Literature Review – XCS224u – Automatic Text Summarization

Alex Ezazi
AI Professional Program
Stanford Online
aezazi@niteo.us

Hamilton Link
AI Professional Program
Stanford Online
hamlink@gmail.com

Eyas Taifour
AI Professional Program
Stanford Online
etaifour@me.com

Abstract

Automatic Text Summarization (ATS) is a process that condenses a lengthy text document into a shorter version through automated means, utilizing Natural Language Processing (NLP) techniques. The condensed version emphasizes the crucial information within the original text based on specific criteria. ATS systems are used to tackle the issue of information overload effectively. In this document, we review recent literature in text summarization. We see an opportunity to develop more explicit information representations to augment summary generation as part of a modular pipeline.

1 General Problem

Text summarization is a multi-faceted problem, and this is reflected in a body of literature that addresses one or more of these facets in diverse combinations with varying success. The strategic goal is simple: take some amount of text and produce a shorter version that retains fidelity. Researchers (Brown et al., 1983) have studied successful human summary strategies and have identified some basic rules:

- Include important ideas.
- Delete trivia.
- Delete repeated ideas.
- Collapse lists.
- Choose or create a topic sentence.

Computationally this translates into many subproblems:

- Collecting relevant source material (text).
- Collecting user constraints on the summary such as style, focus, length etc.
- Identifying the relationships between portions of the source (ranging from individual words to full documents).
- Scoring or extracting the important portions of source (from entities to sections).
- Explicitly representing the assertions being made in the source.
- Generating text for a summary given some subset of: source text, user guidance, portion relationships, indicators of importance, extracted information.
- Checking and finalizing the generated summary.
- Providing access to portions of the source document that support the summary.
- Potentially customizing all the above based on specific domain requirements such as law or medicine.

This complexity is reflected in the diversity of solutions. Early summarization research (e.g. (Mihalcea & Tarau, 2004)) scored sentences, potentially adjusted those scores using relationships, and directly quoted the highest-scoring passages as the final summary. More recent work (e.g. (Xiao et al., 2022)) has developed deep neural networks that consume source material and directly generate a final summary.

2 Summary of Articles

The following section provides an overview of significant research initiatives in the field of text summarization. The literature we selected examines the issues of i) dealing with large

amounts of input, ii) extracting and capitalizing on content details, and iii) divide-and-conquer strategies to the subproblems outlined above.

2.1 Handling More Content

2.1.1 Generating Wikipedia Articles by Summarizing Long Sequences (P. J. Liu et al., 2018)

These authors show that generating Wikipedia articles can be approached as a multi-document summarization problem with a two-stage extractive-abstractive framework. Information is coarsely extracted from source web pages, and then fed into a decoder-only Transformer capable of both consuming long documents and generating output of a user-specified length. They rely on the article Title (Wikipedia topic) and related web searches for the input to their approach, and use this to generate the Lead (the first paragraph) of the Wikipedia article.

The authors investigate 5 extractive methods. Identity (quoting the first X sentences from the citing web pages), TF-IDF, TextRank, SumBasic, and Cheating (using recall of bigrams). The extracted text is concatenated in order, with the most relevant sentences at the beginning. The extractive phase is treated as an Information Retrieval task with the purpose of identifying salient sentences and re-ranking them according to their importance.

After extraction, the authors treat the abstraction problem as a sequence transduction problem from very long sentences to medium output sequences. They first apply an LSTM encoder-decoder with attention (Seq2Seq, (Bahdanau et al., 2014)) as a baseline and optimize Maximum Likelihood. The authors also evaluated Transformer Encoder-Decoder models, and finally introduce a modified decoder-only version, as well as introducing a novel architecture they call transformer decoder with memory-compressed attention (T-DMCA) which allows them to process sentences three times larger by modifying the multi-headed self-attention of the Transformer to reduce memory usage between the Query and the Key. This is done by reducing the number of keys and values at each point using strided convolutions. Alternating layers of local attention and strided attention enabled them to adjust the convolution window size and reduce the number of network activations – with the practical effect of making the maximum possible size of the abstracted summary a selectable parameter.

2.1.2 Discourse-Aware Unsupervised Summarization of Long Scientific Documents (Dong et al., 2021)

This paper demonstrates the use of sentence position and sentence-section similarity to score content for extractive summarization, in a scoring method called HipoRank. The authors assert that sentences that are similar to sections, or fall near the beginning and end of sections are more important and should be prioritized. While the authors do not present a novel use case for document summarization, they focus on summarizing scientific articles, which are substantially longer than many summarization data set sources and have reliable sectional structure. They also emphasize the importance of faithfully preserving the original text in some domains, where the exact wording can be important - such as legal or medical documents, and note the lack of this as a significant shortcoming of abstractive approaches.

The authors approach the problem as follows: sentences are initially embedded, whether with TFIDF vectors, NN embeddings, or other methods – the authors test BERT (Devlin et al., 2016), PacSum BERT (Zheng & Lapata, 2019), SentBERT and SentRoBERTa (Reimers & Gurevych, 2019), and BioMed word2vec (Pyysalo et al., 2013). Once sentences are embedded, section representations are built with mean pooling. Similarity between sentence pairs within each section is calculated, and sentence similarity to each section representation is calculated. Each sentence's importance is calculated as a weighted sum of its position in a section, and each section's position in the document (giving preference to sentences near the beginning or end of a section and sections

near the beginning or end of the document). The highest scoring sentences are selected for summary construction until a predefined word limit is reached.

2.1.3 PEGASUS: pre-training with extracted gap-sentences for abstractive summarization (Zhang et al., 2020)

In PEGASUS, the authors develop a foundational (reusable, pretrained) abstractive summarization model, able to adapt to unseen summarization datasets when fine-tuned with as few as 1000 examples. This model is pretrained with a novel self-supervised objective – the model's ability to reconstruct important "gap" sentences. Unlike BERT where words are randomly masked during pre-training, the PEGASUS authors explore the masking of specific sentences from source paragraphs in addition to BERT's masked-language modeling (MLM), and find that Gap Sentence Generation (GSG) is superior than GSG + MLM with a Gap Sentence Ratio of 30%. The pre-trained model learns to reconstruct (masked) key sentences from a corpus, and it is later fine-tuned on a specific abstractive summarization dataset. Potential gap sentences are scored using a compound ROUGE score when compared to the source paragraph. The authors' assumption is high-scoring sentences are the most salient and therefore should be reproducible. This method encourages the model to learn to generate coherent and relevant sentences that are crucial for the understanding of the document. The authors also note the irrelevance of MLM to the task of abstractive summarization.

The authors tested two different architectures of Pegasus (Base, and Large – each with varying maximum input sizes, hidden dimensions, and number of transformer blocks), as well as 6 variants of GSG to select the most saliant sentences. They included a range of datasets in their experiments and note that pretrained models transfer more effectively to downstream tasks when their domains are aligned better – a possible indication of lack of generalization across domains. They also call out the maximum input size limitation of transformer architectures, which impacts performance on large datasets such as BIGPATENT, arXiv, PubMed and Multi-news (all beyond 1024 tokens).

2.2 Capitalizing on Structure

2.2.1 Knowledge-Graph Augmented Abstractive Summarization with Semantic-Driven Cloze Reward (Huang et al., 2020)

This paper seeks to address two problems in abstractive summarization: that 1) they are often very close to an extractive summary, and 2) they can generate fictious content. The paper presents a novel architecture for abstractive text summarization using a sequential document encoder and a knowledge graph encoder.

The graph encoder is meant to explicitly make available the structured information of the source texts. This document level context helps enhance and connect entities that might be referenced over multiple sentences, as part of different events and/or positioned far from each other in the document. Previous models struggle with this.

Two graphs are tested. One that captures entities' document level context and another that captures paragraph level context. They used Stanford OpenIE to extract (subject, predicate, object) triples and linking subject and object nodes with a directed edge for the predicate.

For the sequential encoder, they added a BiLSTM and attention layer to the last layer of a RoBERTa model. For the decoder, they use a single layer LSTM. They train the system with a new objective function that they describe is "a reward based on a multiple-choice cloze test to drive the model to better capture entity interactions". They follow this with reinforcement learning based on a novel cost function they call "*multiple choice cloze reward"* which they believe helps produce better summaries.

In the final analysis, their results across ROUGE-1, ROUGE-2, and ROUGE-L metrics are slightly worse than a finetuned BART model, but better than any of the other summarizers they tested

against. The main takeaway is that a dual graph encoder may help capture context and connections regarding entities that are referenced in multiple sentences and distances in a document, but it remains to be proven.

2.2.2 PRIMERA: Pyramid-based Masked Sentence Pre-training for Multi-document Summarization (Xiao et al., 2022)

This paper establishes a performant general-domain foundational model for summarization that can be leveraged by fine-tuning or incorporated into other systems. The model has been optimized for summarization by training a long-input sequence-to-sequence model to reconstruct high-scoring summary sentences. PRIMERA is optimized for gap sentence generation, a la PEGASUS, with a novel strategy for sentence selection motivated by perceived shortcomings in earlier work – specifically, the blindness of ROUGE scoring to certain kinds of unimportant repetition. The authors set out to use more explicit extracted information.

PRIMERA is initialized with a pretranied Longformer-Encoder-Decoder large model. This is retrained for summarization by taking clusters of related documents and performing named entity recognition (NER), using SpaCy (Honnibal & Montani, 2017) as a proxy for recurring assertions (dubbed Summary Content Units). Frequently appearing entities are used to signal relevance, and the model selects sentences with the most frequently mentioned entities; *then* the sentence with the highest ROUGE score sum across the cluster of documents is extracted. The model is trained to reproduce these sentences as a nominal summary for the document cluster, making this an unsupervised model (as no actual summaries are required).

Using PRIMERA as a pretrained or fine-tuned model, summary generation is thus a process of feeding new documents to the model and generating text. The model was tested on a multi-document summary data set, with ROUGE scores and human evaluation that scored the summaries for precision, recall and F1 of factual "summary content units" produced as well as the model's grammar, clarity, and coherence.

2.2.3 Enhancing Abstractive Summarization with Extracted Knowledge Graphs and Multi- Source Transformers (Chen et al., 2023)

This paper seeks to address the problem of fictitious or incorrect information prevalent in abstractive summarizations generated by LLMs. They also create a new dataset from Wikipedia articles which are longer and sparser and appear to present a greater summarization challenge than denser news articles.

Building on [2.2.1], this paper presents an approach whereby embeddings from knowledge graphs are utilized to augment text embeddings before being passed to a BART model. Knowledge graph embeddings are meant to capture entity connections and global context. They are seen as a good source of latent semantics that can provide a scaffold for text generation.

The knowledge graph is created using a Graph Attention Network (GAT, (Veličković et al., 2018)) architecture with (subject, predicate, object) triplets as nodes and edges. As with [2.2.1], they used Stanford's OpenIE to extract the triplets. The embeddings generated by the GAT were then "linked" to the text embeddings using the TranE method (Bordes et al., 2013) and the combination was input into a pre-trained BART fine-tuned against a target summary using cross-entropy loss. They seem to indicate that they perform a second round of training with what they term "the entity salience objective", training the model to predict if entities appear in the abstract.

As with (Huang et al., 2020), the results don't seem to be superior to BART, but all of the models tested performed far worse on the authors' Wikipedia based data as compared to news article datasets. This raises the question of whether summarization methods should depend on the density and organization of relevant

content, and whether text summarization is a task that fundamentally changes based on the domain and end-user needs.

2.3 Flexible Workflow

2.3.1 Leveraging BERT for Extractive Text Summarization on Lectures (Miller, 2019)

(Miller, 2019) introduces a more specific use case for the summarization of lecture transcripts. The author asserts that while abstractive text summarization is often preferrable, it suffers from lack of generalizability and requires more computational resources to train. This paper introduces an extractive summarization method that requires no training or fine tuning.

The proposed pipeline embeds sentences with a pretrained model, and finds their 'centroid' sentences using K-means clustering. The author argues that centroids should extract the most meaningful summary information.

Sentences are embedded with general purpose pre-trained transformers. In line with convention, the authors considered the embedding of the sentence separator token [CLS], as well as the full set of final word embeddings. The author found that the embedding of [CLS] does not necessarily produce the best embedding representation for the sentence, and instead opted to average the NxWxE word embedding matrix to produce a better NxE representation of each sentence.

The author experimented using BERT, GPT2, and an ensemble of the two, as well as probing which model layers produced the best embeddings for this task. The ensemble performed the best, but in the interest of keeping training costs down and minimizing inference latency, the author recommended the second to last layer of BERT only. In addition, the author tested Gaussian Mixture Model clustering and observed similar performance to K-means.

Ultimately this model was not the most effective at extractive summarization, and its focus on individual sentences led to problems with indirect references.

2.3.2 On Extractive and Abstractive Neural Document Summarization with Transformer Language Models (Pilault et al., 2020)

To address the problem of summarizing long documents, this paper proposes a structure-informed extractive summarization as input into an abstractive summarizer.

The paper proposes a novel approach that abandons the strictly sequential input paradigm. Using technical papers as a template, extractive summaries of the Abstract, Introduction, and the body of the paper are generated. These are then inputted for abstract summarization into a transformer in the following order: Introduction, extracted summary of the paper, Abstract, the original document less the Abstract and Introduction sections.

The authors experiment with two extractive models: 1) a hierarchical seq2seq sentence pointer 2) a sentence classifier. This is intended to demonstrate the sensitivity of the abstractive summarizer to the extractive input. For generating the abstractive summary, they use a transformer model based on the same architecture as GPT-2 (Radford et al., 2019).

Their results indicate that the hierarchical s2s sentence pointer extractive summarizer input performed better on an arXiv dataset while the sentence classifier performed better on a PubMed dataset. They compared their model to a number of models from previous work (Lead-10, SumBasic, LexRank, etc.,) and achieved generally superior results. BART was not included in the tests.

This paper's primary contribution is the use of reordered extractive summaries of the source sections as supplemental input to the abstractive summarizer.

2.3.3 Controlled Text Reduction (Slobodkin et al., 2022)

This paper articulates the lines between delineating discrete pieces of information (as highlights), and constructing a summary given a document and those highlights. Besides offering one way to decompose the document summarization task, this isolates a new and practical task: taking a document that has been marked up by a user and generating a summary that includes all and only the highlighted facts.

Besides being a desirable feature for some users this separation is also motivated by an interest in affording more performant modular text processing pipelines, capitalizing on the independent optimization of these subtasks, and gaining linguistic insights more readily at this procedural boundary without probing internal layers of an end to end model. The authors demonstrate instruct-able document summarization using Longformer Encoder-Decoder (Beltagy et al., 2020) with global attention tied to tag tokens that delineate spans of highlighted text.

In their experiments, the authors show that LED can capitalize on the highlighting to produce summaries that preferentially include the facts captured in the highlighted text without introducing extraneous details, and that the model's access to the full content is important for producing a coherent summary when compared to generation of summaries from highlighted text alone. They also show that different highlights lead to measurably different summaries. In order to perform theses tests the authors used MTurk and an established procedure to get gold standard "reverse engineered" highlights plausibly corresponding to document/summary pairs, and augmented this with silver data using information extraction and proposition alignment algorithms. They measured the quality of summaries generated from some mixture of document content and highlighted spans using summary fact precision and recall.

3 Comparison

Here we compare how the papers treated overall document structure, represented detailed content, and subdivided the summarization into slightly different tasks.

3.1 Document Structure

In (Dong et al., 2021) the authors frame their scoring function in terms of a weighted asymmetric graph, and describe the scores as centrality scores, but the scores are not related to the usual concept of graph centrality. This is in contrast to scoring systems like LexRank (Erkan & Radev, 2004) and TextRank (Mihalcea & Tarau, 2004), which use preliminary node scores to initialize message passing in the manner of PageRank or GAT. We see a better analog as (Edmundson, 1969), an early example of research into use of document structural cues for sentence scoring; that author would certainly have anticipated the merits of HipoRank.

While they do not use graph structure, (Pilault et al., 2020), (Dong et al., 2021), and (Edmundson, 1969) share a common theme of exploiting informative patterns in document structure. These results suggest that in feature-enhanced models some form of extractive summary and structural information will remain valuable.

3.2 Content Structure

In (Slobodkin et al., 2022) the authors note the relationship of highlighting to extractive summary scoring such as (Miller, 2019) and (Pilault et al., 2020); they also compare highlighting to propositional alignment (Ernst et al., 2021) and more explicit information extraction techniques such as those used in (Huang et al., 2020), (Chen et al., 2023), and (Xiao et al., 2022). This work establishes what we see as a critical niche subtask for continued progress in summarization.

(Huang et al., 2020) uses information extraction to produce structured tuples for scoring; recognizing the shortcomings of current IE algorithms, PRIMERA simplifies and relies on a more reliable named entity resolution algorithm to extract entities as an approximate indicator of

the presence of important facts, in its Entity Pyramid scoring function. Both PEGASUS and PRIMERA use gap sentence generation, but the former relies on simple sentence overlap scores and the latter extracts more specific structured information. This arrangement is one reason we expect information extraction to demonstrably improve summarization at some point. (Chen et al., 2023) train with what they term "the entity salience objective"; this seems to draw on analogous information to PRIMERA's reliance on Entity Pyramid scoring in (Xiao et al., 2022).

Tangentially related, several of our papers (Dong et al., 2021; Slobodkin et al., 2022; Xiao et al., 2022; Zhang et al., 2020), have emphasized ROUGE's limitations and relied on human evaluations via MTturk to perform more nuanced scoring of summary content and fluency. Our sense is that this is a signal of the importance of content structure to the ideals of summarization, and the limitations that past solutions have come up against because of the marginal quality of information extraction solutions.

3.3 Subtask Structure

Part of the preprocessing in (Miller, 2019) sought to simplify the problem by removing challenging sentences which require additional context. This is contrast to the models that explicitly relied upon more complete context such as (Xiao et al., 2022) and (Slobodkin et al., 2022). Note (Miller, 2019) also is a generalization of MEAD (Radev et al., 2000), which performed multi-document summarization by embedding sentences with topic detection and tracking, and extracting sentences that were near cluster centroids. Unlike the other papers reviewed, (Miller, 2019) does not provide any form of evaluation and instead treats the extractive summarization problem only as an information retrieval subtask solved by a heuristic (the distance to K-means clusters). In a sense this creates a noteworthy contrast with the more focused and systematic analysis of the other papers surveyed.

(Xiao et al., 2022) requires a more complicated comparison to other models. They used gap sentence generation as a training objective to create an *unsupervised* foundational encoder-decoder model for summarization. Extractive models typically have a scoring function, and then produce summaries mechanically from the scored content; abstractive models on the other hand are typically supervised and trained on document-summary pairs; hybrid models may do both, feeding content and scores (or in the case of (Slobodkin et al., 2022), even highlighted spans) into the model with the summary as a target. In contrast, PRIMERA scores sentences… but only to select gap sentences and train the model to regenerate those (allowing unsupervised learning in an encoder/decoder system). It does not score input sentences before generating an end-user summary. While end-to-end solutions make summarization straightforward, this also limits transparency if a model does not produce faithful or complete summary text. Both (Huang et al., 2020) and (Chen et al., 2023) seek to address the problem of fictitious or incorrect information prevalent in abstractive summarizations.

4 Future Work

We see potential for future investigation and development in the following areas:

1. Experiment with model architectures to enable ingesting much longer sequences.
2. Develop techniques to capture document level context.
3. Capture entity relationships over long distances in the document.
4. Provide visibility and an "audit trail" from the summary to the original document so users can obtain more detail if needed or verify summary accuracy.

Below, we discuss in more detail how the material we have reviewed inform the areas we have highlighted.

Highlighted text in (Slobodkin et al., 2022) is used as closer substitute for explicitly represented facts (closer than implicit latent representation of those facts in internal or surface text embeddings)… but it still side steps the question

of how to take noteworthy facts and produce text that conveys them, and the question of how to measure the effective conveyance of facts without manual review (they perform this ground truth faithfulness analysis manually). Drawing upon other IE/KE literature we would propose to add the input of a structured factual knowledge — using the source document to provide nuance, context, and style and tone of how the facts should be treated.

While the separation of tasks obviates the need for probing at this procedural boundary, between selecting key information and summarizing content naturally in light of that information… it is possible or even likely that highlighting text is merely a convenient strategy for annotating text and is not a good indicator of how people or deep end-to-end models understand facts in text or decide what to write in a summary. Research building and probing models to study these phenomena remains important future work.

The authors produced data using a thoughtful but expedient method; the result is a data set that is likely to misrepresent the true data distribution of highlighted text and the most desirable summaries. Continued development of representative data sets in multiple languages and for a range of domains, annotator interests, and summary applications remains important future work.

The need to faithfully reproduce original text, cited by (Dong et al., 2021), does not necessarily equate to a need to exactly quote original texts; what it implies is a need to preserve the structure, content, and nuance of the assertions laid out in the original document. Doing so when preparing abstractive summaries is a clearly open problem requiring future work. This is especially the case when seeking to develop systematic summarization methods that do not collapse in the face of out-of-distribution content, which is a particular sticking point for current encoder/decoder neural networks. The authors comment that neural-based models are often blind to the topical influences on content resulting from structured sections in scientific articles, and this raises further questions for future work, examining whether models such as Longformer (Beltagy et al., 2020) that were not tested in this work could be probed for insight as to how they treat sections.

The biggest gap left by (Xiao et al., 2022) is the use of NER results as a proxy for other assertions; research needs to be done to assess how good recurring entities are as a proxy for the important noteworthy facts in a document cluster that would go into a summary.

Explicitly summarizing the logical content of original sources would close key application gaps of summarization related to usability: provenance and granularity. This strategy would directly enable the validation of abstractively generated summaries, addressing the factuality concerns called out by the authors. A summary would ideally be traceable to specific assertions in the original source, even if phrased differently, and when a consumer wants more details about something in a summary it should be possible to expand the summary with more details on a given point (or generally expand from a headline to an executive summary from an abstract, etc.).

In (Huang et al., 2020) the authors suggest that for future work, their model should be built on top of a large pre-trained encoder decoder model (BART). This is an area we will consider exploring.

Finally (Zhang et al, 2019) acknowledge the limitations imposed by Transformers architecture with regards to the maximum input size, which in turns limits summarization on large corpora – this is an invitation to explore more recent architectures which might overcome these limitations.

References

Automatic citation updates are disabled. To see the bibliography, click Refresh in the Zotero tab.

A. Appendix: Legacy Approaches

Here is a collection of pre-transformers methods on doc summarization. While the most highly performing approaches today are dominated by transformers, moving in the direction of LLMs, understanding what was tried prior to massive data sets and supercomputing (including early attention networks) brings interesting context.

(Luhn, 1958) Pulls statistically interesting sentences into an "auto-abstract." Favors *"significant" words vs very frequent or rare words*, and scores significant word clusters as (n^2)/c where n is the number of significant words within a given window size of each other and c is the cluster size (cluster can exceed a window with chaining).

(Edmundson, 1969) Selection of sentences for a document summary using 3 new components that are superior to frequent words: *pragmatic "cue" words, title and heading words, and structural information* from sentence location.

(Carbonell & Goldstein, 1998) Maximum margin relevance (MMR) is used to produce summaries of single documents that avoid redundancy. This is for retrieval and summarization; hi marginal relevance means a document is relevant to a query and contains *minimal similarity to previously selected documents*.

(Radev et al., 2000) MEAD; generation of a single summary of multiple documents (an event cluster pulled from a topic detection and tracking system), using cluster centroids produced by a topic detection and tracking system and two new techniques: cluster-based sentence utility (CBSU, the *relevance of a sentence to a cluster*) and cross-sentence informational subsumption (CSIS, used to eliminate *textually redundant sentences*).

(Erkan & Radev, 2004) Extractive text summarization using LexRank, which computes sentence importance as *eigenvector centrality in a graph of sentences*. The graph is an adjacency matrix weighted with intra-sentence cosine similarity. Sentences are represented as bag-of-word TFIDF vectors containing word counts times inverse document frequency.

(Mihalcea & Tarau, 2004) TextRank is a graph-based ranking model for text processing, using PageRank (Brin & Page, 1998) or alternatively HITS (Kleinberg, 1999) or Positional Function (Herings et al., 2001). Sentences are the nodes in the graph; sentences "vote" for one another based on similarity, in this case word overlap normalized by sentence length. The *highest ranked sentences* are used in extractive summaries.

(Arora & Ravindran, 2008) Latent Dirichlet allocation (Blei et al., 2003) applied to multi-document summarization. LDA is used to build a Bayeisan topic model jointly modeling document topic frequency and topic word frequency, and then singular value decomposition on sentence vectors allows finding sentences that are orthogonal to one another (reducing redundancy) and best represent the topics.

(Ozsoy et al., 2011) Latent semantic analysis strategies use singular value decomposition of a word/sentence co-occurrence matrix (populated with either word frequencies, word presence, word TFIDF, word log entropy, noun frequencies, or normalized TFIDF) for finding semantically similar words and sentences. The authors propose further embellishments to either *reduce the value of multi-topic sentences* or to focus summarization on *primary rather than secondary topics*. These methods are evocative of the similarity scores used for information retrieval in ColBERT (Khattab & Zaharia, 2020).

(Nallapati et al., 2016) Abstractive summarization using attentional Recurrent Neural Network (RNN) encoder-decoder models (Bahdanau et al., 2014). Initial model is a bidirectional gated recurrent unit GRU-RNN (Chung et al., 2014) with the large vocabulary 'trick' (LVT) described in (Jean et al., 2015). The authors also tried augmenting embeddings with POS and NER tags, and TFIDF statistics; using a switching decoder/pointer architecture (that sometimes emits pointers into the source text instead of generating an OOV marker); or using a bidirectional RNN with attention at both word and sentence levels to encode content.

(L. Liu et al., 2017) Uses a Generative Adversarial Network (Goodfellow et al., 2014), simultaneously training an abstractive summary generator — comprising a bidirectional LSTM encoder and an attention-based LSTM decoder per (See et al., 2017) — and a CNN text classifier (Kim, 2014) discriminator model trained to classify summaries as machine or human generated.

B. Linguistic Challenges to Abstract Text Summarization

ATS aims at condensing all important information that needs to be summarized. Although many summarization techniques can be used to create summaries (abstractive, extractive, hybrid), the goal is to create natural human-like summaries. However, existing ATS systems – including ones that rely on Deep Learning – still have several significant challenges with that regard. The following list (Mridha et al., 2021) details a few of these challenges, among many:

- **Anaphora/Cataphora:** The use of pronouns and other referential expressions to connect different parts of a text. These expressions can create challenges in ATS due to the need to resolve the references they make. Summarization solutions need to maintain context, coherence, and logical structure. Dependency resolution is not always straightforward, and syntactic structure is rarely an arbitrator on its own, as shown in the example below:
  - Eyas couldn't fit his hand in his pocket because it was too big. (in which "it" is an anaphoric expression which refers to Eyas's hand).
  - Eyas couldn't fit his hand in his pocket because it was too small. (in which "it" is an anaphoric expression which refers to Eyas's pocket).
- **Evaluation:** Analyzing summarization using precision and recall can yield deceptive results which might not support the desired conclusion. Existing metric formulations such as ROUGE rely partly on the cooccurrence of words in the source text and the summary, which could not be an appropriate measure for abstractive text summarization tasks. Additionally, Metrics such as ROUGE and BLEU seem to show less significance than the human evaluation score – which is still prominently used in many related works, specially via crowdsourcing platforms. Generally, Automatic Evaluation criteria require the existence of gold passages in the selected datasets, which is difficult or expensive to collect.
- **Sentence selection for extractive summarization:** Typically, ATS systems select the most appropriate sentences from the original text and marks them as essential (in the case of extractive text summarization) – But annotating sentences in the source text with an importance score is subjective and typically relies on heuristics or vector representations and similarity matrices or clusters. There seems to be no reliable way to pinpoint the sentences that contain the most salient information.
- **Interpretability:** Abstractive text summarization systems generate text according to condensed representations of the extracted text (or the source text). ATS typically struggle with the complexities of human written languages – therefore ensuring inter-

pretability through abstract models is a difficult task.

- **Long documents summarization:** Existing solutions leveraging transformers have exceeded the performance of typical RNN based solution in performance and time complexity but are limited by the maximum token size of input sentences, forcing researchers to truncate input sentences to the maximum permissible for the transformer architecture of their choice.

In addition to the areas covered above, several drawbacks have been faced with the development of ATS, such as the lack of different, diversified training datasets that include gold passages; ambiguity in word senses; and attaining higher level of abstraction.