auto-*h*MDS: Automatic Construction of a Large Heterogeneous Multilingual Multi-Document Summarization Corpus

Markus Zopf
Research Training Group AIPHES
Department of Computer Science, Technische Universität Darmstadt
Hochschulstraße 10, 64289 Darmstadt, Germany
zopf@aiphes.tu-darmstadt.de

Abstract

Automatic text summarization is a challenging natural language processing (NLP) task which has been researched for several decades. The available datasets for multi-document summarization (MDS) are, however, rather small and usually focused on the newswire genre. Nowadays, machine learning methods are applied to more and more NLP problems such as machine translation, question answering, and single-document summarization. Modern machine learning methods such as neural networks require large training datasets which are available for the three tasks but not yet for MDS. This lack of training data limits the development of machine learning methods for MDS. In this work, we automatically generate a large heterogeneous multilingual multi-document summarization corpus. The key idea is to use Wikipedia articles as summaries and to automatically search for appropriate source documents. We created a corpus with 7,316 topics in English and German, which has variing summary lengths and variing number of source documents. More information about the corpus can be found at the corpus GitHub page at https://github.com/AIPHES/auto-hMDS.

Keywords: Summarization, Corpus, Wikipedia

1. Motivation

More and more data is contained in unstructured information sources such as newswire articles, social media posts and micro-blogging messages. No human is able to process all the data belonging to important topics such as news about elections, opinions about the newest smartphone, statements in political discussions, trending topics in research, or natural disasters. Automatic preparation of information from heterogeneous sources is therefore a key challenge to enable humans to make use of all the data available on the Internet.

Previous work on automatic summarization usually use small and homogeneous datasets to evaluate their models. The application of supervised machine learning methods is limited mainly by the size of the datasets. This is in particularly true for abstractive summarization methods, which are usually trained on the only available large single-document summarization (SDS) corpus (Hermann et al., 2015). Recently, Zopf et al. (2016b) proposed a new method to create large multi-document summarization (MDS) corpora. Instead of using humans to write summaries for a specific topic based on previously collected source documents, they propose to use already available documents which can be considered to be summaries and search for appropriate source documents. With this method, Zopf et al. (2016b) created a heterogeneous multi-document summarization corpus manually.

In this work, we investigate if and how the manually performed process described in Zopf et al. (2016b) can be automated to create a large heterogeneous multi-document summarization corpus. Furthermore, we add German topics in addition to English topics to the new corpus. Most summarization corpora only contain English source documents and summaries. We call the newly created corpus auto-*h*MDS.

The analysis of our newly created corpus show that our corpus is indeed much larger than prior corpora. We show that a simple machine learning method can improve their performance if they are provided with more training data. We also provide results of standard baseline summarization methods to generate a reference point for future research.

2. Related Work

A popular subfield in automatic summarization is multi-document summarization with a focus on newswire articles. Popular multi-document summarization corpora were created for the Document Understanding Conference (DUC) shared tasks (Over et al., 2007). The datasets from the 2001-2004 shared tasks are often used to evaluate summarization systems (Erkan and Radev, 2004; Lin and Bilmes, 2011; Cao et al., 2015; Ren et al., 2016). Furthermore, they were used to evaluate the ROUGE and Pyramid evaluation system (Lin, 2004; Passonneau et al., 2005; Nenkova et al., 2007). More MDS corpora were produced for the Text Analysis Conference (TAC) 2008 and 2009 shared tasks and used by popular summarization models (Gillick et al., 2009) as well. The DUC and TAC corpora are rather small. They usually contain about 50 topics and are therefore too small to be used for training by machine learning models. Due to the small size, the evaluation is also problematic. Summarization models are usually evaluated with noisy automatic evaluation systems (Lin, 2004). Therefore, the evaluation results might be inaccurate if the summarization models are only evaluated on a few topic.

Recently, a large single-document summarization corpus (Hermann et al., 2015) has been published. Due to its size, it was used to train both extractive (Zopf et al., 2016a; Nallapati et al., 2017) and abstractive summarization (See et al., 2017; Tan et al., 2017) models. We see that as soon as larger corpora are available, the development of other summarization models becomes possible. This trend can-

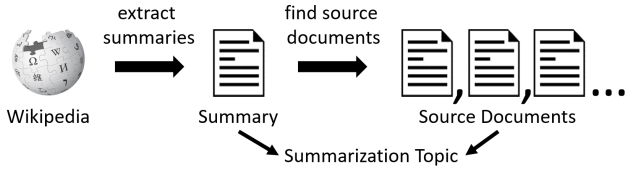

Figure 1: Illustration of the previously proposed corpus construction approach. Wikipedia articles are used as reference summaries (left). Summarization topics can be created in combination with automatically retrieved source documents (right).

not only be observed in summarization. Work on sentence compression (Chopra et al., 2016; Li et al., 2017) use the large Gigaword corpus (Napoles et al., 2012). The small DUC 2004 dataset is only used for testing, not for training. Large datasets are, in addition to new models, a key driver for new AI breakthroughs.[1]

We summarize that available MDS corpora are rather small. Large datasets for single-document summarization are also rare (See et al., 2017). Recent machine learning methods can make use of large SDS corpora for training. Even corpora which do not contain summaries such as the Gigaword corpus can be used as silver standard to train machine learning models. Large heterogeneous corpora are not available (Zopf et al., 2016b).

3. Corpus Construction

The system presented in this work automates the manually performed work in Zopf et al. (2016b). The general idea of the approach is illustrated in Figure 1.

Instead of coming up with topics, searching for source documents, and writing summaries for the source documents, Zopf et al. (2016b) proposed to select already available summaries from Wikipedia and to search for appropriate source documents. The advantage of the new process is that no new text has to be written since the seed for a topic are not the source documents but an already available summary. Since no summaries have to be written, which is usually a difficult and time-consuming task, it seems to be possible to automate the proposed process. In the following, we briefly explain the work performed by Zopf et al. (2016b) and describe how we automate this process.

3.1. Extracting Topics and Summaries

In a first step, Zopf et al. (2016b) select already available text documents on the Internet which can be considered to be a summary of a topic. They use the Wikipedia featured articles as source for summary texts, since they are (i) well-written, (ii) comprehensive, (iii) well-researched, (iv) neutral, and (v) stable according to the Wikipedia featured article criteria[2]. In particular, the first section of each featured article (also called the lead section) is supposed to be a good summary of the topic according to the Wikipedia guidelines. The lead of each featured article contains the most important information about a topic. (Zopf et al., 2016b) extracted 91 English featured article leads manually. We perform this task automatically and retrieved all currently available lead sections of all English and German Wikipedia articles. After creating a list with all Wikipedia featured articles, we use the MediaWiki Action API[3] with properties `extracts|info&exintro&explaintext` to retrieve the lead parts of 7,613 Wikipedia articles. Every article is the seed for one summarization topic. We use the full lead section of each article as summary and do not truncate longer lead sections. The lengths of different summaries can therefore vary a lot. This is an additional new property of the auto-*h*MDS corpus compared to traditional corpora such as the DUC and TAC corpora. These corpora usually have a fixed length for summaries (e.g. 100 words or 665 characters). In the summarization setting provided by auto-*h*MDS, summarization systems have to be able to generate short and long summaries. Details of the summary properties can be found in Section 4.

3.2. Finding Source Documents

In a second step, Zopf et al. (2016b) annotate information nuggets manually. Each extracted information nugget is considered to contain an important piece of information about the topic. The information nuggets are used together with the topic name as query terms in a web search engine to retrieve source documents for the topics. Since the extracted information nugget is used in the web search, the retrieved documents contain the information nugget and can therefore be considered to be source documents for the topics. The cited references in Wikipedia have not been used since we found that they often do not contain the suggested information or were not reachable anymore. By using search results containing the information nuggets, it can be ensured that the link is still alive and that the suggested information is contained in the source document.

The nuggets were labeled by humans in the work performed by Zopf et al. (2016b). Since we aim at creating the corpus fully automatically, a manual labeling step was not appropriate. Instead of extracting information nuggets, we use the sentences contained in the lead sections together with the topic name as search terms. We found that this strategy works very well since sentences in Wikipedia leads are usually rather short and often focused on one piece of information. In the manual extraction performed by Zopf et al. (2016b), usually only one nugget was annotated per sentence which further indicates that searching for sentences is similar to searching for information nuggets. Furthermore, a lot of web pages reuse sentences taken from Wikipedia. This simplifies the web search since it becomes easier to find web pages which fit to the sentences in the lead sections.

To split the lead sections of the retrieved Wikipedia articles, we used the Stanford Segmenter for the English documents and the OpenNLP Segmenter for the German documents. Both are available in the DKPro Core library (Eckart de Castilho and Gurevych, 2014) in version 1.8.0. We found

[1] `https://www.edge.org/response-detail/26587`

[2] `https://en.wikipedia.org/wiki/Wikipedia:Featured_article_criteria`

[3] `https://www.mediawiki.org/wiki/Special:MyLanguage/API:Action_API`

that the automatic sentence splitting worked very well, perhaps also due to the high quality of the Wikipedia featured articles.

For all sentences in all Wikipedia articles, we use the Google Custom Search Engine[4] (CSE) to search for source documents. We use the topic name together with the sentence as query term for the CSE. The *rights* field of the CSE is used to only find sources which can be freely used for non-commercial use-cases. In total, we performed about 93k searches. Since Google charges $5 per 1000 queries, the retrieval of the links for all the sentences costs about $500. This is fairly cheap compared to the costs for paying humans to come up with topics, search for source documents, and write summaries as it was performed for other multi-document summarization corpora.

3.3. Retrieving Source Documents

The result of the invocation of the Google CSE are link lists pointing to web pages which contain the provided query terms (topic name + sentence text). For each sentence, we retrieved up to 10 links. Since some of the sentences occur only rarely on the Internet, we did not obtain 10 links for each query. We obtained about 550k links, in average 5.90 links per query. For each sentence, we tried to downloaded the first web page in the query result. If the pages was not available, we continued with the next URL until we were able to download a page or reached the end of the query result list. To retrieve the best possible snapshot of the web page, we did not only download the HTML code of the web page, but rendered every web page using the Google Chrome browser. We used the Selenium[5] framework to interact with the browser programmatically. This creates better snapshots of web pages since dynamic content can be created or modified (e.g. with JavaScript) before the snapshot is taken. It turned out that using a browser to retrieve the page content improves the quality of the snapshots in particular for web pages which use a lot of JavaScript such as Youtube.

4. Analysis

In this section, we provide a detailed analysis of the result of the effort to automate the construction approach proposed by (Zopf et al., 2016b) and compare the result with previously constructed corpora.

4.1. Corpus Size

In total, we created 5,132 English and 2,481 German topics. Every topics contains one reference summary file, one file which contains one sentence per line (constructed with automatic sentences splitting), and for each sentence a list of URLs. 71,162 and 22,303 sentences are contained in the English and German summaries, respectively. We found 473,754 (in average 6.66 per sentence) and 75,594 (in average 3.39 per sentence) URLs for the German and English corpora, respectively. English and German lead sections have an average length of 13.87 and 8.99 sentences.

[4] https://developers.google.com/custom-search/json-api/v1/overview

[5] http://www.seleniumhq.org

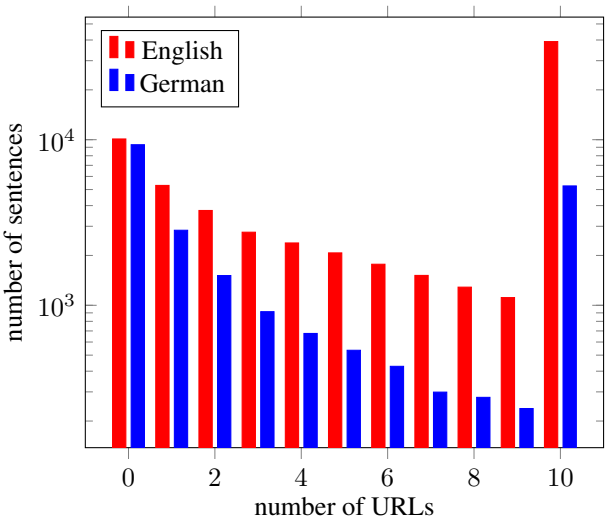

Figure 2: Distribution of retrieved URLs for sentences.

Figure 2 shows a distribution of number of URLs per sentences. We were not able to retrieve results for a significant number of sentences with the search engine in particular for German sentences (English: 10,102 (14.20%), German: 9,325 (41.81%) sentences with an empty search results). One reason for this is the search engine configuration since we only aimed at retrieving only URLs which can be freely used for non-commercial use-cases. We assume that results can be retrieved for almost all sentences if the search is not restricted. Important to note is that lacking source documents is no limitation of the quality of the corpus. Since source documents are missing, the best summary which can generated based on the source documents might not be as good as the reference summary. This, however, only reduces the upper bound reachable by summarization systems without reducing the corpus quality.

Based on the collected URLs, we tried to retrieve one source document for each sentences as described in Section 3.3.. We removed all topics for which we were not able to retrieve any source documents. This resulted in a final corpus size of 5,106 English and 2,210 German summarization topics. Figure 3 shows the distribution of number of source documents across the remaining topics.

	DUC04	TAC09	*h*MDS	auto-*h*MDS
topics	50	44	91	**7,316**
sources	500	440	1,265	**64,744**

Table 1: Size comparison of two standard multi-document summarization datasets, DUC 2004 and TAC 2009, the *h*MDS dataset and the in this paper presented auto-*h*MDS. We report the number of summarization topics and the number of source documents in the corpora.

The main goal of automating the corpus construction is to be able to generate a large corpus for training and evaluating machine learning models. We therefore compare the size of the generated corpus with the DUC 2004[6] and TAC

[6] http://duc.nist.gov/duc2004

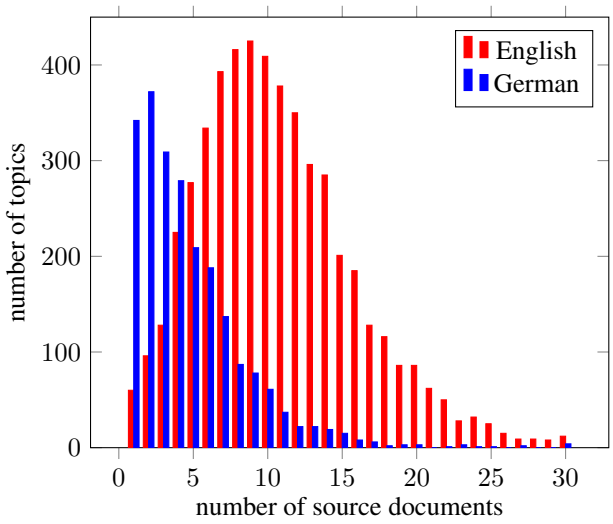

Figure 3: Distribution of source documents. ”30” means ”30 or more”.

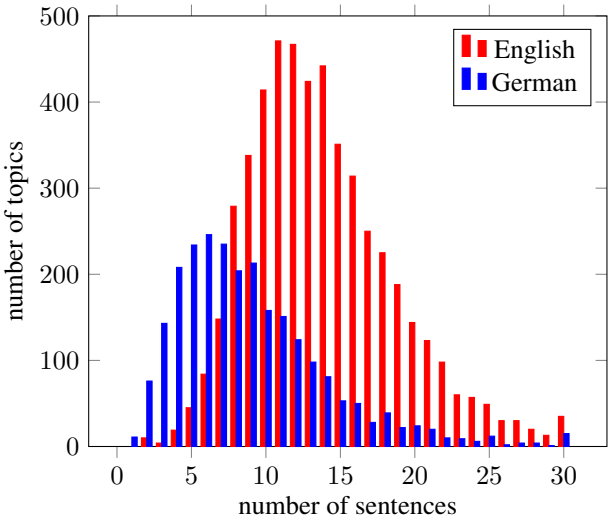

Figure 4: Distribution of sentence lengths in the reference summaries. ”30” means ”30 or more”.

2009[7] multi-document summarization datasets. Both are widely used standard MDS corpora. Other MDS corpora such as other DUC and other TAC corpora have similar sizes. Furthermore, we compare to the manually created *h*MDS corpus. Details are provided in Table 1.

The auto-*h*MDS corpus has many more topics than traditional summarization corpora. In total, we created 7,316 summarization topics. auto-*h*MDS is over 80 times larger in terms of topics (50 times larger in terms of source document) than *h*MDS and about 150 times larger (130 times larger) than standard MDS datasets.

		DUC04	TAC09	*h*MDS	auto-*h*MDS
word	src	672.15	633.89	2972.12	5862.51
	sum	118.12	110.15	245.52	312.42
sent	src	26.28	24.58	268.15	271.36
	sum	6.61	6.16	9.05	12.54

Table 2: Length comparisons according to average number words (word) and average number of sentences (sent) in the source documents (src) and the summaries (sum).

Table 2 provides details about the lengths of source documents and summaries according to the number of words and number of sentences in the corpora. Source documents and summaries in the auto-*h*MDS corpus are longer in comparison to the *h*MDS corpus.

As described in Section 3.1., the length of our summaries can vary in comparison to traditional datasets. Figure 4 illustrates the distribution of sentence lengths for both the English and the German part of the corpus. The English part of the corpus contains longer summaries than the German part. In general, the summaries in auto-*h*MDS are much longer than in traditional corpora where the summary lengths usually range from about 5 to 7 sentences.

4.2. Usability as Training Data

One motivation of building a large summarization corpus is to provide researchers with training data for building summarization systems using machine learning. To evaluate if our corpus can be used by machine learning models, we run experiments where we use the high quality but small *h*MDS corpus and a part of the large automatically created auto-*h*MDS corpus as training data for the summarization model proposed in Zopf et al. (2016a). We evaluate the learned model with the TAC 2009 multi-document summarization corpus. In the experiment, we created short summaries with a maximum length of 50 words. We use different ROUGE-based metrics (Lin, 2004) for evaluation. The results of the experiments are shown in Table 3. In the column *training data*, we report which corpus was used for training as well as how many topics from the corpus were used in brackets.

training data	ROUGE-1	ROUGE-2
*h*MDS (50)	0.2284	0.0307
*h*MDS (91)	**0.2361**	**0.0329**
auto-*h*MDS (50)	0.2362	0.0320
auto-*h*MDS (91)	0.2357	0.0324
auto-*h*MDS (100)	0.2404	0.0328
auto-*h*MDS (200)	0.2481	0.0326
auto-*h*MDS (300)	**0.2485**	**0.0336**

Table 3: Summarization results in the TAC 2009 corpus.

The results indicate that both *h*MDS and auto-*h*MDS can be used similarly well as training data even though the construction of auto-*h*MDS is much cheaper since we do not require any human annotations or interactions. Using only 50 training topics from the *h*MDS corpus yields lower results than using all 91 available topics suggesting that more training data helps the model to learn better. We observe the same effect for the auto-*h*MDS corpus: more training data leads to better results. The best results are achieved with the automatically constructed corpus meaning that the model is able to make better use of more medium-quality training data compared to less high-quality training data. Due to computational limitations, we were only able to use 300 topics of the newly created corpus as training data. We hope that new models which are better suited to learn form

[7] https://tac.nist.gov/2009/Summarization

large multi-document summarization training data will perform even better if more training topics are used.

4.3. Summarization Experiments

The created corpus can not only be used for training machine learning models but can also be used as dataset to test the performance of summarization models. Since the corpus is much larger than traditional corpora and covers a very wide variate of topics and genres, we expect a better performance estimation of summarization systems. Since ROUGE (Lin, 2004) also does not always estimate the performance accurately (Owczarzak et al., 2012), more topics help to improve accuracy in particularly if ROUGE is used as evaluation metric.

In Table 4, we report results for 4 simple extractive summarization baselines which can serve as reference points for future research experiments.

	100 words		200 words	
system	R-1	R-2	R-1	R-2
Random	0.1857	0.0185	0.2553	0.0325
Lead	0.1229	0.0261	0.1056	0.0228
ROUGE-1	0.4302	0.2161	0.4769	0.2117
ROUGE-2	0.4594	0.2927	0.4864	0.2924
Random	0.2290	0.0286	0.2841	0.0434
Lead	0.2524	0.0699	0.2676	0.0790
ROUGE-1	0.5601	0.3812	0.5168	0.3022

Table 4: Summarization performance of different summarization systems in the auto-*h*MDS corpus for different summary lengths (100 and 200 words) and different ROUGE versions (ROUGE-1 and ROUGE-2) for the German (top) and the English (bottom) part of the corpus.

The *Random* baseline chooses sentences randomly until the summary reached the desired length. *Lead* uses the first sentences of the source documents. *ROUGE-1* and *ROUGE-2* choose the best sentences in a greedy fashion according to the ROUGE-1 recall and ROUGE-2 recall score of individual sentences. To compute the scores, both summarization systems use the reference summary. Therefore, both *ROUGE-1* and *ROUGE-2* cannot be considered to be competitive summarization systems but are rather indicators for the best possible scores which can be achieved.

We observe that there is a large performance gap between the *Random* and the *Lead* baselines and the upper bounds achieved by *ROUGE-1* and *ROUGE-2*. This is a promising result since it indicates that the area between random guessing and a very good summarization system is large. The corpus will therefore be usable to distinguish between good and bad summarization systems.

Another interesting observation is that ROUGE scores for the English part are higher than in the German part of the corpus. Not only the baselines achieve higher scores but also the upper bound seems to be higher for English texts.

4.4. Heterogeneity

Last but not least, we analyze the heterogeneity of the created corpus. The topics belong to very diverse genres such as history, religion, sports, science, transport, music, culture, etc. and are therefore even more diverse than the topics in (Zopf et al., 2016b) in which all topics belong to the three genres (i) Art, Architecture, and Archeology (ii) History, and (iii) Law, Politics, and Government. Since we collected all featured articles from Wikipedia, we generated a corpus which covers a lot of interesting topics. The more people are interested in a topic, the more people collect information and work on the according Wikipedia article resulting in high quality articles for generally interesting topics.

5. Conclusions

In this paper, we presented a large, automatically generated multi-document summarization corpus containing topics in English and German. Usually, MDS corpora are rather small and their applicability as training data for machine learning models is limited. Our corpus is much larger than prior corpora and therefore closes this dataset gap. We showed that a machine learning model is indeed able to use the newly created corpus as training data. We hope that larger MDS training corpora will enable researchers to build and train better supervised machine learning models for automatic summarization similarly as we see this trend in single-document summarization with the availability of the large training corpus proposed by Hermann et al. (2015). We make both summaries and link lists containing the links to the source documents freely available. The retrieved source documents are available upon request. Further information can be found at the corpus GitHub page `https://github.com/AIPHES/auto-hMDS`.

6. Acknowledgments

This work has been supported by the German Research Foundation (DFG) as part of the Research Training Group Adaptive Preparation of Information from Heterogeneous Sources (AIPHES) under grant No. GRK 1994/1.

7. Bibliographical References

Cao, Z., Wei, F., Li, S., Li, W., Zhou, M., and Wang, H. (2015). Learning Summary Prior Representation for Extractive Summarization. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 829–833.

Chopra, S., Auli, M., and Rush, A. M. (2016). Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98.

Eckart de Castilho, R. and Gurevych, I. (2014). A broad-coverage collection of portable NLP components for building shareable analysis pipelines. *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*, 2(1):1–11.

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.

Gillick, D., Favre, B., Hakkani-Tür, D., Bohnet, B., Liu, Y., and Xie, S. (2009). The ICSI/UTD Summarization System at TAC 2009. In *Proceedings of the Second Text Analysis Conference*.

Hermann, K., Kocisky, T., and Grefenstette, E. (2015). Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

Li, P., Lam, W., Bing, L., and Wang, Z. (2017). Deep Recurrent Generative Decoder for Abstractive Text Summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2081–2090.

Lin, H. and Bilmes, J. (2011). A Class of Submodular Functions for Document Summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 510–520.

Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 25–26.

Nallapati, R., Zhai, F., and Zhou, B. (2017). SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3075–3081.

Napoles, C., Gormley, M., and Durme, B. V. (2012). Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction & Web-scale Knowledge Extracti*, pages 95–100.

Nenkova, A., Passonneau, R., and McKeown, K. (2007). The Pyramid Method. In *ACM Transactions on Speech and Language Processing*, volume 4, pages 1–23.

Over, P., Dang, H., and Harman, D. (2007). DUC in context. *Information Processing and Management*, 43(6):1506–1520.

Owczarzak, K., Conroy, J. M., Dang, H. T., and Nenkova, A. (2012). An assessment of the accuracy of automatic evaluation in summarization. In *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pages 1–9.

Passonneau, R., Nenkova, A., McKeown, K., and Sigelman, S. (2005). Applying the pyramid method in DUC 2005. In *Proceedings of the Document Understanding Conference*.

Ren, P., Wei, F., and Chen, Z. (2016). A Redundancy-Aware Sentence Regression Framework for Extractive Summarization. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 33–43.

See, A., Liu, P. J., and Manning, C. D. (2017). Get To The Point : Summarization with Pointer-Generator Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1073–1083.

Tan, J., Wan, X., and Xiao, J. (2017). Abstractive Document Summarization with a Graph-Based Attentional Neural Model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1171–1181.

Zopf, M., Loza Mencía, E., and Fürnkranz, J. (2016a). Beyond Centrality and Structural Features: Learning Information Importance for Text Summarization. In *Proceedings of the 20th Conference on Computational Natural Language Learning*, pages 84–94. Association for Computational Linguistics, aug.

Zopf, M., Peyrard, M., and Eckle-Kohler, J. (2016b). The Next Step for Multi-Document Summarization : A Heterogeneous Multi-Genre Corpus Built with a Novel Construction Approach. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 1535–1545.