Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 6731–6739
Marseille, 11–16 May 2020

DaNewsroom: A Large-scale Danish Summarisation Dataset

Daniel Varab and **Natalie Schluter**
IT University of Copenhagen
Copenhagen, Denmark
{djam, natschluter}@itu.dk

Abstract

Dataset development for automatic summarisation systems is notoriously English-oriented. In this paper we present the first large-scale non-English language dataset specifically curated for automatic summarisation. The document-summary pairs are news articles and manually written summaries in the Danish language. There has previously been no work done to establish a Danish summarisation dataset, nor any published work on the automatic summarisation of Danish. We provide therefore the first automatic summarisation dataset for the Danish language (large-scale or otherwise). To support the comparison of future automatic summarisation systems for Danish, we include system performance on this dataset of strong well-established unsupervised baseline systems, together with an oracle extractive summariser, which is the first account of automatic summarisation system performance for Danish. Finally, we make all code for automatically acquiring the data freely available and make explicit how this technology can easily be adapted in order to acquire automatic summarisation datasets for further languages.

Keywords: automatic text summarisation, data collection, danish corpus

1 Introduction

Dataset development for automatic summarisation systems is notoriously English-oriented. This is surprising. On the system user-side, a more feasible access (for example, summaries) to the increasing amounts of digital information informing daily life is of inherent interest to potential users across the globe. At the same time, automatic summarisation provides a challenging NLP test-bed to investigate the limits of deep learning for NLP, and for downstream evaluation of basic core NLP tasks like discourse analysis, co-reference resolution, and other types of parsing (Lee et al., 2018; Li et al., 2016). Yet, only very limited datasets exist in languages other than English (Nguyen and Daumé III, 2019; Schluter and Martínez Alónso, 2016).

By *automatic summarisation dataset* we denote a collection of entire documents each paired up with at least one manually written summary; the summaries of such a dataset are intended as a summaries for those documents and not as headlines, or a list of facts or highlights. In fact, until recently central larger-scale automatic summarisation datasets have not included been composed of any summaries. Namely, Rush et al. (Rush et al., 2015) were the first to recast the English Gigaword dataset (Parker et al., 2011; Napoles et al., 2012) as a headline-type large-scale summarisation generation dataset. And the CNN/Daily Mail dataset (Moritz Hermann et al., 2015), a question answering dataset, was first recast by (Cheng and Lapata, 2016; Nallapati et al., 2016) as an automatic summarisation dataset. These two datasets have been central to more recent automatic summarisation system development.

Headline and highlights datasets are are not ideal for the development of summarisation systems, but because of their scale and in the absence of alternatives, they provided a much needed crucial prerequisite for neural system development.

The advent of the English language Newsroom dataset (Grusky et al., 2018)–a dataset of 1.3 million English article-summary pairs that was created by collecting manually written summaries from news articles provided the first large scale first-class summarisation dataset. To our knowledge, it is also the only existing large-scale automatic summarisation dataset, prior to this paper. With this work, we adopt, extend, and extensively describe an approach to automatically constructing a Danish language automatic summarisation dataset. This essentially (1) provides the first Danish language automatic summarisation dataset, (2) enables neural system development for Danish under a monolingual setting, and establishes (3) the first non-English large-scale automatic summarisation dataset.

Our contributions. With this paper, we contribute the following.

- We establish the **first automatic summarisation dataset for Danish**.
- By contrast to other non-English languages, where new dataset development have been rather limited (i.e., less than 2K document-summary instance pairs) if existent, our dataset, DaNewsroom, is *large-scale*, with more than 1.1 million document-summary instance pairs surviving our quality-control filters. This means, we are presenting **the first non-English large-scale dataset** curated and quality-controlled specifically automatic summarisation system development.
- We **adopt and make key extensions to Grusky et al.'s (2018) methodology** for the development of their Newsroom dataset to the Danish language. In particular, our clarifications, extensions, and associated code presented here **permit researchers to easily develop similar automatic summarisation datasets for other non-English languages**.
- We present the **first account of baseline performance for Danish automatic summarisation** as a point of reference for future neural systems.

We make the code for generation of the dataset, the baseline systems, as well as the dataset itself publicly available[1].

2 Current central datasets for automatic summarisation

We now survey the central datasets for automatic summarisation system development and benchmarking. By "central dataset for automatic summarisation", we mean that the dataset (1) is not a specialised type of summarisation exclusive to a particular domain (like scientific article abstract generation), and that it (2) is typically used in automatic summarisation system benchmarking. All central datasets today are composed entirely of English news articles-summary pairs.

DUC 2004. The DUC2004[2] is currently the most central dataset for automatic summarisation system benchmarking. This is a manually curated multi-document summarisation dataset, whose instance pairs consist of sets of hand-picked, highly related documents paired with summaries about that set of documents, written specifically for the construction of this dataset by different writers.

Despite the added task dimension of having sets of multiple documents to summarise, rather than one single document to summarise, all current state-of-the-art systems to the authors' knowledge first concatenate these multiple documents together into a single document and then summarise the whole as though it were a single document.

Multiple summaries, on the other hand, were meant to provide a more accurate, less author-biased, gauge of system output quality (Nenkova and Passonneau, 2004), by averaging relevant metrics over reference summaries written by different authors.

The DUC 2004 is very small and unsuitable for supervised machine learning in general; as such, it has been primarily used for unsupervised automatic summarisation, and more recently as a test set for neural automatic summarisation systems. The DUC 2004 dataset contains 30 document set-summary set pairs, with an average summary length of 665 bytes/100 words.

CNN/Daily Mail. The CNN/Daily Mail dataset is an automatically generated dataset constructed by crawling cnn.com and dailymail.co.uk. It was originally introduced as a Question Answering dataset (Moritz Hermann et al., 2015) and comprises articles accompanied by information boxes of a couple of bulleted article highlights. These articles were later converted into a summarisation dataset[3] (Cheng and Lapata, 2016; Nallapati et al., 2016) by considering the bullet points as a description of the article and concatenating the listed facts into a single summary. The summaries have on several accounts been described as being highly extractive (Grusky et al., 2018; Chen et al., 2016). The dataset contains over 312K articles of mean length 781 words, accompanied by summaries with mean length 56 words.

Though a useful adaptation, this method for the automatic creation of a new summarisation dataset has two major flaws. First, as we discussed in Section 1, bullet-point highlights are not manually written summaries of articles, and are therefore system development over this dataset does not exactly automatic summarisation system development. Second, data collection is restricted to news outlets who collect highlights in information boxes within news articles. As such, the data collection strategy doesn't correspond to conventional document structure that is generalised across a wide range of news outlets.

Newsroom. Newsroom (Grusky et al., 2018) is a large-scale dataset created by conducting a scrape of news articles from 38 English language news outlets covering the period 1997-2018. The scrape was enabled by the The Internet Archive (archive.org), a non-profit organisation which provides a platform for hosting and accessing past published internet content. Together with archive.org, this work takes advantage of the use of the Semantic Web[4] and properties of Facebook's Open Graph protocol[5] which encouraged online publishers to insert a special metadata summary for each news article. The dataset contains 1.3 million document-summary pairs, with articles of mean length 659 words and mean summary length 27 words.

3 Towards a DaNewsroom

We extend the work of Newsroom (Grusky et al., 2018) and use The Internet Archive[6], a non-profit archiver. Specifically, we use the Wayback Machine[7], a sort of automatic archive system and a product of The Internet Archive. The Wayback Machine has automatically and systematically scraped the internet for the past 20 years. As such, the Wayback Machine provides the history of the web through snapshots and has since collected more than 300 billion websites–all directly accessible through their own online databases. The Wayback Machine provides an API[8] that enables users to query their databases with URLs[9]. Since this historical content is also freely available through the endpoint web.archive.org/web/TIMESTAMP/URL we are equipped with a method to retrieve web content across time, and in this case, news articles from the past. It is with this procedure that Newsroom was collected. Though the process of the acquiring URLs was not described by (Grusky et al., 2018), we provide a reproducible approach with accompanying code for retrieving URLs[10] here.

3.1 Danish News Sites

To construct a dataset from web crawls we first curate a list of sites that will act as search strings for our queries to the Wayback Machine API. This provides the URLs to the stored snapshots hosted in the The Internet Archive's databases. Unlike the English Newsroom, where Grusky

[1] github.com/danielvarab/da-newsroom
[2] https://duc.nist.gov/duc2004/
[3] github.com/abisee/cnn-dailymail
[4] w3.org/standards/semanticweb
[5] ogp.me
[6] archive.org
[7] archive.org/web
[8] web.archive.org/cdx/search/cdx
[9] see documentation at github.com/internetarchive/wayback/tree/master/wayback-cdx-server
[10] github.com/danielvarab/da-newsroom

et al. (2018) used an already curated list of appropriate English language news URLs, no such extensive curated list of Danish media exists, and the Danish Wikipedia[11] only lists nine outlets.

We extend the list from Wikipedia and compose a list of news outlets that are (1) well-known, (2) have existed for the past 20 years, and (3) are included by the Wayback Machine. While the Wayback Machine hosts snaphots of the entire web over time, and in theory across all languages, through manual inspection of coverage of non-English sites it becomes apparent that snapshots are biased towards English sites. A central challenge is therefore the sparse coverage of Danish websites. We list the sites we collect URLs from in Table 1.

DOMAIN	NEWS OUTLET TYPE
altinget.dk	political news outlet
avisen.dk	local news outlet
berlingske.dk	national news outlet
borsen.dk	financial news outlet
bt.dk	tabloid
dagens.dk	local news outlet
dr.dk	national news service
ekstrabladet.dk	tabloid
finans.dk	financial news outlet
fyens.dk	local news outlet
gaffa.dk	music news outlet and blog
ing.dk	tech and science outlet
jyllands-posten.dk	news outlet
kristeligt-dagblad.dk	national news outlet
lokalavisen.dk	collection of local news outlets
nyheder.tv2.dk	national news service
seoghoer.dk	tabloid
version2.dk	tech outlet and blog
videnskab.dk	pop science outlet

Table 1: Danish news sites from which URLs are collected.

We carry out extensive filtering of article-summary pairs based on URL and document contents heuristics (Cf. Sections 3.2 and 5). Figures 1 and 2 show the resulting distribution of article-summary pairs based on domain name and year of publication, respectively.

3.2 Obtaining URLs

Using the list of news sites found in Table 1, we query the Wayback Machine API for URLs. Scraping a domain d is in its most basic form done by calling the archive.org endpoint[12] with the HTTP parameters `url=`d and `matchType=domain`. The `url` parameter acts as a query and specifies a target site, while `matchType` defines which snapshots the query matches (i.e., exact query matches vs. site-match). In addition to these two parameters we use two additional HTTP parameters; `collapse`[13] and `filter`[14]. This removes duplicated URLs and filters out resource/error snapshots. Note that collapsing and filtering also can be done post hoc. We refer to the API documentation for further details of each parameter[15].

This strategy for obtaining URLs produces 14 million URLs snapshots going back 20+ years. A great deal of URLs are, however, of poor quality. In addition, at this large a scale, due to slow download rates from the free archive.org/web service, scraping all possible urls is unfeasible. Therefore detecting noisy (poor quality) urls can help reduce the risk of wasting download time on unusable articles. We therefore filter URLs according to two simple heuristic guidelines.

1. Extract, if any, extension for each URL and prune all instances that contain extensions of common assets such as javascript, stylesheets, fonts, and image files (js/css/tff/png etc.). Most cases of this should be caught by the above *mimetype* filter, however, it only applies to websites that follow conventions and use the appropriate mimetype.

2. Prune URLs that contain the regular expression `(-[a-zA-Z]+){3,}`. This effectively matches URLs that contain three alphabetic sequences delimited by three dashes. We motivate this by the best-practise naming, human-readable URLs (aka "hURLs"), which is a common URL-schema for news outlets that suggests article URLs align with the corresponding article title: for example, *berlingske.dk/samfund/derfor-er-det-saa-svaert-at-vaelge-kampfly*. An example of a URL that is filtered out is *dagens.dk/arkiv/Politik?page=476*. We inspect the results manually and observe a noticeable reduction of unusable pages such as front pages, and web assets.

These two filters reduce the initial 14 million URLs to about 4.8 million, before any document-intrinsic quality control filters (Cf. Section 5).

3.3 Scraping Articles

With a hURL-filtered collection of about 4.86 million candidate URLs we scrape the content found at the end of each candidate URL hosted by the Wayback Machine. We use the Newsroom[16] Python package provided by Grusky et al. (2018) to download articles as well as extract the contents. The package enables concurrent downloads to a compressed format (jsonl+gzip). This is a straight forward, but time consuming process. Downloading documents from a single machine with the default configuration, downloads a mere 1-3.5 articles per second, with frequent stalls and fluctuating download speeds. The final scrape of DaNewsroom took more than a week to finish and resulted in about 3.59 million downloaded articles, a reduction of 26% compared to number URLs initially provided. These lost articles

[11] da.wikipedia.org/wiki/Aviser_i_Danmark#Landsd%C3%A6kkende_dagblade

[12] web.archive.org/cdx/search/cdx

[13] *[...]&collapse=url*

[14] *[...]&filter=statuscode:200&filter=mimetype:text/html*

[15] github.com/internetarchive/wayback/tree/master/wayback-cdx-server

[16] github.com/lil-lab/newsroom

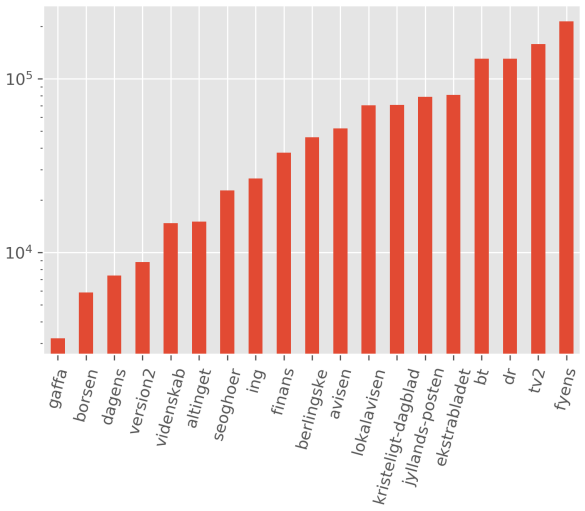

Figure 1: Article count (log-scale) in for each domain name in DaNewsroom.

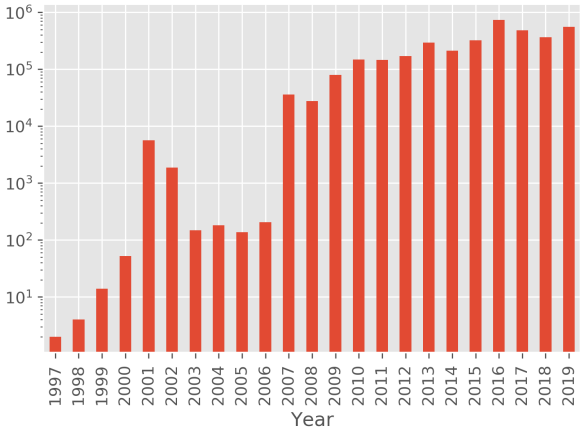

Figure 2: Distribution of articles across years for each collected site in DaNewsroom. The y-axes is plotted in log-scale to highlight the low presence of articles in the late 90's and early 2000's.

may be explained by server errors from the Wayback Machine which are caused by either snapshots not existing or especially lengthy request time-outs.

3.4 Extraction

For extracting samples from the downloaded articles we employ the newsroom-extract command-line tool from the Newsroom package. The package uses Readability[17] to retrieve the main article content and title, and uses SpaCy (Honnibal and Montani, 2017) for tokenisation to compute metrics of compression, coverage and density. The summaries are extracted if there is at least one out three metadata tags: *og:description*, *twitter:description* or *description*. When extracting, we discovered that Danish websites appear not to have embraced, or at least have been slow to adapt to the semantic web metadata tags for summaries. Tags are often present, but contain either empty strings, or site-wide descriptions that are not specific to the article at hand. As shown in Figure 2, there is a corresponding lack of of older articles in the dataset.

[17]pypi.org/project/readability-lxml

Since the Newsroom package is intended for English, we clone the repository and modify it to support multiple languages and in particular Danish tokenisation during extraction.

4 Document-Summary Descriptive Measures

To explore the quality and extractiveness of summaries with respect to documents, Grusky et al. (2018) carried out a series of measurements over *extracted fragments*: greedy n-gram overlaps between an article body and reference summary: coverage, density and compression. We present the definitions used by Grusky et al., and apply these same measures to DaNewsroom. We then propose using these measures as an automatic tool for identifying high-quality article-summary pairs.

Let (A, S) be a instance pair of an article $A = (a_1, a_2, .., a_n)$ and a summary $S = (s_1, s_2, .., s_m)$ consisting of tokens a_i and s_i respectively. And let $|A| := n$ and $|S| := m$.

Extractive Fragments. The set of *extractive fragments* $F(A, S)$ is the set of longest common sequences of tokens in A and S.

Coverage. Coverage measures the extractiveness of a summary–the extent to which the sequences of extractive fragments (the article) covers the the summary itself. As extractiveness increases, coverage tends towards 1. Conversely, as abstractive-ness increases and novel words are introduced, coverage tends towards 0.

$$\textsc{Coverage}(A, S) = \frac{1}{|S|} \sum_{f \in F(A,S)} |f| \tag{1}$$

The next measure takes this into consideration.

Density. Density is identical to coverage, except that the length of fragments (in the summary) is squared. This results in a measure that scores higher for summaries that contain long extractive fragments. If an abstractive summary contains random words from the article, it will also score high in coverage despite being abstractive. By contrast, because the extractive fragments are short, density will indicate extractive-ness.

Thus, combining density with coverage allows one to identify summaries that are mixed extractive and abstractive (so-called "mixed summaries") that compose abstractive-like summaries from short sequences of text found in the article.

$$\textsc{Density}(A, S) = \frac{1}{|S|} \sum_{f \in F(A,S)} |f|^2 \tag{2}$$

Compression. Compression expresses the compression rate of tokens between the article and the summary: the summary to document length ratio.

$$\textsc{Compression}(A, S) = |A|/|S| \tag{3}$$

5 Removing Low Quality Articles

After scraping and and extracting (Section 3), we are left with 3.6 million articles, of which the majority we expect to be of poor quality (given, in particular that Grusky et al. (2018) retained only 1.3 million of the original 100+ million articles). We introduce a few robust high-recall techniques to detect better quality instance pairs and improve the overall quality of DaNewsroom.

- First we removed articles having either empty summaries or article bodies. The portion of empty summaries are 12.1%, and 6.3% of the articles contained an empty body. We observe some overlap with union of the two being 15.5%.
- Secondly, we filter out articles where summaries and bodies are non-unique: if the entire summary or article body is present in more than one document we exclude it from the dataset. This constitutes 45.4% and 31.3% respectively, with a combined presence in almost half of all articles (49.9%).
- Third, we filtered document-summary pairs where the summary was of longer, equal or just slighter shorter length of the article body. Specifically, we filter out articles where $\text{COMPRESSION}(A, S) < 1.5$. Future work could consider further tuning (and increase) of this threshold. We opted err cautiously (for high recall) and keep possibly less interesting samples in the dataset that future work can then filter out, rather than filter out perfectly valid samples (false negatives).

Table 2 summarises the reduction in document-summary pairs across the various stages of filtering. The result of these steps is DaNewsroom.

STAGE	COUNT	% REDUCTION
Filtered URLs	4,859,658	-
Downloaded Articles	3,590,150	73.88%
Post Extraction	3,578,679	73.64%
Basic Filtering	1,175,238	24.18%
Compression Cut-off	1,132,734	23.31%

Table 2: Article filters and the percentage of documents after the entire dataset.

6 Analysis of Measures over DaNewsroom

The above document-summary descriptive measures provide us with a feasible way to ascertain the "extractiveness" or "abstractiveness" of article-summary pairs. In Grusky et al. (2018) there is an emphasis on the signal expressed by the combination of *coverage* and *density* which is displayed in a bivariate plot. We generate a similar plot for DaNewsroom, in Figure 3. In addition we present the same density plot with an increased threshold of $\text{DENSITY}(A, S) < 50$ in Figure 4. This new plot represents 98.6% of DaNewsroom in contrary to that of $\text{DENSITY}(A, S) < 5$ (Figure 3) representing only 43.4% of DaNewsroom.

From Figure 4 we are able to see two clusters of articles. The top-right cluster is composed almost entirely of long extractive summaries: long extractive summaries will have high density. In the bottom left cluster, summaries contain longer spans, though not entire sentences, from the article body. Upon manual inspection of samples this cluster appears to be of particularly high quality.

In Figure 5 we see the compression distribution in DaNewsroom. Recall that compression represents to which extent the summary compresses the article body (token-wise). We observe that summary compression is distributed mainly below 20 followed by a steep long tail. This, together with the mean summary token count (20), tells us that we should not expect particularly long documents.

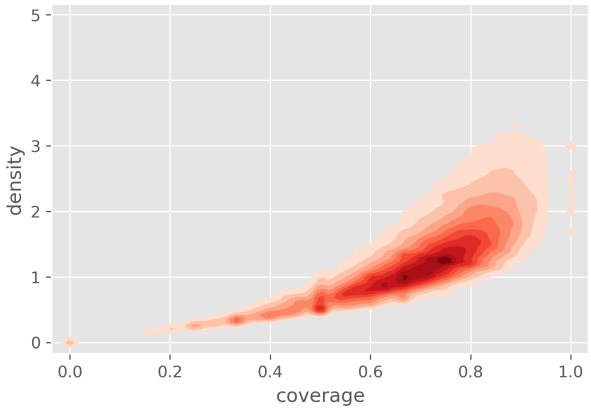

Figure 3: Density distribution where $\text{DENSITY}(A, S) < 5$. The axes are the measures extractive fragment converage (x-axis) and density (y-axis) measures in DaNewsroom.

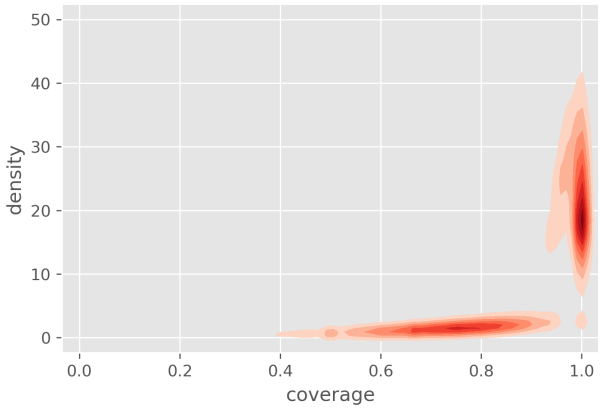

Figure 4: Plot displaying the dataset density between extractive fragment converage (x-axis) and density (y-axis) measures in DaNewsroom where $\text{DENSITY}(A, S) < 50$.

In the appendix, we provide example article-summary excerpts to illustrated the different clusters of the distribution.

7 Getting DaNewsroom

We distribute DaNewsroom as a list of URLs which link to snapshots hosted at The Internet Archive. Together with the modified Newsroom command-line tool, one may reconstruct the dataset. We make the modified Newsroom package and build script freely available[18]. With this approach we hope to encourage extendability as the dataset

[18] github.com/danielvarab/da-newsroom

	DaNewsroom			Newsroom		
	R-1	R-2	R-L	R-1	R-2	R-L
LEAD-3	42.80	35.97	40.11	30.72	21.53	28.65
Oracle	90.13	81.40	90.13	88.46	76.07	88.46
TextRank	26.92	14.95	22.23	22.82	9.85	19.02
ICSISumm	26.83	14.99	22.22	-	-	-

Table 3: F_1-score ROUGE on the test set of Newsroom

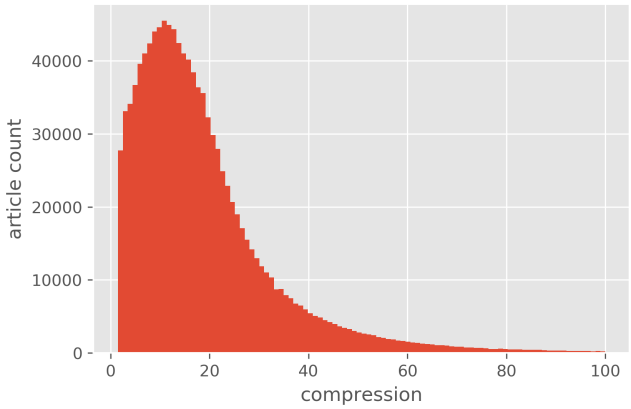

Figure 5: Compression distribution, clipped at 100, in DaNewsroom.

can be easily be extended as well as replicated to other languages following the same methodology.

We split the URLs across sites, grouping URLs by domain and split them into three sets (train/dev/test) over three steps: First we shuffle and split the group into a train, test and dev(elopment) set, with a 80/10/10 ratio. Then we merge all samples belonging to the same split (train, dev or test) and save them to separate files. See Table 4 for descriptive statistics of the dataset and splits.

split		count	$\|A\|$	a $\pm$ s ($\|S\|$)
train	(types)	2,733,973	-	-
	(tokens)	389,008,391	404.8	24.53±12.8
dev	(types)	738,883	-	-
	(tokens)	48,733,808	403.7	24.51±12.6
test	(types)	738,480	-	-
	(tokens)	48,674,409	407.1	24.52±12.7
full	(types)	3,499,762	-	-
	(tokens)	3,146,648	404.9	24.52±12.6

Table 4: Descriptive statistics of tokens and types in the splits of DaNewsroom. Average (a) for articles ($|A|$) and summaries ($|S|$) in addition to standard deviation (s) for summaries are over tokens counts only. The count column is over the set of all article-summary pairs ($|A| + |S|$) in the entire dataset.

8 Baselines

For comparison with future system performance, and since no prior work has been done previously on Danish summarisation, we now introduce report the performance for handful of simple but strong unsupervised baseline models (TextRank, ICSISumm, and Lead-3), together with an oracle extractive model (Fragment Oracle). See Table 5 for an overview of the model performances on DaNewsroom.

8.1 TextRank

TextRank (Mihalcea and Tarau, 2004) is an unsupervised extractive graph-based extractive summarisation system makes use of a version the PageRank algorithm (Page et al., 1999) to importance weight input document sentences for their selection into the output summary. Based on the words of the documents (for nodes) and the lexical similarity (for edges), a text network is formed and words obtain a centrality (of the network) weighting as a measure of their importance, and upon which sentence weighting depends. We use the implementation provided by the Python library Gensim[19] (Řehůřek and Sojka, 2010) which follows the recent extensions proposed by (Barrios et al., 2016). This is the exact system used by (Grusky et al., 2018), except that Gensim does not support custom tokenisation and sentence segmentation. Therefore, for this model, we employ English language tools.

8.2 ICSISumm

ICSISumm (Gillick and Favre, 2009) is an unsupervised summarisation system that generates extractive summaries, by outputting the set of input document sentences that globally and cummulatively contain the most important document concepts (bigrams). Bigram importance is approximated by bigram frequency in the input document set. We use the code associated with the paper[20]. We also include our code extensions in our own repository for reproducibility.[21]

8.3 Lead-3

Lead-3 copies the three first sentences of the article and presents directly them as the summary. The approach takes advantage of the fact that news articles often start with a paragraph that pitches the article. Despite the simplicity of this approach, Lead-3 is one of the strongest baselines for neural automatic summarisation (of online news articles), though it should serve, rather, as a type of lower bound and sanity check during system development.

8.4 Fragment Oracle

We include the Extractive Fragment Oracle as described in (Grusky et al., 2018). This model uses the fragments function $F(A, S)$ and composes a summary by concatenating

[19] radimrehurek.com/gensim/summarisation/summariser.html
[20] github.com/benob/icsisumm
[21] github.com/danielvarab/da-newsroom

	Extractive			Mixed			Abstractive		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
LEAD-3	60.95	59.38	60.69	26.38	13.07	20.31	16.80	3.85	11.57
Oracle	99.36	99.21	99.36	88.72	76.45	88.72	71.24	46.78	71.24
TextRank	34.07	23.00	29.47	22.10	8.15	16.66	15.11	2.73	10.73
ICSISumm	34.20	23.54	29.40	20.71	7.33	14.90	14.72	2.54	9.97

Table 5: F_1 ROUGE scores on three subsets of the development set. Extractive, mixed and abstractive are binned categories of the density measure. These cut-off values are taken directly from (Grusky et al., 2018).

the returned fragments of the function. This model, therefore, has access to the reference summary and acts as a upper bound for extractive methods. Surpassing the performance of this model would require an abstractive summarisation approach.

We note that the Fragment Oracle approach does not attempt to repair or rearrange fragments in any way, and merely concatenates the fragments in the order they are returned by $F(A, S)$. This often results in incoherent summaries that still score high ROUGE scores.

9 Baseline Evaluation

We use the standard ROUGE-1 (R-1), ROUGE-2 (R-2) and ROUGE-L (R-L) for evaluation, as it have been shown to be the ROUGE measures that are the most correlated measures with human judgements of summarisation (Hong et al., 2014). We leverage the Newsroom Python package which follows the default parameters for ROUGE. For word tokenisation for all systems, except TextRank, we use the Danish SpaCy tokeniser. TextRank uses the English tokeniser included in Gensim as it does not support custom tokenisation. We don't employ any lemmatisation. For sentence segmentation we use the English Gensim sentence segmenter.

For PageRank and ICSISumm, we must also input an output summary budget parameter. For PageRank, we employ a grid search, optimising for R-1, on the development set and find 35 to give us the best results. We adopt the same budget for the ICSISumm experiments.

In Table 3 we see the scores for all four models on the test set. We include scores reported in the Newsroom paper for relative comparison. Relatively, results on DaNewsroom follow the same trend as those reported on the Newsroom dataset. LEAD-3 and Oracle significantly outperform the other summarisation systems across both datasets. TextRank and ICSISumm are almost indistinguishable on all three ROUGE metrics on DaNewsroom, only differentiating by at most 0.1 absolute percentage point.

In Table 5 we see the scores produced by the four presented baseline models on three subsets of the development set. These subsets are binned categories of Density(A, S) values. We follow the cut-off values directly from (Grusky et al., 2018) of 1.5 and 8.1875, where $abstractive$ = Density(A, S) $<=$ 1.5, $mixed$ = 1.5 $>$ Density(A, S) $>$ 8.1875, and $extractive$ = Density(A, S) $>$ 8.1875. The distribution of these bins is given in Figure 6. Again, LEAD-3 and Oracle outperform the two remaining models by a large margin. On the extractive subset LEAD-3 jumps to an F_1-score of 60 across all ROUGE metrics, and our Oracle model pushes 100, due to the matching extractive character of the method. TextRank and ICSISumm both, as expected, improve in performance, most likely due to their being are purely extractive methods. Equally expected, these latter models score lower on the two other subsets: *mixed* and *abstractive*.

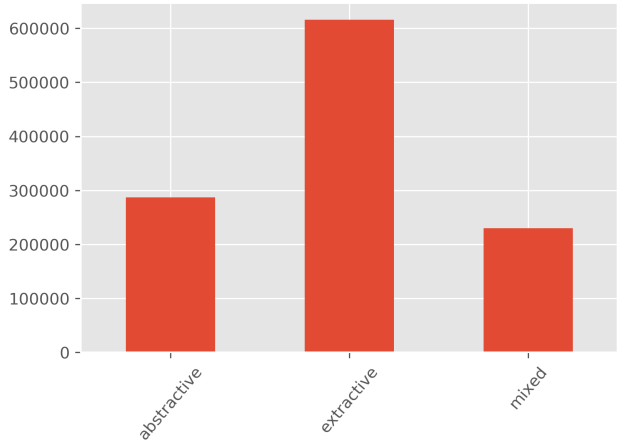

Figure 6: Distribution of binned categories (extractive, abstract or mixed) in DaNewsroom. About half of samples are categorised as extractive, while the remaining half is a mixture of abstractive and mixed samples.

10 Concluding remarks

We have presented the first Danish automatic summarisation dataset, which is also the first large-scale non-English for this task, together with baseline performance over the test sets. Dataset development for automatic summarisation systems has indeed been notoriously English-oriented. However, system performance problems related to automatic performance metrics required to gauge the performance of any realistic development of these systems for English itself, let alone other non-English languages is still problematic (Schluter, 2017), and could impede making the actual business case of automatic summarisation development. With this dataset, we are finally able to gain some understanding of the true performance of currently developed systems outside of the English arena. More over, we have provided explicit guidelines and tools to apply the same method to further languages.

11 Bibliographical References

Barrios, F., López, F., Argerich, L., and Wachenchauzer, R. (2016). Variations of the similarity function of textrank for automated summarization. *CoRR*, abs/1602.03606.

Chen, D., Bolton, J., and Manning, C. D. (2016). A thorough examination of the CNN/daily mail reading comprehension task. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2358–2367, Berlin, Germany, August. Association for Computational Linguistics.

Cheng, J. and Lapata, M. (2016). Neural summarization by extracting sentences and words. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 484–494, Berlin, Germany, August. Association for Computational Linguistics.

Gillick, D. and Favre, B. (2009). A scalable global model for summarization. In Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing, pages 10–18, Boulder, Colorado, June. Association for Computational Linguistics.

Grusky, M., Naaman, M., and Artzi, Y. (2018). Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. 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 708–719, New Orleans, Louisiana, June. Association for Computational Linguistics.

Hong, K., Conroy, J., Favre, B., Kulesza, A., Lin, H., and Nenkova, A. (2014). A repository of state of the art and competitive baseline summaries for generic news summarization. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), pages 1608–1616, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Honnibal, M. and Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Lee, K., He, L., and Zettlemoyer, L. (2018). Higher-order coreference resolution with coarse-to-fine inference. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 687–692, New Orleans, Louisiana, June. Association for Computational Linguistics.

Li, Q., Li, T., and Chang, B. (2016). Discourse parsing with attention-based hierarchical neural networks. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 362–371, Austin, Texas, November. Association for Computational Linguistics.

Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing order into text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pages 404–411, Barcelona, Spain, July. Association for Computational Linguistics.

Moritz Hermann, K., Kocisky, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., and Blunsom, P. (2015). Teaching machines to read and comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama and R. Garnett (Eds.), *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., pp. 1693–1701.

Nallapati, R., Zhou, B., dos Santos, C., GuÌ‡lçehre, 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.

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, AKBC-WEKEX '12, pages 95–100, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nenkova, A. and Passonneau, R. (2004). Evaluating content selection in summarization: The pyramid method. In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004, pages 145–152, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Nguyen, K. and Daumé III, H. (2019). Global voices: Crossing borders in automatic news summarization. In Proceedings of the 2nd Workshop on New Frontiers in Summarization, pages 90–97, Hong Kong, China, November. Association for Computational Linguistics.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Parker, R., Graff, D., Kong, J., Chen, K., and Maeda, K. (2011). English gigaword fifth edition. Linguistic Data Consortium.

Řehůřek, R. and Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pages 45–50, Valletta, Malta, May. ELRA. http://is.muni.cz/publication/884893/en.

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.

Schluter, N. and Martínez Alónso, H. (2016). In Actes de la conférence conjointe JEP-TALN-RECITAL, volume 2, pages 349–354.

Schluter, N. (2017). The limits of automatic summarisation according to ROUGE. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 41–45, Valencia, Spain, April. Association for Computational Linguistics.

Appendix

In this appendix, we present five examples from DaNewsroom. Each article-summary pair displays different properties with respect to the measures described in Section 4. Each example belongs to one of binned categories (extractive, abstract or mixed) described in 9.

Figure 7 shows a *mixed summary*. We observe that the summary consists almost entirely of spans and tokens from the article body, but not an entire sentence. This is still an abstractive summary, but illustrates the special case where the summary to some degree is composed of spans from the article.

Summary: Windows 10 er på gaden, og har du Windows 7 eller en nyere version af styresystemet på din pc eller tablet, kan du opgradere gratis.

Start of Article: Nu er det længe ventede Windows 10 ankommet. 29. juli udkom det seneste i rækken af Microsofts styresystemer. Har du Windows 7, Windows 8 eller Windows 8.1 på din pc, kan du hente og installere det nye styresystem. Gratis.
Du har et år (fra 29. juli) til at tage beslutningen, før der skal betales en opgradering, men på Datatid TechLife kan vi allerede nu fornemme, at Windows 10 bliver godt. [...]

Figure 7: Mixed summary which combines extractive spans to produce an abstractive descriptions of the article.

Figure 8 provides an example of an extractive summary. Here the exact summary is contained as the first sentence in the article. This is a well known tendency in news articles and provides evidence that the LEAD-3 baseline is well-motivated.

Summary: En international lufthavn i Florida har fredag aften været ramme for en skudepisode

Start of Article: En international lufthavn i Florida har fredag aften været ramme for en skudepisode
Mindst fem mennesker har mistet livet i en skudepisode i den internationale lufthavn Fort Lauderdale-Hollywood, der ligger i staten Florida lige nord for byen Miami. Det oplyser det lokale politi. [...]

Figure 8: Extractive summary which is the first sentence in article body.

In Figure 9 we observe a challenging example of an abstractive summary which compresses and selectively filters information contained throughout the article. Details and names are removed, but the general theme of the article remains.

Figure 10 shows another abstractive summary that deviates entirely from the form of the article. This is also a prime example of an abstractive summary. The summary compresses the the salient information contained in the three first sentences into a single information rich sentence.

Summary: EU skal stå sammen om at sende de afviste asylansøgere hjem, der i dag bliver hængende uden lov til at være her. Det skal ske ved at love dem en bedre fremtid i hjemlandet, mener Venstre. DF og S kalder planen er urealistisk og ineffektiv.

Start of Article: Hjælp til at købe 100 kyllinger. Et bidrag til at købe en taxi. Eller måske støtte til at etablere en mekanikerbiks? EU bør have et fælles og fast finansieret økonomisk program, som kan lokke afviste asylansøgere uden lovligt ophold i EU til at rejse hjem og starte forfra. Det mener Venstres europaparlamentariker, Morten Løkkegaard, og udviklingsminister Ulla Tørnæs i et fælles udspil til en afrika- og migrationspolitik, hvor det, de kalder »hjemsendelsesstøtte,« er et centralt element. [...]

Figure 9: Abstractive summary that effectively summarises the salient information expressed through a long article.

Summary: Skuespilleren, som sprang fra rollen som Martin Rohde i dramaserien, følger ikke med i de nye afsnit

Start of Article: I de to første sæsoner af dramaserien 'Broen' havde svenske Saga Norén kollegialt seskab af danske Martin Rohde, spillet af Kim Bodnia. Men før tredje sæson sprang den danske skuespiller fra, da han var utilfreds med, hvordan hans rolle udviklede sig. I stedet er Thure Lindhardt blevet Sagas nye makker i DR1's hitserie. [...]

Figure 10: Abstractive summary which compresses the first five sentences into a single sentence.

Finally, Figure 11 shows another extractive summary where the entire summary may be found in the article. This time, it is as a single sentence, found as the second sentence in the article.

Summary: I Frankrig har en ulykke i forbindelse med et rallyløb kostet to personer livet, mens 15 personer blev kvæstet.

Start of Article: Ulykken skete i en lille by nær Toulon i Sydfrankrig. I Frankrig har en ulykke i forbindelse med et rallyløb kostet to personer livet, mens 15 personer blev kvæstet. Ifølge øjenvidner mistede føreren af bilen kontrollen i et sving. Bilen fortsatte med høj fart ind i en gruppe tilskuere. Den ene dræbte var en tilskuer, mens den anden var official ved løbet. Blandt de 15 kvæstede var mange børn
Føreren af bilen slap med lettere kvæstelser. Ulykken skete i en lille by nær Toulon i Sydfrankrig. [...]

Figure 11: Extractive summary that uses entire sentence from article body as summary.