TL;DR: Mining Reddit to Learn Automatic Summarization

Michael Völske and **Martin Potthast** and **Shahbaz Syed** and **Benno Stein**
Faculty of Media, Bauhaus-Universität Weimar, Germany
<firstname>.<lastname>@uni-weimar.de

Abstract

Recent advances in automatic text summarization have used deep neural networks to generate high-quality abstractive summaries, but the performance of these models strongly depends on large amounts of suitable training data. We propose a new method for mining social media for author-provided summaries, taking advantage of the common practice of appending a "TL;DR" to long posts. A case study using a large Reddit crawl yields the Webis-TLDR-17 corpus, complementing existing corpora primarily from the news genre. Our technique is likely applicable to other social media sites and general web crawls.

1 Introduction

Given a document, automatic summarization is the task of generating a coherent shorter version of the document that conveys its main points. Depending on the use case, the target length of a summary may be chosen relative to that of the input document, or it may be limited. Either way, a summary must be considered "accurate" by a human judge in relation to its length: the shorter a summary has to be, the more it will have to abstract over the input text. Automatic *abstractive* summarization can be considered one of the most challenging variants of automatic summarization (Gambhir and Gupta, 2017). But with recent advancements in the field of deep learning, new ground was broken using various kinds of neural network models (Rush et al., 2015; Hu et al., 2015; Chopra et al., 2016; See et al., 2017).

The performance of these kinds of summarization models strongly depends on large amounts of suitable training data. To the best of our knowledge, the top rows of Table 1 list all English-language corpora that have been applied to training and evaluating single-document summarization networks in the past two to three years; only the two largest corpora are of sufficient size to serve as training sets by themselves. At the same time, all of these corpora cover more or less the same text genre, namely news. This is probably due to the relative ease by which news articles can be obtained as well as the fact that the news tend to contain properly written texts, usually from professional journalists. Notwithstanding the usefulness of existing corpora, we argue that the apparent lack of genre diversity currently poses an obstacle to deep learning-based summarization.

Table 1: Top rows: commonly used English-language corpora; bottom row: our contribution.

Corpus	Genre	Training pairs
English Gigaword	News articles	4 million
CNN/Daily Mail	News articles	300,000
DUC 2003	Newswire	624
DUC 2004	Newswire	500
Webis-TLDR-17	**Social Media**	**4 million**

In this regard, we identified a novel, large-scale source of suitable training data from the genre of social media. We benefit from the common practice of social media users summarizing their own posts as a courtesy to their readers: the abbreviation TL;DR, originally used as a response meaning "too long; didn't read" to call out on unnecessarily long posts, has been adopted by many social media users writing long posts in anticipatory obedience and now typically indicates that a summary of the entire post follows. This provides us with a text and its summary—both written by the same person—which, when harvested at scale, is an excellent datum for developing and evaluating an automatic summarization system. In contrast to the state-of-the-art corpora, social media texts are written informally and discuss everyday topics, albeit mostly unstructured and oftentimes poorly written, offering new challenges to the community. Thus, we endeavored to extract a usable dataset specifically suited for abstractive summarization from Reddit, the largest discussion forum on the web, where TL;DR summaries are extensively used. In what follows, we discuss in detail how the data was obtained and preprocessed to compile the Webis-TLDR-17 corpus.

2 Related Work

The summarization community has developed a range of resources for training and evaluating extractive and abstractive summarization systems geared towards a diverse set of different summarization tasks. Table 1 reviews the datasets most commonly used for the basic task of single-document summarization, focusing on datasets used in recent, abstractive approaches.

The English Gigaword Corpus has been the most important summarization resource in recent years, as neural network models have made great progress toward the task of generating news headlines from article texts (Rush et al., 2015; Nallapati et al., 2016). The dataset consists of approximately 10 million news articles along with their headlines, extracted from 7 popular news agencies: Agence France-Presse, Associated Press Worldstream, Central News Agency of Taiwan, Los Angeles Times/Washington Post Newswire Service, Washington Post/Bloomberg Newswire Service, New York Times Newswire Service, and Xinhua News Agency. About 4 million English article-title pairs have typically been used to train, evaluate and test recent summarization systems.

The famous Document Understanding Conference (DUC), hosted by the US National Institute of Standards and Technology (NIST) from 2001 to 2007, yielded two corpora that have been applied to single-document summarization. The DUC 2003 and DUC 2004 corpora consist of a few hundred newswire articles each, along with single-sentence summaries. Generally considered too small to train abstractive summarization systems, past research has focused on the use of various optimization methods—such as non-negative matrix factorization (Lee et al., 2009), support vector regression (Ouyang et al., 2011), and evolutionary algorithms (Alguliev et al., 2013)—to select salient sentences for an extractive summary. Beyond that, recent works in abstractive summarization have used DUC corpora for validation and testing purposes.

In addition to the Gigaword and DUC corpora, whose document-summary pairs consist of only a single sentence in the summary, Nallapati et al. (2016) present a new abstractive summarization dataset based on a passage-based question answering corpus constructed by Hermann et al. (2015). The data is sourced from *CNN* and *Daily Mail* news stories, which are annotated with human-generated, abstractive, multi-sentence summaries.

Next to the English resources listed in Table 1, the LCSTS dataset collected by Hu et al. (2015) is perhaps closest to our own work—both in terms of text genre and collection method. Their dataset comprises 2.5 million content-summary pairs collected from the Chinese social media platform Weibo, a service similar to Twitter in that a post is limited to 140 characters. Weibo users frequently start their posts with a short summary in brackets.

3 Dataset Construction

Reddit is a community centered around social news aggregation, web content rating, and discussion, and, as of mid-2017, one of the ten most-visited sites on the web according to Alexa.[1] Community members submit and curate content consisting of text posts or web links, segregated into channels called *subreddits*, covering general topics such as Technology, Gaming, Finance, Wellbeing, as well as special-interest subjects that may only be relevant to a handful of users. At the time of writing, there are about 1.1 million subreddits. In each subreddit, users submit top-level posts—referred to as submissions—and others reply with comments, reflecting, contradicting, or supporting the submission. Submissions consist of a title and either a web link, or a user-supplied body text; in the latter case, the submission is also called a *self-post*. Comments always have a body text—unless subsequently deleted by the author or a moderator—which may also include inline URLs.

Large crawls of Reddit comments and submissions have recently been made available to the NLP community.[2] For the purpose of constructing our summarization corpus, we employ the set of 286 million submissions and 1.6 billion comments posted to Reddit between 2006 and 2016.

[1] http://www.alexa.com/siteinfo/reddit.com

[2] http://files.pushshift.io/reddit/

Table 2: Filtering steps to get the TL;DR corpus.

Filtering Step	Subreddits	Submissions	Comments
Raw Input	617,812	286,168,475	1,659,361,605
Contains tl.{0,3}dr	37,090	2,081,363	3,755,345
Contains tl;dr[3]	34,380	2,002,684	3,412,371
Non-bot post	34,349	1,894,094	3,379,287
Final Pairs	32,778	1,667,129	2,377,372

Table 3: Examples of content-summary pairs.

Example Submission

Title: Ultimate travel kit
Body: Doing some traveling this year and I am looking to build the ultimate travel kit ... So far I have a Bonavita 0.5L travel kettle and AeroPress. Looking for a grinder that would maybe fit into the AeroPress. This way I can stack them in each other and have a compact travel kit.
TL;DR: What grinder would you recommend that fits in AeroPress?

Example Comment (to a different submission)

Body: Oh man this brings back memories. When I was little, around five, we were putting in a new shower system in the bathroom and had to open up the wall. The plumber opened up the wall first, then put in the shower system, and then left it there while he took a lunch break. After his break he patched up the wall and left, having completed the job. Then we couldn't find our cat. But we heard the cat. Before long we realized it was stuck in the wall, and could not get out. We called up the plumber again and he came back the next day and opened the wall. Out came our black cat, Socrates, covered in dust and filth.
TL;DR: plumber opens wall, cat climbs in, plumber closes wall, fucking meows everywhere until plumber returns the next day

3.1 Corpus Construction

Given the raw data of Reddit submissions and comments, our goal is to mine for TL;DR content-summary pairs. We set up a five-step pipeline of consecutive filtering steps; Table 2 shows the number of posts remaining after each step.

An initial investigation showed that the spelling of TL;DR is not uniform, but many plausible variants exist. To boil down the raw dataset to an upper bound of submissions and comments (collectively posts) that are candidates for our corpus, we first filtered all posts that contain the two letter sequences 'tl' and 'dr' in that order, case-insensitive, allowing for up to three random letters in-between. This included a lot of instances found within URLs, which were thus ignored by default. Next, we manually reviewed a number of example posts for all of the 100 most-frequent spelling variants (covering 90% of the distribution) and found 33 variants to be highly specific to actual TL;DR summaries,[3] whereas the remaining, less frequent, variants contained too much noise to be of use.

The Reddit community has developed many bots for purposes such as content moderation, advertisement or entertainment. Posts by these bots are often well formatted but redundant and irrelevant to the topic at hand. To ensure we collect only posts made by human users—critically, some Reddit users operate TL;DR-bots that produce automatic summaries, which may introduce undesirable noise—we filter out all bot accounts with the help of an extensive list provided by the Reddit community,[4] as well as manual inspection of cases where the user name contained the substring "bot."

For the remaining posts, we attempt to split their bodies at the expression TL;DR to form the content-summary pairs for our corpus. We locate the position of the TL;DR pattern in each post, and split the text into two parts at this point, the part before being considered as the content, and the part following as the summary. In this step, we apply a small set of rules to remove erroneous cases: multiple occurrences of TL;DRs are disallowed for their ambiguity, the length of a TL;DR must be shorter than that of the content, there must be at least 2 words in the content and 1 word in TL;DR. The last rule is very lenient; any other threshold would be artificial (i.e., a 10 word sentence may still be summarizable in 2 words). However, future users of our corpus probably might have more conservative thresholds in mind. We hence provide a subset with a 100 word content threshold.

Reddit allows Markdown syntax in post texts, and many users take advantage of this facility. As this introduces some special characters in the text, we disregard all Markdown formatting, as well as inline URLs, when searching for TL;DRs.

After filtering, we are left with approximately 1.6 million submissions and 2.4 million comments for a total of 4 million content-summary pairs. Table 3 shows one example each of content-summary pairs in submissions and comments. The development of the filtering pipeline went along with many spot-checks to ensure selection precision. As a final corpus validation, we reviewed 1000 randomly selected pairs and found 95% to be correct, a proportion that allows for realistic usage. Nevertheless, we continue on refining the filtering pipeline as systematic errors become apparent.

3.2 Corpus Statistics

For the 4 million content-summary pairs, Table 4 shows distributions of the word counts of content and summary, as well as the ratio of summary to content word count. On average, the content body of submissions tends to be nearly twice as long as

[3] tl dr, tl;dr, tldr, tl:dr, tl/dr, tl; dr, tl,dr, tl, dr, tl-dr, tl'dr, tl: dr, tl.dr, tl ; dr, tl_dr, tldr;dr, tl ;dr, tl\dr, tl/ dr, tld:dr, tl;;dr, tltl;dr, tl~dr, tl / dr, tl :dr, tl - dr, tl\\dr, tl. dr, tl:;dr, tl|dr, tl;sdr, tll;dr, tl : dr, tld;dr

[4] https://www.reddit.com/r/autowikibot/wiki/redditbots

Table 4: Length statistics for the TL;DR corpus.

	Min	Median	Max	Mean	σ
Comments					
Total	3	164	6,880	225.21	210.22
Content	2	144	6,597	202.99	199.19
Summary	1	15	1,816	22.21	27.81
Summ. / Cont.	0.00	0.11	1.00	0.16	0.16
Submissions					
Total	3	296	9,973	416.40	384.72
Content	2	269	9,952	382.75	366.99
Summary	1	22	3,526	33.65	47.87
Summ. / Cont.	0.00	0.08	1.00	0.12	0.13

that of comments, whereas the fraction of the total word count in the summary tends to be higher for submissions (about 11% being typical) than for comments (8%). As the length of a post increases, the length of the summary tends to increase as well (Pearson correlations of 0.40 for submissions and 0.35 for comments), while the ratio of summary to content word count increases only slightly (correlations of 0.11 and 0.07).

3.3 Corpus Verticals

The corpus allows for constructing verticals with regard to content type, content topic, and summary type. Content type refers to submissions vs. comments, the key difference being that submissions include an author-supplied title field, which can serve as an additional source of summary ground truth. Comments may perhaps inherit the title of the submission they were posted to, but topic drift may occur. The submission of the example comment in Table 3 was befittingly entitled “So I found my cat after 6 hours with some power tools...”, referring to a picture of a cat stuck in a wall.

Content topic refers to the subreddit a submission or comment was posted to. While subreddits cover trending topics as well as online culture very well, thus ensuring a broader range of topics than news can deliver, there is currently no ontology grouping them for ease of selection.

In our data exploration, we observed that Reddit users write TL;DRs with various intentions, such as providing a “true” summary, asking questions or for help, or forming judgments and conclusions. Although the first kind of TL;DR posts are most important for training summarization models, yet, the latter allow for various alternative summarization-related tasks. Hence, we exemplify how the corpus may be *heuristically* split according to summary type—other summary type verticals are envisioned.

To estimate the number of true summaries, we extract noun phrases from both content and summary, and retain posts where they intersect. Only 966,430 content-summary pairs—580,391 from submissions and 386,039 from comments—pass this test, but this is a lower bound: since abstractive summaries may well be semantically relevant to a post without sharing any noun phrases.

To extract question summaries, we test for the presence of one of 21 English question words,[5] as well as a question mark, in the summary. We can isolate a subset of 78,710 content-summary pairs this way (see Table 3 top), which allow for training tailored models yielding questions for a summary.

Many posts contain abusive words in the content, the TL;DR, or both (see Table 3 bottom). While retaining vulgarity in a summary may be appropriate, it seems rarely desirable if a model introduces vulgarity of its own. To separate 299,145 vulgar summaries, we use a list of more than 500 English offensive words from Google’s now defunct “What Do You Love” project.[6] Come to think of it, these may still be used to train a swearing summarizer, if only for comedic effect.

4 Conclusion

We show how social media can serve as a source of large-scale summarization training data, and mine a set of 4 million content-summary pairs from Reddit, which we make available to the research community as the Webis-TLDR-17 corpus.[7] Preliminary experiments training the models proposed by Rush et al. (2015) and Nallapati et al. (2016) on our dataset have been promising: by manual inspection of individual samples, they produce useful summaries for many Reddit posts; we leave a quantitative evaluation for future work.

Our filtering pipeline, data exploration, and vertical formation allow for fine-grained control of the data, and can be tailored to one’s own needs. Other data sources should be amenable to mining TL;DRs, too: a cursory examination of the CommonCrawl and Clueweb12 web crawls unearths more than 2 million pages containing the pattern—though extracting clean content-summary pairs will likely require more effort for general web content than for self-contained social media posts.

[5] Extension of the word list at https://en.wikipedia.org/wiki/Interrogative_word with “can”, “should”, “would”, “is”, “could”, “does”, “will” after manual analysis of the corpus.

[6] Obtained via https://gist.github.com/jamiew/1112488

[7] https://www.uni-weimar.de/medien/webis/corpora/

References

Rasim M. Alguliev, Ramiz M. Aliguliyev, and Nijat R. Isazade. 2013. Multiple documents summarization based on evolutionary optimization algorithm. *Expert Syst. Appl.* 40(5):1675–1689. https://doi.org/10.1016/j.eswa.2012.09.014.

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*. pages 93–98. http://aclweb.org/anthology/N/N16/N16-1012.pdf.

Mahak Gambhir and Vishal Gupta. 2017. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review* 47(1):1–66. https://doi.org/10.1007/s10462-016-9475-9.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. pages 1693–1701. http://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend.

Baotian Hu, Qingcai Chen, and Fangze Zhu. 2015. LCSTS: A Large Scale Chinese Short Text Summarization Dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*. Association for Computational Linguistics, pages 1967–1972. http://www.aclweb.org/anthology/D15-1229.

Ju-Hong Lee, Sun Park, Chan-Min Ahn, and Daeho Kim. 2009. Automatic generic document summarization based on non-negative matrix factorization. *Inf. Process. Manage.* 45(1):20–34. https://doi.org/10.1016/j.ipm.2008.06.002.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*. pages 280–290. http://aclweb.org/anthology/K/K16/K16-1028.pdf.

You Ouyang, Wenjie Li, Sujian Li, and Qin Lu. 2011. Applying regression models to query-focused multi-document summarization. *Inf. Process. Manage.* 47(2):227–237. https://doi.org/10.1016/j.ipm.2010.03.005.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*. pages 379–389. http://aclweb.org/anthology/D/D15/D15-1044.pdf.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. *CoRR* abs/1704.04368. To appear in ACL'17. http://arxiv.org/abs/1704.04368.