23.08	4.06	21.48	62
ABS$_{\text{PP-0.63}}$ (1.2M)	28.08	7.07	26.14	49
ABS$_{\text{PP-0.67}}$ (0.3M)	24.36	4.76	22.64	57
ABS$_{\text{PP+SYN-1}}$ (2.4M)	31.92	10.2	29.9	51
ABS$_{\text{PP+SYN-5}}$ (6.6M)	32.98	11.13	30.88	50
ABS$_{\text{PP+SYN-10}}$ (12M)	32.8	11.2	30.71	49
ABS$_{\text{PP-UB}}$ (575K)	38.42	15.98	35.8	65
ABS$_{\text{PAR}}$ (1M)	38.68	16.36	36.15	62

Table 3: Comparison of abstractors trained on parallel (ABS$_{\text{PAR}}$) vs. pseudo-parallel data (ABS$_{\text{PP-}\theta_s}$, using different sentence alignment thresholds θ_s; ABS$_{\text{PP-UB}}$ is the upper bound for large-scale alignment) or using a mixture of pseudo-parallel and synthetic data (ABS$_{\text{PP+SYN-N}}$, using the ABS$_{\text{PP-0.63}}$ dataset and backtranslated data from the top N beam hypotheses). We always use the LEAD extractor.

attention mechanism (Bahdanau et al., 2014). We pick this model size to be comparable to recent work (See et al., 2017; Chen and Bansal, 2018). We set the vocabulary size to 50k and train both models until convergence with Adam (Kingma and Ba, 2015); P_{PM} uses beam search with a beam of 5 during testing.

Because both of our extractor variants are unsupervised, we directly apply them to the articles to select salient sentences. We always set the number K of sentences to be extracted to the average number of summary sentences in the target dataset, $K = 4$ for the CNN/DailyMail dataset, and $K = 25$ for the scientific article dataset.

Evaluation details. We evaluate our systems using the ROUGE-1/2/L F1 metrics (Lin, 2004) and METEOR (Banerjee and Lavie, 2005), which is often used in machine translation.

5. Experiments on CNN/DailyMail

We use the CNN/DailyMail (CNN/DM) dataset (Hermann et al., 2015) consisting of pairs of news articles from CNN and Daily Mail, along with summaries in the form of bullet points. We choose this dataset because it allows us to compare our approach to existing fully supervised methods and to measure the gap between unsupervised and supervised summarization. We follow the preprocessing pipeline of (Chen and Bansal, 2018), splitting the dataset into 287k/11k/11k pairs for training/validation/testing. Note that our method relies only on the bullet-point summaries from this training set.

Obtaining synthetic data. To obtain training data for our sentence abstractor P_{PM}, we follow the procedure from Section 3.2.. We align all summaries from the CNN/DM training set to 8.5M news articles from the Gigaword dataset (Napoles et al., 2012), which *contains no articles from CNN or Daily Mail.* After alignment[3], we obtain 1.2M pseudo-parallel setnence pairs that we use to train our backtranslation model P_{BT}. Using P_{BT}, we synthesize $J = 5$ article sentences for each of the 1M summary sentences by picking the top 5 beam hypotheses. Our best sentence paraphrasing dataset used to train our final abstractor P_{PM} contains 6.7 million sentence pairs, 18% of which are pseudo-parallel pairs and 82% are backtranslated pairs.

5.1. Results on CNN/DailyMail

Baselines. We compare our models with several supervised and unsupervised baselines. LSTM is a standard bidirectional LSTM model trained to directly generate the CNN/DM summaries from the full CNN/DM articles. EXT-ABS is a hierarchical model consisting of a supervised LSTM extractor and separate abstractor (Chen and Bansal, 2018), both of which are individually trained on the CNN/DM dataset by aligning summary to article sentences. Our work best resembles EXT-ABS except that we do not rely on any parallel data. EXT-ABS-RL is a state-of-the-art summarization system that extends EXT-ABS by jointly tuning the two supervised components using reinforcement learning. We additionally report the performance of our unsupervised extractive baselines, LEAD, and LEXRANK, as well as the result of an oracle (ORACLE) which computes an upper bound for extractive summarization by aligning the ground truth summary sentences to their original articles using ROUGE-1.

Automatic evaluation. Our best abstractive models trained on non-parallel data (LEAD + ABS$_{\text{PP+SYN-5}}$ and LEXRANK + ABS$_{\text{PP+SYN-5}}$ in Table 1) performed worse than the baselines trained on parallel data. However, the results are promising: for example, the ROUGE-L gap between our LEAD model and the supervised LSTM model is only 1.8. When comparing against the EXT-ABS and EXT-ABS-RL models, which perform supervised sentence extraction followed by supervised sentence abstraction, the gap is larger. Furthermore, we observe that, on this dataset, applying our abstractors to the LEAD and LEXRANK extractive baselines leads to a decrease in ROUGE. This could be due to the much shorter length of our abstractive summaries in comparison to the length produced by other systems, indicating that our systems potentially summarize much more aggressively.

Model analysis. In Table 3, we compare the effect of training our abstractor on pseudo-parallel datasets of different sizes (ABS$_{\text{PP-}*}$) as well as on a mixture of pseudo-parallel and backtranslated data (ABS$_{\text{PP+SYN-}*}$). For reference, we also include results from aligning the original dataset of CNN/DM articles and summaries directly. We construct a *parallel* dataset (ABS$_{\text{PAR}}$) of sentence pairs by aligning the original CNN/DM document pairs using ROUGE-1; as well as a pseudo-parallel dataset (ABS$_{\text{PP-UB}}$) by applying the large-scale alignment method to the CNN/DM documents, (without using the document labels directly). The performance difference between ABS$_{\text{PAR}}$ and ABS$_{\text{PP-UB}}$ provides an estimate of the performance loss due to pseudo alignment.

Our best pseudo-parallel abstractor performs poorly in comparison to the parallel abstractor. Adding additional synthetic data is helpful but insufficient to compensate for the performance gap. Furthermore, we observe a diminishing improvement from adding synthetic pairs. By contrast, the large-scale alignment method constructs a pseudo-parallel upper bound that almost perfectly matches the parallel dataset, indicating that potentially the main bottleneck in our system is the domain difference between the articles in Gigaword and the CNN/DM.

Example summaries. In Table 2, we also provide example summaries produced on the CNN/DM dataset. Our final model trained on additional backtranslated data produced much more relevant and coherent sentences than the model trained on pseudo-parallel data only. Despite having seen no parallel examples, the system is capable of generating fluent, abstractive sentences. However, in comparison to the abstractor trained on parallel data, there is still room for further improvement.

6. Experiments on Scientific Articles

The rate of scientific publications grows exponentially (Hunter and Cohen, 2006), calling for efficient automatic summarization tools (Nikolov et al., 2018). An important frontier in scientific text summarization is generating summaries that not only synthesize an article but make it also more accessible to non-specialists (Vadapalli et al., 2018a; Vadapalli et al., 2018b; Tatalovic, 2018). A major challenge towards achieving this goal is the lack of parallel datasets of articles and high-quality summaries.

We introduce the novel task of automatically generating a press release for a scientific article. Although there are already large repositories of scientific articles and press releases, two major obstacles are preventing manual alignment across these resources:

1. Because press releases address very different audiences, they are often written and published separately from the article. Furthermore, there is often no metadata present in the text (such as a digital object identifier (DOI)) that could be used to link a press release to its original scientific article.

2. Even when there is metadata that can be utilized, the full text of the original article may not be accessible, either because it is published in a closed access journal under a restrictive license, or because the full text of the article is not available in an easily parsable format (e.g., only a PDF is available).

These practical challenges call for alternative summarization approaches. The summarization method described in this paper circumvents these problems because it exploits the large existing collections of papers and press releases that exist within open repositories.

[3] We follow the set-up from (Nikolov and Hahnloser, 2019) using the Sent2Vec embedding method (Pagliardini et al., 2018) for computing document/sentence embeddings. We use hyperparameters $\theta_d = 0.5$ and $\theta_s = \{0.60, 0.63, 0.67\}$.

Approach	R-1	R-2	R-L	MET	#
ORACLE	43.79	14.27	41.1	20.05	969
Unsupervised extractive baselines					
LEAD	41.2	11.85	38.87	17.11	688
LEXRANK	38.78	10.31	36.57	16.66	800
Abstractive summarization without parallel data (this work)					
LEAD + ABS$_{\text{PP+SYN-1}}$	42.47	12.11	40.25	15.78	529
LEXRANK + ABS$_{\text{PP+SYN-1}}$	41.04	10.94	38.9	15.38	562

Table 4: Metric results on the Scientific summarization test set. **R-1/2/L** are the **ROUGE-1/2/L** F1 scores; **MET** is **METEOR**, while **#** is the average number of tokens in the summaries.

Dataset	Documents	Tok. per sent.	Sent. per doc.
PubMed	1.5M	24 ± 14	180 ± 98
MEDLINE	16.8M	26 ± 13	7 ± 4
EurekAlert	358K	29 ± 15	25 ± 13
Wikipedia	5.5M	25 ± 16	17 ± 32

Table 5: Datasets used to extract pseudo-parallel monolingual sentence pairs in our style transfer experiments.

LEAD: global reference models , such as preliminary reference earth model (prem ; dziewonski and anderson , 1981) , iasp91 (kennett and engdahl , 1991) and ak135 (kennett et al. , 1995) were created with different data sets , techniques and assumptions , leading to differences in some regions of earth .
LEAD + ABS$_{\text{PP+SYN-1}}$: the study was created with different data sets , techniques and assumptions , leading to differences in some regions of earth .
LEAD: astroglia from the postnatal cerebral cortex can be reprogrammed in vitro to generate neurons following forced expression of neurogenic transcription factors , thus opening new avenues towards a potential use of endogenous astroglia for brain repair .
LEAD + ABS$_{\text{PP+SYN-1}}$: the brain ' s neural network has been reprogrammed to generate neurons from the postnatal cerebral cortex .
LEAD: sequenced the angiotensinogen gene and found that multiple rare variants contribute to variation in angiotensinogen levels ; interestingly , most of these variants sit on the same haplotype background created by three common snps.26 fourth , three common tag snps encompassing mc1r (mim 155555) were independently associated with melanoma in a recent gwas.27 however , resequencing of the candidate gene , mc1r , indicates that these signals can be completely explained by the combined effects of several rare nonsynonymous mutations , suggesting that ignoring rare variants can lead to incorrect inferences on the potential role of candidate genes carrying common snps identified by gwas (f. demenais at al .
LEAD + ABS$_{\text{PP+SYN-1}}$: the researchers found that these variants can lead to incorrect inferences on the potential role of candidate genes carrying common snps .

Table 6: Example sentence paraphrases produced on the scientific dataset (using the LEAD extractor).

Datasets used for alignment. To extract pseudo-parallel sentence pairs for paraphrasing, we rely on four collections of documents. We combine two datasets of scientific articles: PubMed[4], which contains the full text of $1.5M$ open access papers, and Medline[5], which contains over $17M$ scientific abstracts. We obtain press releases through scraping $350k$ articles from Eurekalert[6], an aggregator that covers many scientific disciplines. As an additional resource that contains scientific texts written for a more general audience, we use all articles on Wikipedia, without applying any filtering. Table 5 contains an overview of these datasets.

Evaluation dataset. To create a parallel testing dataset, we use regular expressions to detect digital object identifier (DOI) mentions within the press releases. Given a DOI, we query for the full text of a paper using the Elsevier ScienceDirect API[7]. Using this approach, we were able to compose 6821 parallel pairs of the full text of a paper and its press release (which amounts to only 2% of all press releases available). We use these 6821 pairs as our testing set in our experiments, and exclude them from the alignment procedure.

Extracting pseudo-parallel data. To obtain a pseudo-parallel dataset for scientific sentence paraphrasing, we aligned the scientific papers to the press releases. After alignment[8], we extracted $80k$ pseudo-parallel pairs in total. We additionally aligned PubMed and Medline to all articles on Wikipedia[9], obtaining out-of-domain pairs for this task. We merged all pairs into a single dataset, consisting of $370k$ sentence pairs. We used these pairs to train a backtranslation model P_{BT} from which we synthesized 1 article sentence for each sentence in the press release dataset[10]. Our final sentence paraphrasing dataset used to train the abstractor P_{PM} contains $8.6M$ pairs.

6.1. Results

Baselines. Since there are no existing parallel datasets of scientific articles and press releases, here we do not report results using supervised methods. We compare the performance of our models (LEAD + ABS$_{\text{PP+SYN-1}}$ and LEXRANK + ABS$_{\text{PP+SYN-1}}$) to the two unsupervised extractive base-

[4] www.ncbi.nlm.nih.gov/pmc/tools/openftlist

[5] www.nlm.nih.gov/bsd/medline.html

[6] https://www.eurekalert.org/

[7] https://dev.elsevier.com/sd_apis.html

[8] We use a document similarity threshold $\theta_d = 0.6$ and a sentence similarity threshold $\theta_s = 0.74$.

[9] Using $\theta_d = 0.6$ and $\theta_s = 0.78$.

[10] We synthesize only a single sentence because the press release dataset contains many more sentences than CNN/DailyMail.

lines, LEAD and LEXRANK respectively, as well as to the result of the ORACLE (same as in Section 5.1.) that yields an upper bound for extractive summarization.

Model analysis. Our results are in Table 4. Both of our abstractive models outperformed their respective extractive baselines, indicating that our abstractors are beneficial for this task and have learned useful sentence transformations. The summaries are much shorter after abstraction, indicating that the paraphrased summaries are meaningfully compressed. Surprisingly, the gap between the baselines and the oracle is small, indicating that abstractive summarization is essential for achieving good performance on this task.

Example summaries. We find that the model trained on backtranslated data is often conservative choosing to leave many input sentences almost unchanged. However, as shown by examples in Table 6, the model learned useful transformations by compressing long sentences and utilizing vocabulary adapted to the language of press releases (e.g. ”the researchers” in the last example). The sentence compressions can sometimes seem too drastic when looked at in isolation (e.g., from ”astroglia from the postnatal cerebral cortex” to ”the brain’s neural network” in the second example).

7. Conclusion

We developed an abstractive summarization system that does not rely on parallel resources, but can instead be trained using example summaries and a large collection of non-matching articles, making it particularly relevant to low-resource domains and languages. On the CNN/DailyMail benchmark, our system performed competitively to a fully supervised LSTM baseline trained on the document level. We also achieved promising results on the novel task of automatically generating a press release for a scientific journal article.

Future work will focus on developing novel unsupervised extractors, on decreasing the gap between abstractors trained on parallel and non-parallel data, as well as on developing methods for combining the abstractor and extractor into a single system.

Acknowledgments

We acknowledge support from the Swiss National Science Foundation (grant $31003A\_156976$) and thank the anonymous reviewers for their useful comments.

8. Bibliographical References

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Banerjee, S. and Lavie, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proc. of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Chen, Y.-C. and Bansal, M. (2018). Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of ACL.*

Chu, E. and Liu, P. (2019). Meansum: A neural model for unsupervised multi-document abstractive summarization. In *International Conference on Machine Learning*, pages 1223–1232.

Dohare, S., Gupta, V., and Karnick, H. (2018). Unsupervised semantic abstractive summarization. In *Proceedings of ACL 2018, Student Research Workshop*, pages 74–83.

Edunov, S., Ott, M., Auli, M., and Grangier, D. (2018). Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

Erkan, G. and Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Fevry, T. and Phang, J. (2018). Unsupervised sentence compression using denoising auto-encoders. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 413–422. Association for Computational Linguistics.

Hermann, K. M., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

Hunter, L. and Cohen, K. B. (2006). Biomedical language processing: what’s beyond pubmed? *Molecular cell*, 21(5):589–594.

Isonuma, M., Mori, J., and Sakata, I. (2019). Unsupervised neural single-document summarization of reviews via learning latent discourse structure and its ranking. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2142–2152, Florence, Italy, July. Association for Computational Linguistics.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR).*

Lample, G., Ott, M., Conneau, A., Denoyer, L., and Ranzato, M. (2018). Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium, October-November. Association for Computational Linguistics.

Lample, G., Subramanian, S., Smith, E., Denoyer, L., Ranzato, M., and Boureau, Y.-L. (2019). Multiple-attribute text rewriting. In *International Conference on Learning Representations.*

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out.*

Nallapati, R., Zhou, B., dos Santos, C., GuÌlçehre, Ç., and Xiang, B. (2016). Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany, August. Association for Computational Linguistics.

Nallapati, R., Zhai, F., and Zhou, B. (2017). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Napoles, C., Gormley, M., and Van Durme, B. (2012). Annotated gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 95–100. Association for Computational Linguistics.

Narayan, S., Cohen, S. B., and Lapata, M. (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.

Nenkova, A., Maskey, S., and Liu, Y. (2011). Automatic summarization. In *Proc. of ACL*, page 3. Association for Computational Linguistics.

Nikolov, N. I. and Hahnloser, R. (2019). Large-scale hierarchical alignment for data-driven text rewriting. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2019*.

Nikolov, N., Pfeiffer, M., and Hahnloser, R. (2018). Data-driven summarization of scientific articles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France, may. European Language Resources Association (ELRA).

Pagliardini, M., Gupta, P., and Jaggi, M. (2018). Unsupervised learning of sentence embeddings using compositional n-gram features. 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 528–540. Association for Computational Linguistics.

Radev, D. R., Jing, H., Styś, M., and Tam, D. (2004). Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938.

Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal, September. Association for Computational Linguistics.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *Proc. of ACL*, volume 1, pages 1073–1083.

Sennrich, R., Haddow, B., and Birch, A. (2016). Improving neural machine translation models with monolingual data. In *Proc. of ACL*, pages 86–96. Association for Computational Linguistics.

Tatalovic, M. (2018). AI writing bots are about to revolutionise science journalism: we must shape how this is done. *JCOM: Journal of Science Communication*, 17(1):C1–C1.

Vadapalli, R., Syed, B., Prabhu, N., Srinivasan, B. V., and Varma, V. (2018a). Sci-blogger: A step towards automated science journalism. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1787–1790. ACM.

Vadapalli, R., Syed, B., Prabhu, N., Srinivasan, B. V., and Varma, V. (2018b). When science journalism meets artificial intelligence: An interactive demonstration. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 163–168.