ARTEMIS: A Novel Annotation Methodology for Indicative Single Document Summarization

Rahul Jha*, Keping Bi†, Yang Li*, Mahdi Pakdaman*
Asli Celikyilmaz*, Ivan Zhiboedov‡, Kieran McDonald*
* Microsoft Corporation
† Umass Amherst
‡ Facebook Inc

Abstract

We describe ARTEMIS (Annotation methodology for Rich, Tractable, Extractive, Multi-domain, Indicative Summarization), a novel hierarchical annotation process that produces indicative summaries for documents from multiple domains. Current summarization evaluation datasets are single-domain and focused on a few domains for which naturally occurring summaries can be easily found, such as news and scientific articles. These are not sufficient for training and evaluation of summarization models for use in document management and information retrieval systems, which need to deal with documents from multiple domains. Compared to other annotation methods such as Relative Utility and Pyramid, ARTEMIS is more tractable because judges don't need to look at all the sentences in a document when making an importance judgment for one of the sentences, while providing similarly rich sentence importance annotations. We describe the annotation process in detail and compare it with other similar evaluation systems. We also present analysis and experimental results over a sample set of 532 annotated documents.

1 Introduction

Given an input source document, summarization systems produce a condensed summary which can be either informative or indicative. Informative summaries try to convey all the important points of the document (Kan et al., 2002, 2001b), while indicative summaries hint at the topics of the document, pointing to information alerting the reader about the document content (Saggion and Lapalme, 2002). An informative summary aims to replace the source document, so that the user does not need to read the full document (Edmundson, 1969). An indicative summary, on the other hand, aims to

Original Document

(1) This content should be viewed as reference documentation only, to inform IT business decisions . . .
(2) Microsoft employees need to stay aware of new company products, services, processes, and personnel-related developments in an organization that provides them . . .
(3) The SMSG Readiness team at Microsoft developed a suite of applications that delivers training and information to Microsoft employees according to employee roles . . .
(4) Microsoft Information Technology (Microsoft IT) is responsible for managing one of the largest Information Technology (IT) infrastructure environments in the world.
(5) It consists of 95,000 employees working in 107 countries worldwide.
(6) The Sales, Marketing, and Services Group (SMSG) at Microsoft is responsible for servicing the needs of Microsoft customers and partners.
(7) It is essential that these 45,000 employees remain informed about products and services within their areas of expertise and, in turn, to educate and inform . . .
(8) The SMSG Readiness (SMSGR) team at Microsoft is responsible for ensuring that SMSG employees have all of the tools and knowledge they require to deliver . . .
(. . . document truncated)

Summary 1

(2) Microsoft employees need to stay aware of new company products, services, processes, and . . .
(3) The SMSG Readiness team at Microsoft developed a suite of applications that delivers training and . . .
(4) Microsoft Information Technology (Microsoft IT) is responsible for managing one of the largest . . .

Summary 2

(3) The SMSG Readiness team at Microsoft developed a suite of applications that delivers training and . . .
(6) The Sales, Marketing, and Services Group (SMSG) at Microsoft is responsible for servicing the needs of . . .
(8) The SMSG Readiness (SMSGR) team at Microsoft is responsible for ensuring that SMSG employees have . . .

Summary 3

(2) Microsoft employees need to stay aware of new company products, services, processes, and . . .
(4) Microsoft Information Technology (Microsoft IT) is responsible for managing one of the largest . . .
(8) The SMSG Readiness (SMSGR) team at Microsoft is responsible for ensuring that SMSG employees have . . .

Figure 1: One of the documents from our web-crawled sample annotated dataset along with indicative summaries annotated by three different judges. The sentence numbers in round brackets are not in the original document but are added here for readability. Summary sentences are truncated for readability as well.

[†] Work done while an intern at Microsoft.

[‡] Work done while an employee of Microsoft.

Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems (Eval4NLP), pages 69–78,
November 20, 2020. ©2020 Association for Computational Linguistics

help the user decide whether they should consider reading the full document (Kan et al., 2002).

The content of indicative summaries can be composed in several ways. For example, it can contain sentences extracted from the source document which relate to its main topic (Barzilay and Elhadad, 1997; Kupiec et al., 1995), generated text describing how a document is different from other documents (Kan et al., 2001a), topic keywords (Hovy and Lin, 1997; Saggion and Lapalme, 2002) as well as metadata such as length and writing style (Nenkova and McKeown, 2011).

Document management systems such as Google Docs, Microsoft OneDrive and SharePoint and Dropbox can use indicative summaries to help their users decide whether a given document is relevant for them before opening the full document. Indicative summaries can also be used in information retrieval systems as previews for documents returned in search results. Document summarization systems deployed in these real-world systems need to be able to summarize documents from a wide variety of domains.

However, existing summarization datasets are highly domain-specific, with a majority of them focusing on news summarization (Nallapati et al., 2016; Grusky et al., 2018; Sandhaus, 2008; Graff et al., 2003). One of the reasons for this bias towards news summarization is the availability of naturally occurring summaries for news, which makes it easier to create large-scale summarization datasets automatically by scraping online sources. Apart from the domain bias, they are also susceptible to noise which can affect upto 5.92% of the data (Kryscinski et al., 2019).

In order to train and evaluate multi-domain summarization models for use in document management systems, we need to build representative datasets geared towards this use case. Towards this goal, we present ARTEMIS (Annotation methodology for Rich, Tractable, Extractive, Multi-domain, Indicative Summarization), a hierarchical annotation process for indicative summarization of multi-domain documents. Figure 1 shows a sample document crawled from the web with three annotated summaries obtained using ARTEMIS.

ARTEMIS's hierarchical annotation process allows judges to create indicative summaries for long documents through divide-and-conquer. Judges successively summarize larger and larger chunks of a document in multiple stages, at each stage reusing sentences selected previously. The hierarchical process means that judges only look at a small set of sentences at each stage.

Compared to previous annotation methods, where judges need to consider all the document sentences together when building a summary (Tam et al., 2007) or create expensive semantic annotations (Nenkova and Passonneau, 2004), ARTEMIS is a low-cost annotation approach that produces rich sentence importance annotations. Judges are able to use ARTEMIS to annotate documents averaging 1322 words (77 sentences) in 4.17 minutes on average, based on an initial sample of annotation tasks. This is almost twice the length of documents in summarization datasets such as CNN/Dailymail at 766 words (Nallapati et al., 2016) and NEWSROOM at 659 words (Grusky et al., 2018).

ARTEMIS's annotation process aims at selecting a set of sentences that contain relevant information about the main topics of a document rather than conveying all the relevant information in a document. Given this, summaries annotated by ARTEMIS are indicative in nature and suited for document management and information retrieval systems, where they can be used as part of document preview to help a user decide whether a document is relevant for them.

The rest of this paper is organized as follows. Section 2 describes the annotation process in detail and Section 3 relates our method to previous annotation methods for summarization. Section 4 presents a number of analyses characterizing the ARTEMIS annotation process in terms of label distribution and judge agreement by using a sample annotated document set. Section 5 presents evaluation results for a set of baseline summarization models on the sample annotated document set. Finally, Section 6 presents some concluding remarks and points to future work.

2 Annotation Methodology

Figure 2 shows a high-level diagram representing the annotation process for ARTEMIS. Given a document as input, the preprocessing step consists of first dividing the document into sections, each of which is further divided into paragraphs. The section and paragraph boundaries are computed based on a set of heuristics that depend on signals like explicit section headers as well as constraints on the number of sentences shown at each screen.

The hypothetical document in Figure 2 is di-

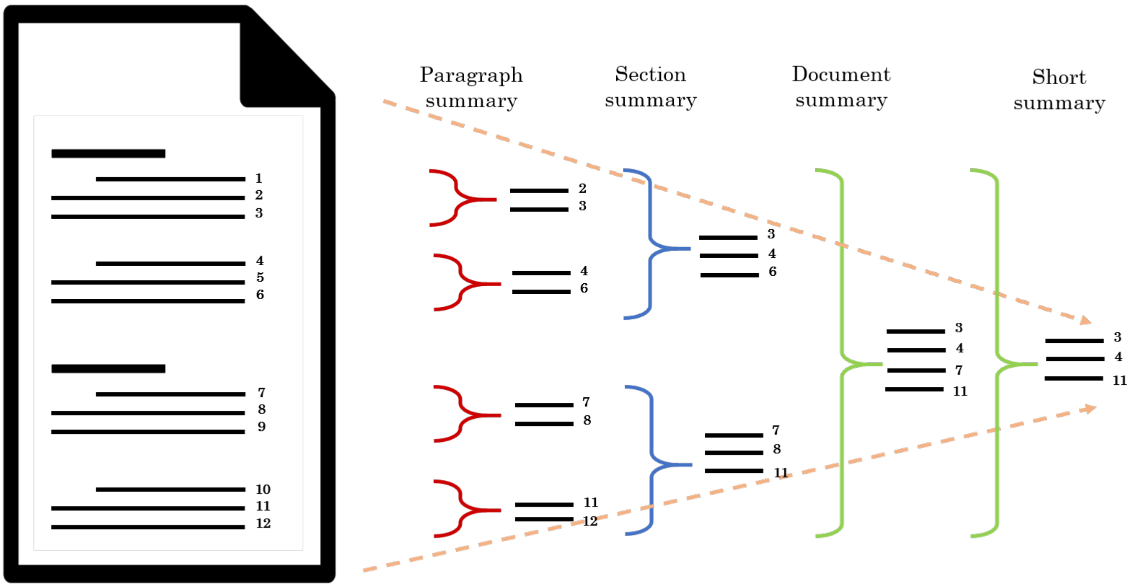

Figure 2: A schematic of ARTEMIS annotation process. A document is divided into sections and paragraphs. The judges summarize paragraphs, sections and the document hierarchically, at each step using sentences selected at the previous step.

vided into two sections with two paragraphs each. The first section contains sentences $\{1 \mathrel{..} 6\}$, with two paragraphs containing sentences $\{1 \mathrel{..} 3\}$ and $\{4 \mathrel{..} 6\}$ respectively. The second section contains sentences $\{7 \mathrel{..} 12\}$, again with two paragraphs containing sentences $\{7 \mathrel{..} 9\}$ and $\{10 \mathrel{..} 12\}$.

A salient sentence is defined as a sentence that includes a main concept or idea for summarizing the text, or a fact or an argument emphasized by the author[*]. Several example sentences are provided in the judge guidelines to help them distinguish salient sentences from non-salient sentences. At a high-level, the judges are trained to select sentences that allow a reader to decide whether to read the full document or not.

To summarize the document, judges proceed in a bottom-up manner starting from paragraphs (left-to-right in Figure 2). A judge is first asked to summarize each paragraph in a section by selecting a few salient sentences. A minimum number of sentences are required for each paragraph-summary [†]. Once a paragraph has been summarized, the annotation continues to the next paragraph till paragraph-level summaries are created for all the paragraphs in a section. For the document in Figure 2, the judge selected sentences $\{2, 3\}$ for the first paragraph and sentences $\{4, 6\}$ for the second paragraph.

Once all the paragraphs in a section are summarized, the judge is asked to create a summary for the entire section. However, the judge doesn't have to look at all the sentences in the section to build the section-level summary. Instead, they only select from the set of sentences previously selected to summarize the paragraphs of the section. For example, for summarizing the first section in Figure 2, the judge only needs to select from the set of sentences $\{2, 3, 4, 6\}$, instead of the entire set of sentences $\{1 \mathrel{..} 6\}$ that comprise the section. In the example, the judge decided to use the sentences $\{3, 4, 6\}$ for summarizing the first section.

Once a section is summarized, the annotation proceeds to the next section in a similar manner. Once all the sections of a document are summarized, the judge is asked to build the document summary by selecting from sentences that they had previously selected to build the section-level summaries. In Figure 2, the judge selected sentences $\{3, 4, 7, 11\}$ for the document level summary. Finally, the judge is asked to build a short summary for the document by selecting three most salient sentences from their document-level summary.

ARTEMIS's hierarchical annotation process considerably reduces the cognitive load on the judges. By reusing judgements made at previous steps, judges are able to successively summarize long documents by divide-and-conquer. For creating

[*] Authors can emphasize sentences either through formatting or discourse cues.

[†] The judge can also mark incomplete and grammatically incorrect sentences as defective, which are not counted when computing the minimum threshold for paragraph-summary.

Document Sentences	**#Para**	**#Sec**	**#Doc**	**#Short**
(1) This content should be viewed as reference documentation only, to inform IT . . .	0	0	0	0
(2) Microsoft employees need to stay aware of new company products, services, processes, and personnel-related developments in an organization that provides . . .	2	2	2	2
(3) The SMSG Readiness team at Microsoft developed a suite of applications that delivers training and information to Microsoft employees according to employee . . .	4	4	4	3
(4) Microsoft Information Technology (Microsoft IT) is responsible for managing one of the largest Information Technology (IT) infrastructure environments in the world.	5	4	2	2
(5) It consists of 95,000 employees working in 107 countries worldwide.	0	0	0	0
(6) The Sales, Marketing, and Services Group (SMSG) at Microsoft is responsible for servicing the needs of Microsoft customers and partners.	3	1	1	1
(7) It is essential that these 45,000 employees remain informed about products and services within their areas of expertise and, in turn, to educate and inform . . .	1	0	0	0
(8) The SMSG Readiness (SMSGR) team at Microsoft is responsible for ensuring that SMSG employees have all of the tools and knowledge they require to deliver . . .	4	3	2	2

Table 1: Detailed view of the annotation for the web-crawled document shown in Figure 1. Against each sentence, we show the number of judges that selected the sentence at paragraph, section, document and short summary stage.

the document-level summary in the hypothetical example in Figure 2, the judge only needs to look at the 6 sentences $\{3, 4, 6, 7, 8, 11\}$ selected for the two section-level summaries, instead of having to go over the entire set of 12 sentences.

Table 1 shows a more detailed view of the annotation for an actual document annotated through ARTEMIS with five judges (This is the same document that was used in Figure 1). For each of the first eight sentences in the document, it shows the number of judges that selected the sentence at paragraph, section, document and short summary stage. This table gives an insight into the kind of information available from the annotation.

Sentences *(1)* and *(5)* were deemed by every judge as not salient. Sentence *(3)* was selected by four judges as salient up to document-summary level, but one of the judges dropped it at short-summary level. Similarly, sentence *(4)* was selected at paragraph-summary level by five judges, but only two judges kept it till the document and short-summary level. In Section 4, we present statistics on a sample annotated document set that characterize the annotation process in more detail.

3 Related Work

We now compare ARTEMIS with existing summarization evaluation methods. We start with discussing Relative Utility, which is most related to our methodology, and describe how ARTEMIS obtains similar judgments, but with a light-weight process where judges don't need to look at the entire input document when annotating a sentence. Following this, we discuss DUC evaluations, ROUGE and the Pyramid method. Finally, we discuss some of the recent trends in summarization evaluation.

3.1 Relative Utility

Tam et al. (2007) introduce Relative Utility (RU) as an evaluation metric to account for Summary Sentence Substitutability (SSS) problem in co-selection metrics. Co-selection metrics are evaluation metrics for extractive summarization that depend on text unit overlap with ideal reference summaries created by judges. The SSS problem arises because the judges only provide information about the sentences that they selected for a fixed-length summary. However, other sentences in the document might be equally good candidates for the summary. Human judges often disagree about which are the top n% of the sentences in a document (Mani, 2001).

To address the SSS problem, in RU evaluation judges are asked to assign a utility score to each sentence in a document on a scale of 0 to 10. Given these utility scores, the score for any arbitrary extractive summary can be computed based on the utility of the sentences in the summary.

In RU, to assign the utility score to a sentence in the document, a judge needs to compare the sentence with every other sentence in the document. This can be difficult for long documents. ARTEMIS is a light-weight process that achieves an approximation of this. By assigning graded importance scores to paragraph, section, document and short summary level labels, we can obtain an approximate utility score for each sentence. For example scores $\{1, 2, 3, 4\}$ could be assigned to sentences selected at paragraph, section, document and short summary level and a score of 0 could be assigned to sentences not selected at any level.

3.2 DUC evaluations and ROUGE

DUC (Document Understanding Conferences) were a series of conferences run to further progress in summarization. DUC 2001-2004 focused on single and multi-document summarization (Dang, 2005). In DUC evaluation for summary content, first a single human judge creates a model summary for each document. The model summary is split automatically into content units. For evaluating a system generated summary, a human judge compares the sentences in the system summary with model content units and estimates the fact overlap.

The use of a single model summary in DUC evaluations raised concerns in the research community and led to the proposal of Pyramid evaluation, which we describe in Section 3.3. Lin (2004a) concluded that given enough samples, the use of single model summaries was valid, but using multiple model summaries increased correlation with human judgments.

In later years, DUC experimented with ROUGE (Lin, 2004b), an automatic metric for summary evaluation that uses n-gram co-occurrence statistics for scoring system generated summaries against the model summaries. ROUGE is the standard automatic evaluation method used in recent summarization evaluations, which we describe in Section 3.4.

In ARTEMIS, the sentences selected by judges for document or short-level summary can be used as model summaries for ROUGE evaluation, as we demonstrate in Section 5. In addition, the labels for sentences at different summary levels could be used to train a pair-wise sentence ranking system such as LambdaMart (Burges, 2010) or come up with more refined evaluation metrics.

3.3 Pyramid evaluation

Nenkova and Passonneau (2004) introduced Pyramid method as a more reliable method for summary evaluation by incorporating the idea that no single best model summary exists. Given a set of human-generated model summaries for a document, the Pyramid method starts by manually identifying Summary Content Units (SCUs) in the model summaries. A SCU represents a single unit of information (e.g. “Two men were indicted”) which can have different surface realizations in different summaries (e.g. “Court indicted two men”, “Two men have been indicted”).

The weight of an SCU is the number of model summaries it appears in. Thus, an SCU appearing in five model summaries has a higher weight than an SCU appearing in three model summaries. Given the SCU inventory over all model summaries, the Pyramid score of a system generated summary is obtained based on the number and weights of the SCUs in the summary. Nenkova and Passonneau (2004) observe that the number of SCUs grows as the number of model summaries increases, confirming a similar observation by van Halteren and Teufel (2003), supporting the claim that different judges deem different facts as important.

Finding SCUs in model summaries and then matching them to system summaries is an expensive semantic judgment task. Once created, the SCU inventory can be used to assign an importance weight to any sentence in a system generated extractive summary based on the weights of SCUs in it. Our methodology provides a cheaper method for assigning importance weight for each sentence in a document. In ARTEMIS, multiple judges select each sentence for multiple summaries at paragraph, section, document, and short-summary levels. These judgments provide a low-cost way of obtaining an importance weight for a sentence, without expensive SCU annotation.

3.4 Recent Trends in Summarization Evaluation

Recent summarization evaluations are done using large scale datasets collected automatically from the web. Most of these datasets are from the news domain, including CNN/DailyMail (Nallapati et al., 2016), NEWSROOM (Grusky et al., 2018), New York Times (Sandhaus, 2008) and Gigaword (Rush et al., 2015). Some of the other domains investigated are scientific articles (Cohan et al., 2018), patents (Sharma et al., 2019), and Reddit stories (Kim et al., 2019).

Datasets built from naturally occurring summaries found online tend to focus on domains for which manually written summaries are easily available such as news and scientific articles. These datasets are not sufficient for building a multi-domain document summarization application. Additionally, given the nature of data collection, often only a single summary is available for each document. This makes error analysis of individual examples difficult because different judges might deem different information as summary-worthy (Louis and Nenkova, 2013) as discussed in Section 3.3. ARTEMIS provides a methodology for obtaining

Partition	# Sentences	# Documents
Train	19748	266
Dev	11488	138
Test	9898	128

Table 2: Sample dataset used for the data analysis in this paper.

Partition	Average Count Per Document	Average number of sentences
Section	6.92 ± 0.91	12.04 ± 0.72
Paragraph	15.25 ± 2.17	5.46 ± 0.06

Table 3: Average number of paragraphs and sections per document and the average number of sentences in each, along with the 95% confidence interval.

rich summary annotations for open-domain documents with multiple judges.

Summaries collected from online sources are also prone to noise. Kryscinski et al. (2019) manually inspected CNN/DailyMail and NEWSROOM datasets and found that the problem of noisy data affects upto 5.92% of the summaries in different splits. Examples of noise they found include links to other articles and news sources, placeholder texts, unparsed HTML code, and non-informative passages in the reference summaries. In ARTEMIS, such noisy text is excluded from annotation by explicit labeling of defective sentences.

Hardy et al. (2019) proposed a new summarization evaluation approach called HIGHRES, which uses multiple judges to highlight salient information in original documents. Once the highlights are obtained, a system summary can be evaluated manually by asking judges to compare the system summary against highlights, or by a modified ROUGE evaluation that weighs n-grams by the number of times they were highlighted. HIGHRES is complementary to our hierarchical annotation approach and both the methods can be used together for obtaining rich summary annotations.

4 Annotated Data Analysis

We present analysis on a sample dataset of 532 Microsoft Word documents crawled from the web with no domain restrictions, thus creating an open-domain dataset. We extracted the text from the Word documents for our annotation. The annotation framework does not rely on Word document format and can be used to annotate any document for which the raw text can be extracted.

The data was annotated by a set of managed judges who were trained extensively for ARTEMIS annotation process using detailed guidelines and illustrative examples. For additional quality control, we used a set of gold documents annotated by the development team for initial qualification tests for the judges as well as their ongoing evaluation. We divided the sample dataset into train, dev and test partitions, as shown in Table 2. Unless otherwise stated, the statistics presented are computed over the dev partition.

4.1 Distribution Statistics

Table 3 shows how the sentences of a document are divided across paragraphs and sections for the annotations. On an average, there are about 7 sections and 15 paragraphs in each document. The number of sentences in each section averages about 12, while the number of sentences in each paragraph averages about 5. Note that when summarizing a section, a judge has to look at much smaller number of sentences than 12, thanks to the hierarchical annotation process.

To understand where the salient sentences lie for the documents, we divide each document into 10 equally sized bins and plot what fraction of sentences selected for the doc-level summaries lie in each bin. Each bin on an average contains 8.34 ± 0.38 sentences. Figure 3 shows the distribution of sentences selected for the doc-level summaries across the bins. More than 50% of the selected sentences lie in the first bin and more than 90% of the sentences lie in the first five bins. This shows that there is a bias for the summary sentences to be towards the first half of a document. However, the annotators don't form summaries by just selecting the first few sentences, as shown by the poor ROUGE-F1 scores obtained by the Lead-3 baseline in Section 5.

Another characterization of the annotation system can be done based on what fraction of salient sentences selected at each stage make it to the next stage. Table 4 shows this for all the stages of annotation. Looking at the diagonal first, we see that 82.44% of the sentences selected as salient for paragraph-level summaries are also selected for section-level summaries, but only 69.57% of the sentences selected for section-level summaries are selected for document-level summaries. From document-level summaries to short summary level, again 84.57% of the salient sentences are kept. This shows that a larger number of sentences get filtered between the section and document level.

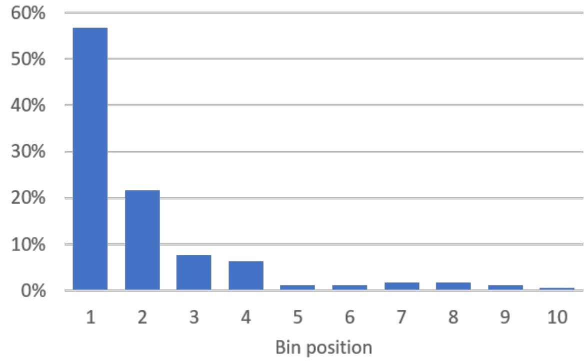

Figure 3: Distribution of sentences selected for doc-level summaries across 10 equally sized bins for each document.

	Section	Document	Short
Paragraph	82.44%	57.36%	48.51%
Section		69.57%	58.84%
Document			84.57%

Table 4: Filtration ratios for salient sentences between different stages. For example, the first row (Paragraph) shows what percentage of sentences selected at paragraph level survive till section, document and short-summary level. Table cells corresponding to filtration between same or out-of-order stages in the pipeline are colored gray.

Overall, only 48.51% of the sentences selected for paragraph-level summaries are used for the final three-sentence short summaries.

4.2 Agreement Statistics

We compute Krippendorff's alpha over the entire annotated document set by treating each of paragraph, section, document and short summary level judgements as ordinal ratings. Across the set of all judges, the Krippendorff's alpha is 0.46. This is consistent with previous findings that summary content selection is a subjective task with moderate agreement (Mani, 2001). Nenkova and Passonneau (2004) report a Krippendorff's alpha of 0.81 for their annotations. However, they measure agreement on the task of assigning SCU's to words, which is a less subjective task than assigning importance to a content unit. They also use a distance metric for computing Krippendorff's alpha that takes into account SCU size, which is not described in detail in their paper.

For additional agreement evaluation, we had 10 documents evaluated by two sets of judges. The first set of judges was comprised of 4 developers involved in the design of ARTEMIS and its guidelines. The second set of judges was comprised of 5 managed judges trained for doing the annotations. For each set of judges, a sentence was considered to be selected for document-level summary if at least 2 judges selected it. Given these judgements, the Kappa score between the two sets of judges was 0.43, which is considered moderate agreement (Landis and Koch, 1977).

#	Paragraph	Section	Document	Short
1	11.2 ± 1.0	9.4 ± 0.8	6.5 ± 0.4	5.4 ± 0.3
2	5.1 ± 0.6	4.1 ± 0.5	2.8 ± 0.3	2.4 ± 0.2
3	2.5 ± 0.4	2.0 ± 0.3	1.5 ± 0.2	1.3 ± 0.2

Table 5: Average number of salient sentences at each stage corresponding to the minimum number of judges needed to mark a sentence as salient (out of a total of five judges) along with 95% confidence intervals.

Table 5 shows the average number of sentences selected at the different annotation levels if we use a minimum of 1, 2, or 3 judges to mark a sentence as salient out of the 5 total judges that annotate each document. We see that with 2 judges, there is agreement for 2.4 sentences for the final short summary, which is restricted to 3 sentences per judge. Even with 3 judges, there is agreement on 1.3 sentences for the final short-summary level.

5 Experiments

We evaluate a number of baseline methods on the sample annotated document set, partitioned into train, dev and test as described in Table 2.

For these experiments, the document-level summary created by each judge for a document is treated as an independent reference summary and we evaluate the candidate summary against all the reference summaries using ROUGE-F scores.

- **Lead-3** baseline selects first three sentences of a document as the summary.
- **Oracle** scores are obtained using a jack-knifed procedure. Reference summary from each judge is considered a predicted summary and evaluated against all the other reference summaries for the document. The Oracle ROUGE score is computed by averaging the scores for all judge summaries.
- **Cheng&Lapata** (Cheng and Lapata, 2016) is an encoder-decoder summarization model where each sentence is first encoded using a CNN (Convolutional Neural Network). These sentence level encodings are then passed

Method	Rouge-1	Rouge-2	Rouge-L
Lead-3	44.94	34.37	43.39
Cheng & Lapata	60.21	49.81	58.62
SummaRunner	63.56	53.57	61.89
Seq2SeqRNN	63.89	54.22	62.36
Oracle	73.28	66.60	72.20

Table 6: Results for different baselines on the test data.

through an RNN (Recurrent Neural Network) to create contextual encodings for each sentence. The encoding for the final sentence of the document is fed into the decoder, which uses another RNN with attention over input sentence encodings to predict the label for each sentence. At each decoding step, the decoder state also depends on the probability of the previous sentence being part of summary.

- **SummaRunner** (Nallapati et al., 2017) uses a hierarchical RNN to compute contextual encodings for each sentence in the input. These encodings are average pooled and passed through a non-linear transformation to create an encoding for the document. In a second pass, a logistic layer makes a binary decision for each sentence based on the sentence encodings, the document representation as well as factors modeling previously selected summary sentences and sentence position.

- **Seq2SeqRNN** is a method introduced in Kedzie et al. (2018) that uses an RNN to encode the input sentences. A separate RNN based decoder is used to transform each sentence into a query vector which attends to the encoder output. The attention weighted encoder output and the decoder GRU output are used together to predict the output label.

We used the code released by Kedzie et al. (2018) for reproducing Cheng&Lapata, SummaRunner and Seq2SeqRNN systems. The ROUGE-F score for each system on the test data is shown in Table 6.

The Lead baseline achieves a ROUGE-1 score of 44.94, which is significantly lower than the other systems as well as the Oracle. This shows that compared to news summarization, selecting the first few sentences is a much weaker baseline for open-domain summarization.

The SummaRunner system does better than Cheng&Lapata, potentially due to its incorporating multiple signals for content, salience, novelty and position. Seq2SeqRNN performs the best, which is consistent with the results reported in Kedzie et al. (2018). There is still a gap between these systems and the Oracle method, which achieves a ROUGE-1 score of 73.28.

6 Concluding Remarks

In this paper, we described ARTEMIS, a novel hierarchical annotation methodology for indicative, extractive summarization. We described the annotation process in detail and compared it with Relative Utility, DUC evaluation methodology, the Pyramid method as well as other recent methods for summary content evaluation. We also presented analysis over a sample annotated dataset to characterize various properties of annotation process such as distribution of salient sentences and judge agreement. Finally, we showed experimental results for a set of baseline summarization systems using the annotated dataset.

Indicative summaries are useful in a number of scenarios involving information triage such as document management and information retrieval systems. However, summarization models for such systems need to be able to summarize documents from multiple domains. Most existing summarization datasets are single-domain and focused towards news, and hence are not sufficient for training and evaluating models for these applications. ARTEMIS provides a low-cost methodology for annotating multi-domain indicative summaries compared to systems such as Pyramid and Relative Utility while producing similarly rich annotations.

ARTEMIS summary annotations contain sentences that provide information about important topics in the document. The summaries are indicative because they do not aim to convey all the important points for a given information need, but instead, give a sense of what topics are covered in the document. The set of annotations in ARTEMIS can be seen as a coarse partitioning between important and non-important sentences in an input document. Thus, models trained on these annotations can also be used as an importance signal in a larger pipeline for creating informative summaries.

References

Regina Barzilay and Michael Elhadad. 1997. Using lexical chains for text summarization. In *In Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 10–17.

Chris J.C. Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. Technical Report MSR-TR-2010-82.

Jianpeng Cheng and Mirella Lapata. 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. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. 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 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Hoa Trang Dang. 2005. Overview of duc 2005. In *In Proceedings of the Document Understanding Conf. Wksp. 2005 (DUC 2005) at the Human Language Technology Conf./Conf. on Empirical Methods in Natural Language Processing (HLT/EMNLP*.

H. P. Edmundson. 1969. New methods in automatic extracting. *J. ACM*, 16(2):264–285.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

Max Grusky, Mor Naaman, and Yoav Artzi. 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*, pages 708–719, New Orleans, Louisiana. Association for Computational Linguistics.

Hans van Halteren and Simone Teufel. 2003. Examining the consensus between human summaries: initial experiments with factoid analysis. In *Proceedings of the HLT-NAACL 03 Text Summarization Workshop*, pages 57–64.

Hardy Hardy, Shashi Narayan, and Andreas Vlachos. 2019. HighRES: Highlight-based reference-less evaluation of summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3381–3392, Florence, Italy. Association for Computational Linguistics.

Eduard H. Hovy and Chin-Yew Lin. 1997. Automated text summarization in summarist. In *ACL 1997*.

Min-Yen Kan, Judith L. Klavans, and Kathleen R. Mckeown. 2002. Using the annotated bibliography as a resource for indicative summarization. In *In Proceedings of LREC 2002, Las*.

Min-Yen Kan, Kathleen R. McKeown, and Judith L. Klavans. 2001a. Applying natural language generation to indicative summarization. In *In Proc. of the EACL Workshop on Natural Language Generation*, pages 1–9.

Min-Yen Kan, Kathleen R. McKeown, and Judith L. Klavans. 2001b. Domain-specific informative and indicative summarization for information retrieval. In *In: Workshop on text summarization (DUC 2001*, pages 1629–1636.

Chris Kedzie, Kathleen McKeown, and Hal Daumé III. 2018. Content selection in deep learning models of summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1818–1828, Brussels, Belgium. Association for Computational Linguistics.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2019. Abstractive summarization of Reddit posts with multi-level memory networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2519–2531, Minneapolis, Minnesota. Association for Computational Linguistics.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. *ArXiv*, abs/1908.08960.

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 95, page 6873, New York, NY, USA. Association for Computing Machinery.

JR Landis and GG Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159174.

Chin-Yew Lin. 2004a. Looking for a few good metrics: Automatic summarization evaluation - how many samples are enough?

Chin-Yew Lin. 2004b. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Annie Louis and Ani Nenkova. 2013. Automatically assessing machine summary content without a gold standard. *Computational Linguistics*, 39(2):267–300.

Inderjeet Mani. 2001. Summarization evaluation: An overview.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, 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*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Ani Nenkova and Kathleen McKeown. 2011. Automatic summarization.

Ani Nenkova and Rebecca Passonneau. 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. Association for Computational Linguistics.

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*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Horacio Saggion and Guy Lapalme. 2002. Generating indicative-informative summaries with sumum. *Computational Linguistics*, 28(4):497–526.

Evan Sandhaus. 2008. he new york times annotated corpus ldc2008t19.

Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213, Florence, Italy. Association for Computational Linguistics.

Daniel Tam, Dragomir R. Radev, and Gunes Erkan. 2007. Single-document and multi-document summary evaluation using relative utility. Technical Report CSE-TR-538-07, University of Michigan. Department of Electrical Engineering and Computer Science.