Unsupervised Summarization of Public Talk Radio

by

Shayne O'Brien

B.S., State University of New York at Geneseo (2017)

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2019

Author ..... Signature redacted .....
Program in Media Arts and Sciences, School of Architecture and Planning
May 24, 2019

Certified by ..... Signature redacted .....
Deb Roy
Professor
Program in Media Arts and Sciences
Thesis Supervisor

Accepted by ..... Signature redacted .....
Tod Machover
Academic Head
Program in Media Arts and Sciences

Unsupervised Summarization of Public Talk Radio

by

Shayne O'Brien

Submitted to the Program in Media Arts and Sciences, School of Architecture and Planning
on May 24, 2019, in partial fulfillment of the
requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

Talk radio exerts significant influence on the political and social dynamics of the United States, but labor-intensive data collection and curation processes have prevented previous works from analyzing its content at scale. Over the past year, the Laboratory for Social Machines and Cortico have created an ingest system to record and automatically transcribe audo from more than 150 public talk radio stations across the country. Using the outputs from this ingest, I introduce "hierarchical compression" for neural unsupervised summarization of spoken opinion in conversational dialogue. By relying on an unsupervised framework that obviates the need for labeled data, the summarization task becomes largely agnostic to human input beyond necessary decisions regarding model architecture, input data, and output length. Trained models are thus able to automatically identify and summarize opinion in a dynamic fashion, which is noted in relevant literature as one of the most significant obstacles to fully unlocking talk radio as a data source for linguistic, ethnographic, and political analysis. To evaluate model performance, I create a novel spoken opinion summarization dataset consisting of compressed versions of "representative," opinion-containing utterances extracted from a hand-curated and crowdsource-annotated dataset of 275 snippets. I use this evaluation dataset to show that my model quantitatively outperforms strong rule- and graph-based unsupervised baselines on ROUGE and METEOR while qualitatively demonstrating fluency and information retention according to human judges. Additional analyses of model outputs show that many improvements are still yet to be made to this model, thus laying the ground for its use in important future work such as characterizing the linguistic structure of spoken opinion "in the wild.'

Thesis Supervisor: Deb Roy
Title: Professor
Program in Media Arts and Sciences

Acknowledgments

First and foremost, I would like to thank my advisor Deb Roy for providing me with the opportunities to study under his tutelage. My perspective has been greatly augmented during my time with the Laboratory for Social Machines (LSM) and I feel that I am now much more able to see the "bigger" picture than I was when I started. MIT is a magical place that is easy to get lost in, but Deb guided me through it.

Second, it is my pleasure to thank my wonderful readers Alexander "Sasha" Rush and Stephen Ansolabehere for providing me with guidance in shaping this thesis. Their feedback was invaluable to the rigor of my research and I am grateful to have had the pleasure of working with them both over the past year.

Third, I want to thank all of the incredible friends and lab-mates that I have met during my time in Cambridge for keeping me sane and grounded. My years at MIT were two of the most prosperous of my life and I genuinely owe my ability to say this to them. They know who they are and thanks to them, I have come to better learn who I am.

Fourth, thank you to everyone in my life that I consider to be family, blood-related or not. They cheer me on along every phase of my life and are the ones who stare at me in disbelief when I fall and say I do not want to get up. My family are my rock and their belief in me is one of—if not the—only aspects of my life that is completely and utterly invariant. I will never be able to fully express my gratitude for having people like them in my life, but here is my best try: thank you so, so incredibly much.

Lastly, thank you to the reader for investing time into this document. My goal in writing this thesis was to provide inspiration and insight. If it does so for even a single person, I will rest satisfied that my efforts were brought to fruition.

To my mother, who passed away one year before I began at MIT. While I was not able to share this journey with you, your teachings were what made it possible.

Unsupervised Summarization of Public Talk Radio

by

Shayne O'Brien

The following people served as readers for this thesis:

Signature redacted

Thesis Reader .....

Alexander "Sasha" Rush
Assistant Professor of Computer Science
Harvard University

Signature redacted

Thesis Reader ..

.....

Stephen Ansolabehere
Professor of Government
Harvard University

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Talk radio is an untapped goldmine of public opinion. It reaches millions of Americans every week, the majority of whom use it as one of their primary sources of news [23, 24]. To political and ethnographic researchers, the format of talk radio is appealing: show hosts share their opinions on some issue or topic and then invite listeners to "call-in" and voice their thoughts on the matter. A concise overview of mainstream and peripheral perspectives is presented in this way. Information propagation of this type happens on thousands of stations across the country every day, making talk radio a valuable source for understanding how the American people think.

Historically, sifting through talk radio has been too labor-intensive a task for its use in large-scale analyses [91]. It would take a monumental number of human hours to manually search through this audio in hopes of finding opinion-related content. Furthermore, there are no large datasets summarization datasets for transcribed speech and annotating enough data to train a supervised neural network model to complete the task would be prohibitively expensive. As a result, most previous analyses of talk radio have been practically limited to using only a couple dozen hours of audio across a few stations. This is an inherent limitation of these studies given that billions of words are broadcast across America every single day.

Over the past year, the Laboratory for Social Machines (LSM) and Cortico[1] have

[1]Cortico is a nonprofit co-founded by Professor Deb Roy in 2017 to scale, deploy, and augment the research of the Laboratory for Social Machines.

set up an ingest system to continuously collect U.S. talk radio broadcastings. By combining this ingest with modern methods in natural language processing (NLP), we have begun the process of unlocking the aforementioned potential of public talk radio to better understand the public sphere. In this thesis, I lay out a key aspect to this process: a model that is able to summarize transcribed speech without the need for labeled data.

1.1 Motivation

Although identification of opinion-containing content can be crudely accomplished through simple keyword searching, this method is not comprehensive nor does it satisfy a researcher's need to quickly understand the retrieved data. Since summary statistics cannot semantically interpret language, readers interested in understanding public opinion on talk radio must go through these data by hand. Unfortunately, snippets of talk radio tend to consist of several hundreds or thousands of words; it would take the average reader several minutes to process a single one. In contexts with an overabundance of text data and a lack of human resources to process them by hand, automatic summarization proves to be a valuable tool for fast and efficient distillation of information. This statement holds especially true in the presence of high levels of noise such as music, advertisements, and traffic and weather announcements.

Despite significant efforts to develop spoken dialogue systems, colloquial speech summarization of transcribed text remains an open problem. This may be due to several of its characteristics that make it difficult to model such as its stream of consciousness formatting, frequent disfluencies, the diminishing likelihood of domain independence as a conversation grows in length, the loss of non-verbal information such as physical gestures and dialogue acts during recording, the erasure of verbal features such as prosody during transcription, the relative lack of its study compared to more highly-structured contexts such as newspaper articles, and an absence of consistent grammar usage and speaking style in most individuals.[2]

[2]Noam Chompsky is an example of an exception to this statement.

The last of these points creates a lack of applicability of out-of-the-box transfer of models learned on available summarization corpora. Since a large speech-based opinion summarization dataset does not exist and would be prohibitively expensive to create, options for supervised and transfer learning are limited. In this thesis, I propose to approach opinion summarization of colloquial speech in an unsupervised manner to facilitate this issue. As an added benefit, relying on such a framework allows output summaries to be less vulnerable to bias by minimally relying on human guidance outside of decisions regarding model architecture, input data, and output length. Such outputs should, in theory, produce a more accurate depiction of American public opinion by avoiding presumptions about what the data have to say.

1.2 Framing

Since spoken language is typically verbose, I[3] will be interpreting summarization in the light of compression. Under this framework, I will seek to reduce the size of an input sequence (e.g. an utterance or radio snippet[4]) by removing items (i.e. words from utterances and utterances from snippets) that consist purely of "extraneous" information. Extraneous information can be defined as any item found in a sequence that, when removed, does not alter the core meaning and fluency of the original sequence. Sequences that are compressed down to remove their extraneous information can be thought of summaries of the original sequence.

To illustrate this concept, consider the utterance "I, uh, I don't know ... things just, uh, they just well ... [laughter] things just seem shady on both sides." A large portion of this utterance consists of extraneous information in the form of disfluencies (e.g. "uh," "uhm," [laughter]), filler words and phrases (e.g. "just," "they just well," "I don't know"), and repetitions (e.g. "things"). While these words provide contextual clues that the speaker may be unsure of, uncomfortable or hesitant with they were

[3]In this thesis I use "I" and "my" to refer to work done by myself and "we" and "our" to refer to work done by LSM and/or Cortico. I also use "we" and "our" in mathematical contexts such as Section 4.1 despite it being my original work.

[4]See Chapter 4 for formal definitions of "utterance" and "snippet."

about to say, they are unnecessary to understand the principal message that the speaker is trying to communicate: "things seem shady on both sides." This is an example of an ideal compression that would be output by a model, perhaps with ellipses wherever one or more words were removed to indicate a compression has been performed.

From a compression perspective, utterances are relatively self-contained in that modifications can be easily validated by comparing to the original with respect to fluency and expression of intent. Extraneous information for snippets, on the other hand, is more difficult to define due to the nuances of how meaning is recursively created in colloquial conversations. Utterances typically build off one another to convey semantic intent. The subtle relationships between them thus makes it difficult to disentangle how removing one utterance would affect the presentation of the information presented in those that remain. This can be understood as removing context, which is pragmatically important to fully understand an utterance's motivations and contributions.

To the end of practically subverting this catch-22, I define extraneous information at the snippet-level to be utterances that are not "representative" of a piece of information found in the original snippet. Thus, representative items holistically represent the intent of a sequence when understood with knowledge that their union is an extract of the original. For example, one representative item of the snippet "`I have so, so many colored dogs. One is red and one is blue and the rest are multi-colored. All of them bark.`" would be "`I have so, so many colored dogs.`" The compression of this representative utterance could then be "`I have many colored dogs.`" While this may be a trivial example, it gets to the heart of what I define by negation to be extraneous information at the snippet-level: items that should not serve as summaries due to their lack of either relative or absolute representation of the original sequence.

1.3 Contributions

The primary contributions of this thesis are as follows:

- Methods for preprocessing and filtering transcribed conversational speech with a non-trivial word error rate to enhance the quality of outputs from non-state-of-the-art ASR systems that are operating at-scale

- The introduction of hierarchical (utterance- and snippet-level) compression for neural unsupervised summarization and techniques for enabling such a setup to produce human-readable outputs

- A new compression-based speech summarization dataset with multiple levels of annotation for extracting and compressing opinion in conversational speech

1.4 Thesis Outline

The current chapter is an introduction to this thesis and a brief overview of its framing, motivations, and contributions. The remaining chapters are arranged as follows:

- Chapter 2 provides a high-level overview of the system presented in this thesis, which consists of the radio ingest and corresponding summarization model.

- Chapter 3 showcases related work regarding unsupervised summarization, summary evaluation, existing speech summarization datasets, and opinion analysis of public talk radio content.

- Chapter 4 details the experimental setup of this thesis, including: a technical formulation of hierarchical compression; data preprocessing; noising methods tested; and the summarization model's architecture.

- Chapter 5 describes how I created the evaluation dataset used in this thesis and analyses of its inter-annotator agreement.

- Chapter 6 discusses the way I evaluate my model quantitatively and qualitatively and provides insights into which aspects of the experimental setup did and did not work.

- Chapter 7 summarizes the thesis by giving a high-level overview of the results, limitations, and potential avenues for future work. I then close the thesis by reflecting on what I would do differently if my work were to be reproduced.

Chapter 2

System Overview

The primary objective of this thesis is to create an unsupervised neural network model that is able to identify and summarize spoken opinion from automatically transcribed colloquial conversations. In this chapter I detail the radio ingest system used to collect data and the model I created to summarize them. The ingest system outlined in this chapter is not a contribution of this thesis; it was developed by other members of LSM and is its own independent project [15].

2.1 Radio Ingest

At the time of this thesis, LSM and Cortico have been ingesting talk radio data for more than one calendar year. These data come from 157 radio stations across 39 states[1] in the U.S., translating to approximately 4,000 hours of audio per day. Audio from these stations is collected by scraping the live stream from radio stations' publicly available webpages.[2] These audio streams are collected in three megabyte "snippets," each consisting of spoken "utterances," as continuously as possible. Since these live streams are occasionally interrupted by network difficulties, data collection processes are respawned and reconnected when dropped for any reason.

As audio is collected, it is passed into a Kaldi-based [70] automatic speech recog-

[1]States not included in the ingest at the time of this writing were Hawaii, Oregon, Kansas, Arkansas, Louisiana, Missouri, Indiana, Kentucky, North Carolina, Vermont, and Maine.

[2]The thesis title is "Unsupervised Summarization of Public Talk Radio" for this reason.

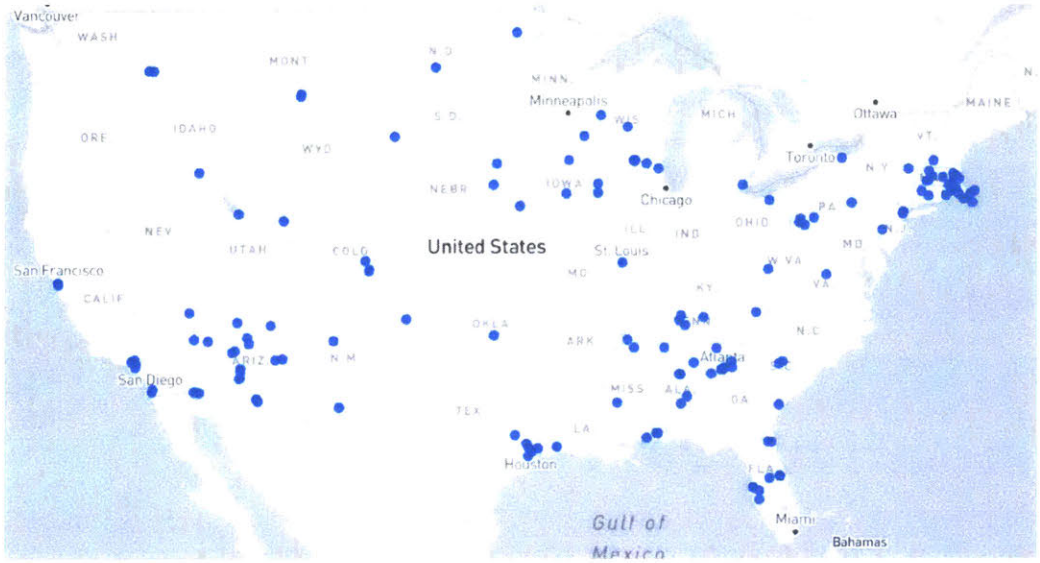


Figure 2-1: Locations of stations that are part of our radio ingest system as of April 2019 according to the Cortico EarShot API for exploring talk radio data [27].

nition (ASR) model. Kaldi is a popular speech recognition toolkit centered around provably correct algorithms that are non-specific to speech. The architecture of our ASR model is based off an entry [68] in the U.S. National Intelligency Agency's Automatic Speech Recognition in Reverberant Environments challenge [41]. It was chosen for its reasonable accuracy and efficient decoding procedure given the magnitude of data that it intakes. The principal difference between [68] and our model is that we redefine its vocabulary and retrain its language model on conservative talk radio from "The Rush Limbaugh Show" [82]. The language model is periodically retrained on the National Public Radio (NPR) "Talk of the Nation" and "Morning Edition" shows to maintain coverage over named entity mentions [72, 73].

Given inputs of html-scraped audio, the ASR model outputs plain text. Metadata is then collected as a post-processing step using auxiliary diarization.[3] and speaker gender imputation models from [3] and basic statistical calculations from the model. Examples of metadata that are available can be found in Figure 2-2 In evaluating the ASR model's performance, we observed a word-error-rate (WER) of 13.1% on a sample of 100 hours of data aired after the language model's temporal coverage. An industry-standard speech-to-text API from Google Cloud [48] scored a WER of

[3]Diarization is the partitioning of input audio into segments by speaker identity.

7.1% on the same test set. While their WER was about half of ours, it came at an hourly service cost of 40 times more. We thus deemed the performance of our model relatively acceptable given resource constraints.


```
{'audio_key': 'speechbox/stream_out/2018-12-01/WZAI/15_34_52.ra
w',
 'callsign': 'WZAI',
 'city': 'Brewster',
 'content': "we've been re broadcasting old broadcast of even o
lder broadcast in an attempt to give you an incentive to donate
money today but now that my friends and now now",
 'denorm_content': "We've been re broadcasting old broadcast of
even older broadcast in an attempt to give you an incentive to
donate money today but now that my friends and now now",
 'diary_band': 'S',
 'diary_env': 'U',
 'diary_gender': 'M',
 'diary_speaker_id': 'S12',
 'mean_word_confidence': 0.964333333333333,
 'segment_end_global': 1543678525.6,
 'segment_idx': 2,
 'segment_start_global': 1543678516.08,
 'segment_start_relative': 24.08,
 'show_confidence': 0.89,
 'show_name': 'Wait Wait',
 'show_source': 'NPR',
 'signature': '2f18d92f',
 'state': 'MA'}
```


Figure 2-2: An example of the types of metadata that are collected for utterances in our corpus. Note that "`diary`" is shorthand for diarization and refers to the outputs from the LIUM package [3]. Examples of fields in the metadata include the audio key of where the datum is stored, the global start and end times of the utterance, the mean word confidence of the utterance (low-to-high range [0.00, 1.00]), and whether the utterance was in-studio or a call-in (`diary_band`: 'S' or 'T').

Since this radio ingest generates the inputs to the summarization model that I describe in Section 2.2, it is crucial to note the following characteristics of the data source. First, it has a reasonably strong geographic bias relative to the talk radio universe of the United States [1]. This is due to a prioritization of areas with which LSM has collaborative connections, e.g. Boston, Massachusetts and Birmingham, Alabama. Many states are represented by our ingest, but not at levels proportional to land area nor population density. I invite the reader to consider Figure 2-1 to draw their own conclusions with respect to the system's location preferences.

Second, nearly two-thirds of the raw content collected can be determined to be syndicated, or repeated across more than one station in our ingest. Examples of syn-

dicated content include music, advertisements, shows that are broadcast across many stations such as NPR's "All Things Considered," and repeated airings of the same show on the same station. While we try to select stations to ingest that predominantly air original content and have heuristics in place to remove syndicated segments where frequently observed, syndication on public talk radio is virtually unavoidable. Syndicated content is identified using acoustic fingerprinting [22] or by removing all snippets from shows found on more than one radio station. I use the latter, more aggressive method to remove syndicated content from my analyses. See Section 4.2 for more details.

Third and as stated in the prelude to this Chapter, I did *not* actively participate in the development of the radio ingest. The implication of this is that many aspects of the data collection process were out of my control, such as the ASR model's minimum acceptable WER performance and which stations would be ingested by our system. While I agree with many of choices made by the system's primary contributors, these decisions may have unintended and unnoticed effects on the quality, generalizability, and performance of my model. I ask the reader to keep this context in mind when considering the final contributions of this thesis.

2.2 Summarization Model

Whereas most existing methods in unsupervised summarization approach the task are graph-based, I approach the problem neurally by introducing the concept of "hierarchical compression." Hierarchical compression attempts to remove verbosity and extraneous information from spoken language by: 1) greedily scoring utterances based on their information content, salience, position in the snippet, and novelty relative to the current summary; 2) pruning scored utterances down to only the top K, where K is the length in utterances of the desired output summary; and 3) compressing the content of the preserved utterances to be as concise as possible while retaining fluency and information with respect to the original. A mathematical formulation and details on how the model architecture is designed to be able to do this are described fully in

Section 4.1.

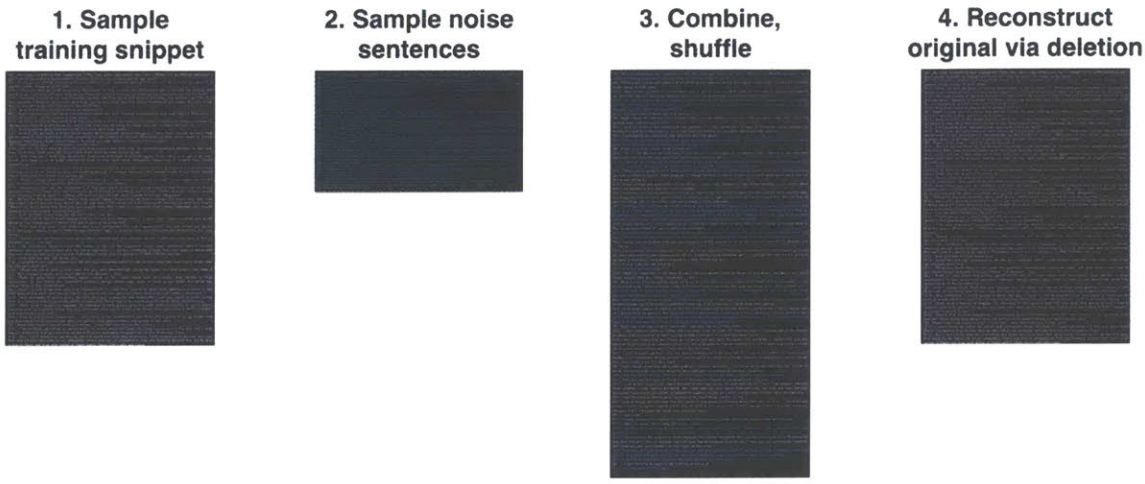


Figure 2-3: Snippet-level compression process overview. Noise (blue) is added to training snippets (grey) so that models can learn to remove extraneous, non-opinion related information.

In this setup, there are two principal hyperparameters which can be set by the user: number of utterances to be selected as summary items, and the proportion of these utterances' tokens that will be removed during compression. By modeling the problem sequentially at the snippet- and utterance-levels, respectively, hierarchical compression allows for native computational tractability of long snippets, handling of issues involved with modeling long-term and cross-utterance dependencies of language [52, 31, 9, 17], and opportunities for user interaction with the system for summary output size.

Hierarchical compression models are trained by learning to remove added noise, which consists of the content of randomly sampled utterances from the training corpus. At the snippet-level, utterances are added to the snippet in varying locations depending on which of the noising methods described in Section 4.3 is used. Intuitively, the goal in mind is to teach the model to focus on selecting representative and coherent sets of utterances to serve as a summary; the added noise should be considered extraneous by the model as it has no relation to the core intent of the original snippet. Each of these preserved utterances is then injected with word-level noise by randomly sampling additional utterances from the training corpus and subsam-

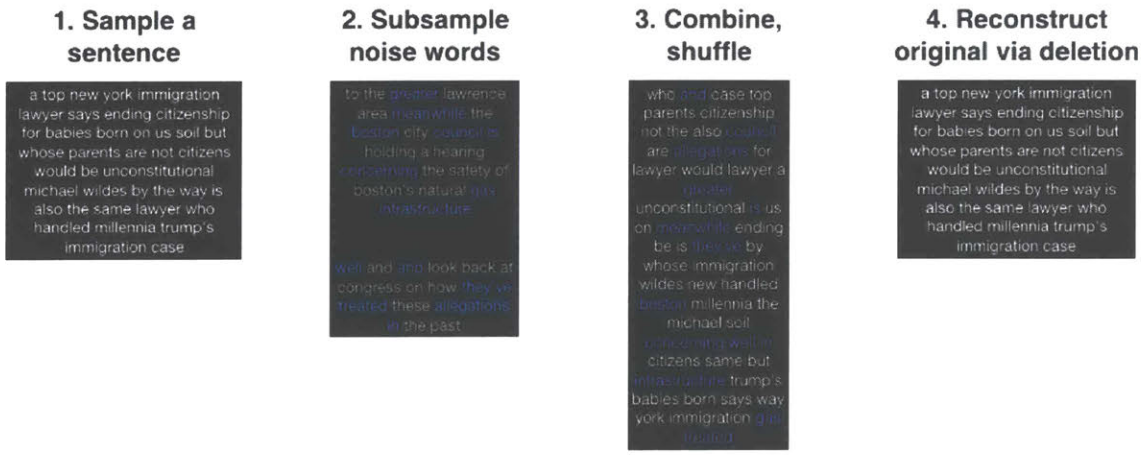


Figure 2-4: Utterance-level compression process overview. Noise (blue) is added to original utterance content (grey) so that models can learn to remove verbosity and extraneous details.

pling their words for insertion. The model then attempts to reconstruct the original in a similar fashion at the snippet-level. See Section 4.3 for descriptions of noising methods used. Visual overviews of this process can be found in Figures 2-4 and 2-3.

Since hierarchical summarization uses extraction rather than rephrasing, the degree of compression is measured by the proportion of words for utterances and utterances for snippets that were removed from the original. In the advent of the model receiving an utterance or snippet that does not contain meaningful information, such as in the case of an utterance consisting of exclusively filler words or a poorly transcribed snippet, the compression rate would be 100%. If the utterance is already as concise as possible, for example in utterances of less than six words, the compression rate would be 0%.

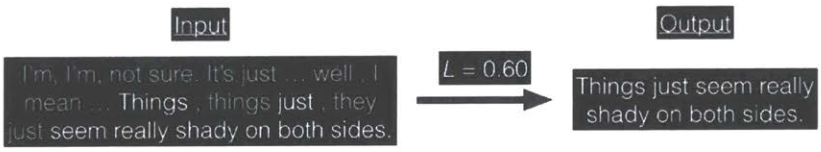


Figure 2-5: Example of utterance compression. The original utterance is compressed to a factor of $L = 0.60$ its original length. Grey words are considered extraneous and therefore removed. The output is shorter than the input while still preserving its overall meaning.

Chapter 3

Related Work

In this chapter I provide a survey of literature related to unsupervised summarization, summary evaluation, existing speech-based summarization datasets, and opinion analysis on public talk radio. Since the literature on summarization is vast, I have not included any related work with respect to supervised models. Readers interested in an overview of approaches to text summarization should refer to [6] and [76]. Beyond text summarization, I have also opted not to include methods on summarizing opinion as they predominantly focus on applying aspect extraction and sentiment analysis to product reviews and social media posts. For more on that topic, please see [14], [7] and [54].

3.1 Unsupervised Summarization

I will now give an overview of existing approaches to unsupervised summarization. Readers should to the cited references for respective authors' preprocessing and algorithmic details. I omit them here for ease of reading as they vary across works and are usually specific to a single domain.

3.1.1 Rule-based

In canonical news summarization datasets such as Gigaword [40, 67] and CNN Daily Mail [20], rule-based algorithms tend to perform strongly. Explorations provided in [62] and [21] speculate that this is because journalists are trained to write the main information first and then accompanying details later. While this is not necessarily true for speech, rule-based approaches have been shown to still be reasonable baselines for this context. Examples of commonly used rule-based algorithms at the snippet-level include: `Lead-N`, which takes the first N utterances of a snippet and outputs them as the summary [20]; `AllText`, which maps to the identity and thereby performing no summarization at all [33]; and `Oracle`, which provides 100% recall for utterance selection [5]. These baselines provide the utterances for baseline compression algorithms at the utterance-level, including: `RANDOM`, where n words are selected randomly from the source utterance without regard to ordering; `PREFIX`, in which the first 75 characters are output as the compressed summary [77]; and `F8W`, which is similar to `PREFIX` except the first eight (or alternatively, n) words of the utterance are taken instead [93].

3.1.2 Graph-based

Graph-based methods approach summarization by framing it as either a multi-utterance compression or utterance extraction task. Under multi-utterance compression, a graph is constructed to represent the words of co-occurring utterances via connected nodes and the shortest path between similar words of different utterances is output as the summary. To build the graph, the first utterance of a snippet is mapped to nodes representing words. Remaining utterances are then iteratively added to the graph by mapping them onto a pre-existing nodes with the same tokenization and part-of-speech.

Once the word graph is constructed, adjacent utterances words are connected with directed edges. Edge weights are initialized to one for nodes that were not connected previously and the number of nodes that have been mapped onto it otherwise. A

selection of candidate output compressions is then found by running the K-shortest paths algorithm, pruning candidates that do not meet the minimum length and sorting them based on their summed inverted edge weight scores. This method is said to guarantee that every input utterance corresponds to a loopless path and words referencing similar entities or actions are likely to all map onto the same node.

Following the introduction of this method by [96], subsequent efforts have focused on ways to improve the quality of these summaries by reranking candidate outputs using: weighting of nodes via probabilistic n-gram language models [34]; keyphrase extraction [18, 75]; word importance scoring [88]; budgeted submodular maximization [57, 58]; and k-core graph degeneracy [10]. In addition to these modifications, there has been some work specific to summarizing redundant opinions [37] and automatic speech recognition data [80].

In the utterance extraction setup, utterances are encoded using some method and represented in a graph by nodes. These nodes are connected by undirected edges with weights given by the connected nodes' cossine similarity. An eigenvector centrality measure, e.g. PageRank [64], is then run over the graph to score utterance nodes. The highest scoring utterances are iteratively output as the summary, with some methods normalizing for factors such as length. Examples of encodings used include pre-trained recurrent models [43], continuous bag-of-words [60], and TF-IDF [74].

Some works further refine this approach by first clustering nodes into n groups, where n represents the desired output length in utterances, and picking the utterances closest to the cluster centroids as members of the output summary [38]. In others, many utterances near the centroid are taken as mini-summaries and a shortest path is approximated between them similarly to in multi-utterance compression [89]. Beam search [94] has also been used to iteratively add utterances near these centroids to the summary and instead optimizing for maximal marginal relevance [39].

3.1.3 Neural Compression

Despite the fair amount of previous work on graph-based approaches to unsupervised summarization, neural approaches to the problem are still very new. The only

recent work [33] approaches unsupervised summarization neurally using a bidirectional LSTM autoencoder with attention [11, 85]. The model is fed noised inputs and trained to generate non-extraneous tokens by maximizing the log probability of the original, unnoised input. Noise is introduced to each training sample by appending randomly subsampled tokens and shuffling the ordering of its tokens. In order to improve fluency and grammaticality as well as control output length, the authors use InferSent embeddings [26] and a scalar "length countdown" feature in their model's decoder. The results of this method were on-par with standard rule-based baselines. Their qualitative evaluations, though, demonstrated competitive performance against "a trained supervised model" for grammatical correctness and information retention.

3.2 Summary Evaluation

Fully evaluating summaries produced by text summarization models is difficult due to the ambiguous nature of what makes a summary "good" [42]. While it is generally agreed upon that at the very least summaries should be shorter than the original while maintaining its central information, other aspects such as fluency, information-density, precision, and recall could be deemed as equally or more important than these two criteria. Unfortunately, an unsupervised way for evaluating summaries across several different analysis perspectives has still yet to be discovered. Modern metrics for evaluating summaries therefore focus upon comparing human-created summaries of a snippet to those generated by a model. The two most popular of these are `ROUGE` [55] and `METEOR` [12]. These are typically supplemented using the qualitative evaluation procedure for measuring fluency and information retention introduced by [90].

`ROUGE`, short for Recall-Oriented Understudy for Gisting Evaluation, is a recall-oriented metric that compares n-gram overlaps between the model output and the reference summary. This is evaluated separately on a variety of n-grams, e.g. $n \in \{1, 2, 3, 4\}$. In these cases the metric is denoted by `ROUGE-N`. The longest common substring can also be used as an evaluation measure, `ROUGE-L`, as well as the skip-bigram plus unigram based co-occurrences statistics, `ROUGE-SU`. The latter of these

two is intended to allow for insertion of words between bigrams seen in the reference summary. ROUGE has been shown to correspond well to human evaluation on at least one news summarization dataset [56].

Two of the big criticisms of ROUGE is that it only measures word overlap as opposed to semantic meaning and that it does not emphasize precision. While workarounds to the former point are currently being actively researched using the advent of utterance embeddings [81], the latter point was actually intended as an alternative to a precision-focused metric called BLEU [66] that is commonly used for machine translation. Since BLEU does not always extend well to summarization evaluation as it tends to favor very short outputs [56] and was meant to be used for utterance-level evaluations, METEOR (Metric for Evaluation of Translation with Explicit ORdering) was created.

METEOR uses the harmonic mean of unigram precision and recall, with recall weighted higher than precision, to judge generated model outputs compared to a reference. It has the notable feature of stemming and synonymy matching to account for word families and was originally demonstrated to have very strong (0.964) correlation with human judgments of outputs at the corpus level. While it is used for automatic summarization evaluation as a complement to ROUGE, it is important to note that it was originally created for evaluating machine translation systems. It is helpful and worth using in text summarization contexts, but was ultimately not designed with this particular use case in mind.

As an alternative to these quantitative measures, Turner and Charniak introduce a procedure for evaluating generated summaries qualitatively in [90]. They randomly selected outputs from their model to be presented to human "judges." These judges were asked to rate the outputs on their grammaticality and information retention relative to the original. Judges scored outputs using a scale ranging from 1 to 5, with 1 being poor and 5 being excellent. They note that while both grammaticality and information retention are somewhat arbitrary, the latter is implicitly tied to the idea of information importance as well. This makes the measurement somewhat unreliable given the variability across judges.

3.3 Speech Summarization Datasets

While popular speech summarization datasets exist, they are from narrow domains of meeting transcripts and do not consist of primarily colloquial opinion. Nevertheless, I outline two of the most well-known of these datasets here. Both of these are too small to be used for training text-based neural networks and seem to be more intended for finding relationships between physical gestures and speech than speech summarization.

The first is the Augmented Multi-party Interaction (AMI) Meeting Corpus. This dataset consists of 100 hours of meeting recordings [19]. The data consist of orthographic transcription and annotations for verbal and non-verbal phenomena such as dialogue acts (gestures) and head movements. Two-thirds of the data were collected for an explicitly designed environment in which four subjects roleplayed different position in a design team. In this roleplay, subjects collaborated on a project that was completed over the course of one day. The remaining third are from naturally occurring meetings over a range of domains. AMI's test set consists of 20 transcriptions for which a human annotator wrote an abstractive summary of 290 words on average. The average WER of this dataset is 36%.

The other is the International Computer Science Institute (ICSI) Meeting Corpus dataset, created by researchers from University of California, Berkeley [50]. It is similar in nature to the AMI dataset except in these data speakers were recorded from close-sitting microphones. These data were collected from a variety of speakers in 75 naturally-occurring meetings over a three-year period. In the end, 72 hours of audio were transcribed. The average abstractive summary length for the ICSI consists of 670 words and its test set of just six meetings. The WER of this corpus is 37%.

3.4 Analyzing Opinion on Public Talk Radio

Opinion-based analysis of talk radio is a research topic that has received a large amount of attention from the political science community since the 1940s. In the

interest of brevity, I consider five studies on analyzing talk radio for public opinion in this proposal. Additional studies of interest that are not presented here include [79], [45], [91], [44], and [46].

Armstrong and Rubin [8] investigated why callers and non-callers listened to talk radio. They found that callers were less mobile in their everyday lives, thought personal communication to be less rewarding, and felt talk radio was more important in their lives than non-callers. The authors suggest that talk radio provides callers with an "accessible and nonthreatening alternative to interpersonal communication." They also introduce the concept of "talk-show democracy" in alluding to the power of televangelism and influence of talk radio on American politics.

In an analysis by Barker and Knight [13], Rush Limbaugh is used as a case study to support the notion that talk radio listeners tend to agree with hosts on issues that are discussed on air. For topics that are not addressed on the show, listeners typically have independent views from the station. This suggests talk radio listenership is self-selecting. Barker and Knight also find that regular listening can be predictive of a change in attitudes when messages are negative. The same was not observed to be true for positive messages.

Yanovitzky and Cappella [95] consider the effects of call-in political talk on their audiences. Using cross-lagged correlations and a fixed-effects conditional logit model to analyze whether listeners select sources consistent with pre-existing political views or whether the hosts influence the audience, they make three key observations. First, they observe that the impact of talk radio on political attitudes over time is small. Second, they show that there is evidence of causal association between attitudes toward political figures and media reception. Lastly, they find that using political knowledge as a "surrogate" for media reception does not skew the topic of talk radio conversations toward prominent politicians in any way.

A study by Lee [36] find that radio listeners discuss public affairs more frequently with their acquaintances. They are also more willing to express opinions when asked, more active in ideological discussions, and more positive toward the value of political debates. In exploring the interaction between these observations, the author finds

that some of these tendencies are conditioned by the extent to which the listeners' political attitudes agree with that of the show. These observations are in slight opposition with the conclusions of Armstrong and Rubin [8].

Finally, Owen [63] proposes that we reconsider how we look at radio as television increasingly replaces it as America's preferred media. In an empirical study, the author observes that talk radio hosts tend to be more hostile toward political institutions than their counterparts in news and television. Owen suggests that Americans have developed a passionate and personal preoccupation with talk radio, which has changed the dynamic of how it is consumed.

The primary weakness of these studies that are otherwise rigorous in methodology is the amount of data that they used. This thesis seeks to alleviate this problem by establishing ways to model our talk radio ingest corpus. It is my hope that the methods laid out in Chapters 4 and 5 enable the conclusions of these studies to be revisited. I also hope that they enable new studies that have previously been unable to be explored with conclusions that could only be reached from analyses over thousands of hours of transcribed data. Ideas for future work are given in Section 7.3.

Chapter 4

Experimental Setup

In this chapter I provide the technical details of hierarchical compression and several noising methods that I tested to train unsupervised compression models. The emphasis of this chapter is to provide a general methodology for unsupervised neural summarization of conversational dialogue. Exhaustive experimentation of model architecture choices, loss setups, and noising method hyperparameters is left as future work.

With regard to terminology, an "utterance" is defined as a spoken word, statement, or vocal sound and a "snippet" is defined as a collection of utterances within a three megabyte segment of audio. "Snippets" and "utterances" can be understood as the transcribed speech equivalent to written "documents" and "sentences," respectively. I also refer to snippets as sequences, utterances as both sequences and items (of a snippet), and words as items.

4.1 Technical Formulation

Let $\tilde{d}$ be a given input snippet consisting of N utterances $\{\tilde{u}_1, \ldots, \tilde{u}_N\}$ and suppose g_ϕ is a neural network. Our objective is to learn a parameterized mapping $g_\phi(\cdot)$ from $\tilde{d}$ to the output sequence $d = \{\tilde{u}_I, \ldots, \tilde{u}_M\}$ of length M such that $\{\tilde{u}_I\}_{1 \leq I}^{M} \in \tilde{d}$ for all I and $|d| < |\tilde{d}|$. This notation means that we input a snippet $\tilde{d}$ of size N into a model that reduces it to size M by removing a subset of its contents. Ideally removed

items do not significantly contribute to the overall information content of d; if we let $H(\cdot)$ be some arbitrary function that measures information content, then in a perfect scenario we would have $H(d|\tilde{d}) = H(\tilde{d})$.[1] Pragmatically, it is more realistic to seek to achieve a lossy compression $H(d|\tilde{d}) \approx H(\tilde{d})$ by maintaining *almost* all of the information content from $\tilde{d}$.[2]

We will now consider hierarchical compression at the utterance-level. Suppose snippet-level compression is performed by g_ϕ and we now have $g_\phi(\tilde{d}) = d = \{\tilde{u}_I, \ldots, \tilde{u}_M\}$ with $H(d|\tilde{d}) \approx H(\tilde{d})$ and $\tilde{u}_I = \{w_1, \ldots, w_n\}$. The compression process at the utterance-level is virtually the same as at the snippet-level except with different variables. Namely, we now want to learn a neural network mapping $f_\theta(\cdot)$ such that $f_\theta(\tilde{u}) = u = \{w_i, \ldots, w_n\}$ with $\{w_i\}_{1 \leq i}^{n} \in \tilde{u}_I$ for all i, $|u| \leq |\tilde{u}|$, and $H(u|\tilde{u}) \approx H(\tilde{u})$ without violating $H(d|\tilde{d}) \approx H(\tilde{d})$. Note that this setup is fully extractive, meaning that the output cannot be a paraphrase of the input since $\{\tilde{u}_I\}_{1 \leq I}^{M} \in \tilde{d}$ and $\{w_i\}_{1 \leq i}^{n} \in \tilde{u}_I$ for all i, I.

In our case both f_θ and g_ϕ will be optimized using only a raw input corpus of text without any labels, thereby making this learning environment fully unsupervised. To learn $g_\phi : \tilde{d} \rightarrow d$, which corresponds to snippet-level compression, we inject noise into a training example d to form $\tilde{d}$ and teach the model to reconstruct the original snippet d. The amount of noise injected into d and how it is injected depend on the desired output length ranges and intuitive learning objectives. Loss for updating the parameters of g_ϕ via backpropagation can be represented using binary cross entropy.

In tandem with learning g_ϕ for sequence compression of $\tilde{d}$ to $\hat{d}$ we also learn the mapping $f_\theta : \tilde{u} \rightarrow u$ to remove verbosity corresponding to a given utterance. The training setup for learning f_θ is similar to the way we learn g_ϕ; for each utterance, we additional utterances from the training corpus that will be used to subsample noise. The loss to be used for backpropagating error for a given $\tilde{u_I}$ reconstruction can be computed similarly to before, except now the loss is multi-class to account for all

[1]Take for example the trivial example where we have the sequence $\tilde{d} = \{1, 1, 1\}$ and we know that $\tilde{d}$ can only ever be fed into e.g. an average pooling function. Then if $d = \{1\}$, $H(d|\tilde{d}) = H(\tilde{d})$.

[2]Lossy compression is effectively what we seek to accomplish in our modeling procedures given the nuances of language and interactions between utterances.

possible items in a given vocabulary. Loss can thus be computed as cross entropy for this setup.

At inference-time, unmodified snippet inputs are hierarchically compressed. The snippet-level model focuses on greedily finding utterances to add to the summary representation by considering learned representations intended to capture information content, salience, position in the snippet, and novelty relative to the current summary. The high-level objective is to minimize the difference between $H(d|\tilde{d})$ and $H(d)$, where d is the subset of utterances extracted from $\tilde{d}$ to form a summary. The utterance-level model then takes the selected utterances as inputs and compresses each of them while trying not to violate neither $H(u|\tilde{u})$ nor $H(d|\tilde{d})$. For simplicity, we assume the effects of compressing $\tilde{u}$ to u have a marginal effect on $H(d|\tilde{d})$ and thus do not explicitly enforce this in the model.

4.2 Data Preprocessing

The outputs of a non-state-of-the-art ASR system that operates at-scale are inevitably going to be noisy due to the variability of its input audio quality. While it would be ideal to be able to use all of the collected data to improve the models presented here, it is not feasible for two primary reasons. The first is that the system sometimes creates transcriptions with non-negligible word-error-rates that make the transcribed audio out to seem like gibberish. I have manually found these to be commonplace from the early months of the ingest, for stations for which data collection recently began, and following network interruptions for the data ingest. Since the quality of the data put into a machine learning model is often reflective of the quality of its outputs, I made the decision that it was necessary to prune the corpus of such items.

The second reason pertains to computational tractability. As stated previously, we are ingesting over one billion words of audio per month. This equates to a couple dozen GB of text data ingested over the past year. While I would have liked to use the full corpus to train my models, I decided that it would be better to first start with a smaller chunk of the data; this thesis is the first use of the radio ingest from a

modeling perspective and I did not want to bite off more than I could chew given the amount of time I had to complete this thesis. I considered this a big but necessary compromise in order to better establish a starting point for modeling the ingest. I leave incorporating the remainder of the corpus into training as important future work.

To prune the corpus down, I did the following. First, I spoke to the ASR system's principal contributor to check the history of system changes with respect to transcription quality. A major update was launched at the end of November that improved WER from >25% to the 13.1% reported in Section 2.1.[3] In the interest of maintaining consistency in how data were produced I discarded all data prior to that point and, intuitively, try to marginalize the variance of the effects of the system on final outputs. This decision was also inspired by my interest in seeing how well my model could handle data from well outside its temporal training range. The evaluation dataset (see Chapter 5) consisted of data between the second week of April, 2018 and the last week of February, 2019–December thus seemed a suitable month to train with without sacrificing inputs with a substantially better WER accuracy nor being too close to either end of the data ingest timeline.[4]

The month of December contained 1,384,179 non-empty, non-syndicated transcriptions. Snippets were identified as non-syndicated as those that only considering radio shows that were broadcast on one station in the full dataset. I pruned these data down to remove low-quality by first identifying snippets with high average mean word confidence ($\geq 85\%$). The mean and median confidence values for the month of December were 0.888 and 0.897 before removal and 0.904 and 0.905 after removal, respectively. This process removed 204,770 snippets in total, corresponding to approximately the bottom 15% of the month's data. While confidence scores do not formulate a valid probability distribution for the model used to transcribe, this was the best heuristic I had available for a first pass over the data. After performing this

[3]While our system is 40 times cheaper per hour than Google Cloud's Text-to-Speech API, re-transcribing six months of radio to match the new performance was a non-trivial request in terms of resources.

[4]I make no assumptions about the distribution of the vocabulary over the entire corpus.

step I was left with 1,179,409 snippets.

For the remaining snippets, I then applied the following cleaning process to each of its utterances:

1. First, I split the utterance into individual tokens. The ASR outputs consist entirely of lowercase with the exception of apostrophes for common contractions such as in "`i'm`" or "`she's`", so removing non-alphabetical characters was not necessary.

2. I then converted sequentially written numbers (e.g. "`one five five five three two four`") into the single special token "`##`". Sequences of numbers separated by a single connector such as `and` were further concatenated to be one token. I took special care to include transcription errors such as `for` for four and "`oh`" for zero in this tokenization process.

3. Next, I removed utterances that were classified as advertisements. Advertisements were considered to be any utterances with "`dot`" and any of "`com`", "`org`", or "`gov`" in them. Utterances with defined by seven or more numbers in a row in them, indicating a phone number, the bigram "`free shipping`" or "`while supplies`" were identified as advertisements and removed as well.[5]

4. After removing advertisements, I then removed trailing filler words from the beginning and end of the utterance. While exploring the data I found that these are typically artifacts of speakers beginning a speaker turn as another speaker ends theirs or a speaker trying to collect their thoughts. Filler words are words and phrases such as "`yeah`", "`uh-huh`", "`um well`", "`okay right`", and "`by the way`".

5. During the same pass as the previous step, I also removed all but one of consecutive duplicates of spoken words such as "`he he`", "`no no`", and "`right right right right`". This was done at the same time as the previous step so as not to incorrectly remove words that became filler phrases upon removal of duplicates.

[5]To validate this methodology, I checked 500 removed utterances by hand and found that 488 of them, or 97.6%, were truly advertisements.

6. Lastly, I checked to see if the utterance contained three or more non-stopwords from the NLTK package stopword list [16]. If it did then I split contractions such as "`i'm`" into two words e.g. "`i`" and "`'m`" and added it into the training corpus. Otherwise, I assumed the utterance to be uninformative to the overall meaning of the snippet and discarded it from consideration.

Preprocessing the data following these steps removed an additional 187,084 utterances, leaving 992,301 for the training data after removal of the 24 snippets also found in the evaluation dataset. Once utterances were cleaned, I then combined consecutive turns from the same speaker using the radio ingest's metadata acquired from the LIUM diarization package [3]. Turns were combined iteratively until the turn reached 56 tokens in length, after which a new turn was started. Utterances longer than 56 tokens were left alone.

In preliminary experiments, cleaning the utterances in this way resulted in significantly faster learning of models with lower losses than models with no preprocessing applied to the inputs at all.[6] Several members of LSM also roughly judged these outputs to seem qualitatively better, although no rigorous quantitative analysis was performed given the aggressive cleaning strategies of other works that also deal with ASR outputs [80, 38, 89, 59].

4.3 Noising Methods

With a reasonable sized corpus of cleaned talk radio data in hand, I will now elaborate the strategies of how models are learned to hierarchically compress snippets. Their noising modification, respective learning objective, intuitive purpose[7], and hyperparameters are described as follows:

1. `Identity`: Map an input through the identity to itself, making the output unchanged identical to the input. The objective is thus a simple reconstruction.

[6] Besides preprocessing, all other variables (e.g. seed, architecture, hyperparameters, data quantity and alignment) were left untouched.

[7] I make no claims as to whether the models actually learn their intuitive purpose.

The idea here is to learn to not compress short, information dense, and already fluent sequence, as well as learn better representations of what a natural sequence looks like. There are no hyperparameters associated with this method.

2. `Shuffle`: Append additional items to the sequence and randomly shuffle their order, optionally keeping n-grams of items together. The learning objective is to reorder the original items while ignoring added noise. This is intended to give the model a strong sense of sentence fluency and order. The hyperparameters for this method are the proportion of the original length that should be added as noise, denoted k, and n for the number of consecutive shuffled items to be kept together.

3. `Intersperse`: Randomly insert additional n-grams of items throughout the sequence, optionally weighting the frequency of different n-gram insertion lengths. Since humans complete the fully extractive compression task by dropping out words, I hoped that this method would teach the model to mimic an expert's behavior. The hyperparameters for this method are the proportion of the original length that should be added as noise, denoted k, a list of weights for sampling n-grams that up the noise, and a list of possible n values.

4. `Insert`: Randomly insert a corrupted sequence of items into the sequence at either a random index; the start of the sequence; the end of the sequence; both the start and end of the sequence; the start and end of the sequence, with the noise split evenly between the two; the start and end of the sequence, with the noise split at a random index. Optionally these methods can be randomly sampled from to choose which one is applied. An inserted noising sequence is corrupted by replacing a preset proportion of its items with random tokens or filler phrases from the vocabulary. The objective is to maintain an n-gram in the sequence while omitting the rest, which is effectively when the speaker gets to the point of what they are trying to say. The hyperparameters for this method include corruption rate p and the insertion type.

5. `Replace`: Replace an n-gram in the sequence with some given probability. The learning objective is to abstract what the replaced n-gram originally was based on its surrounding context. The intuition was that this could enable the model to be abstractive, correct the ASR system on unknown or incorrectly transcribed tokens at inference-time, or mimic a piece of the learning setup of the recent state-of-the-art language model BERT for better sequence encodings [29]. This method can also be modified slightly to mimic dropout [84]. Hyperparameters include dropout probably ϵ and n for how many consecutive items to drop out.

6. `Repeat`: Randomly repeat a seen n-gram later on in the sentence. The objective is to omit duplicate phrases, as a denoising model should know to discard information it has already captured. The hyperparameters for this method are n and the probability ϵ of dropping an item out.

7. `Multinoise`: Sampling from a one or more of the above methods to noise an input sequence. The hyperparameters for multinoising are the selected methods, their respective hyperparameters, and a list of sampling probability weights for each method.

One important design consideration that was alluded to in the description for `Insert` was to make sure that the input can only be mapped to a single valid output. For example, if inserted sequences are not noised, then the model could predict the target to be the inserted noising sequence. This is conceptually problematic because experts could not be expected to consistently agree on which sequence should be output. The above methods were designed with this in mind to avoid such a problem. Furthermore, it is important to remember that in any given setup noise must be added to the original input in order for the model to learn to compress under our unsupervised setup. As such, non-additive noising methods such as `Replace`, `Identity`, and `Repeat`[8] are intended to be used as inputs to `Multinoise` in combination with some additive noising method.

[8]While `Repeat` adds noise, it is noise that is subsampled from the sequence itself as opposed to noising sequences sampled from the rest of the training corpus. The use case of such a standalone training method is narrow, and as such I include it with `Replace` and `Identity`.

4.4 Model Architecture

The model architecture used in this thesis is inspired by the three works [61], [5], [33], and [92]. In this section I give overviews of these models and provide notes of modifications that I made to better handle issues in summarization like fluency and content selection [34]. The most important consideration of this Chapter is that the final model actually consists of two *separately optimized* models presented in Sections 4.4.1 and 4.4.2 and is therefore *not* end-to-end. A discussion of the difficulties encountered in training an end-to-end version of these two models and an argument for why an end-to-end model is not actually necessary to this task is provided in Section 6.3.

4.4.1 Self-attentive Selection

The works of Nallapati et al. [61] and Al-Sabahi et al. [5] form the basis of the self-attentive selection model presented in this section. Both of these attempt to tackle the problem of supervised summarization of news documents by greedily computing summary membership probability based on interpretable, learned scalar features such as information content, salience, position in the snippet, and novelty with respect to the original summary. The two works diverge in that Al-Sabahi et al. encode sentences using self-attention [11] at the sentence- and document-level to extract the top scoring sentences as the summary whereas Nallapati et al. do not use attention but they do have a decoder in their model that also allows for abstractive outputs. The approach I use to select utterances as summary representation items is most similar to the architecture of Al-Sabahi et al.

For ease of notation, we will now drop the $\tilde{\cdot}$ from Section 4.1. Suppose we are given a snippet d consisting of a sequence of N utterances $\{u_1, \ldots, u_N\}$ that we wish to summarize. We will represent the word sequences $u_I = \{w_1, \ldots, w_n\}$ using a bidirectional $LSTM$ [43] that concatenates the forward and backward hidden states h_t at time step t to represent each of the tokens $w_t \in \mathbb{R}^k$ where k is the hidden dimensionality of the word embeddings. Let $\overrightarrow{(\cdot)}$ a forward pass over a vector or

matrix of data, $\overleftarrow{(\cdot)}$ be a backward pass, and $[\cdot]$ denote the concatenation operation. Then

$$\overrightarrow{h_t} = \overrightarrow{LSTM_w}(w_t, \overrightarrow{h_{t-1}}) \tag{4.1}$$

$$\overleftarrow{h_t} = \overleftarrow{LSTM_w}(w_t, \overleftarrow{h_{t+1}}) \tag{4.2}$$

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}] \tag{4.3}$$

Self-attention is then applied over these hidden states to obtain a weighted representation of how much each sentence contributes to the overall sentence representations. If we let $H_u = (h_1, h_2, \ldots, h_n)$, then the word-level attention of an utterance is computed as

$$a_u = \texttt{softmax}(W_t^1 \texttt{tanh}(W_t^2 H_u^T)) \tag{4.4}$$

where W_t^1 and W_t^2 denote learned matrices of parameters. Given a_u, we can compute the utterance representations as an average-pooled sum of the weighted word-level $LSTM_w$ hidden states

$$u_i = a_u H_u \tag{4.5}$$

Using the utterance representation, we repeat this to obtain a snippet representation. Namely,

$$\overrightarrow{h_I} = \overrightarrow{LSTM_u}(u_I, \overrightarrow{h_{I-1}}) \tag{4.6}$$

$$\overleftarrow{h_I} = \overleftarrow{LSTM_u}(u_I, \overleftarrow{h_{I+1}}) \tag{4.7}$$

$$h_I = [\overrightarrow{h_{I-1}}, \overleftarrow{h_{I+1}}] \tag{4.8}$$

Let $H_d = [h_{u_1}, h_{u_2}, \ldots, h_{u_N}]$ be the hidden states of $LSTM_u$. Then to obtain a snippet representation, we compute another attentional vector

$$a_d = \texttt{softmax}(W_u^3 \texttt{tanh}(w_u^4 H_d^T)) \tag{4.9}$$

where W_u^3 and W_u^4 are learned matrices of parameters. The document representation d is then given by an average pooling over the utterance-level $LSTM_u$ hidden states

using $H_d = (u_1, \ldots, u_N)$ as input:

$$d = a_d H_d \tag{4.10}$$

Using these utterance and snippet representations, we then compute the probability of an utterance belonging to the summary y_I, given by

$$P(y_I = 1 | s_J, o_J, d) = \texttt{sigmoid}(C_J + M_J - N_J + P_J + b) \tag{4.11}$$

where `sigmoid` is the sigmoid activation function, b is the bias,

$$C_J = W_C u_J \tag{4.12}$$

is parameterized by W_C and intended to capture the information content of u_J with respect to snippet d,

$$M_J = S_j^T W_u d \tag{4.13}$$

is designed to measure the salience of u_J contextualized by d,

$$P_J = W_P p_J \tag{4.14}$$

incorporates the positional embedding p_J of u_J via an embedding matrix that intakes h_J,

$$N_J = u_J^T W_N \texttt{tanh}(o_J) \tag{4.15}$$

and is the novelty of the utterance relative to the current summary representation

$$o_J = \sum_{I=1}^{J} h_I P(y_I = 1 | h_I, o_I, d) \tag{4.16}$$

During training, binary cross entropy is optimized. At both train-time and inference-time, greedy decoding is used to compute whether an utterance should be incorporated into the summary. Only the top K scoring utterance are passed into the next

module, which consists of a pointer network and is described in Section 4.4.2.

4.4.2 Pointer Compression

Utterances that are not pruned by the self-attentive encoder module from the previous section are compressed by using a pointer network. In its essence, a pointer network is a sequence-to-sequence [85] model that has a slight modification in how it produces final outputs. Recall that in a sequence-to-sequence model we encode an input using an encoder $LSTM$ and are given a series of hidden states like H_u as the output. We then wish to decode some output using another $LSTM$ that takes as its initialization the last hidden state of the encoder. This serves the purpose of providing a history on which the decoder should condition. The decoder then autoregressively produces hidden states that are dependent on the ones that come before it. These can be mapped via a projection to some output domain. One use case is machine translation, where the encoder takes in e.g. English and outputs e.g. paraphrased English.

The only difference that the pointer network makes over this setup is to compute an attention vector between the hidden states of the encoder states $(h_{e_1}, \ldots, h_{e_n})$ and decoder hidden states $(h_{d_1}, \ldots, h_{d_m})$ by computing

$$c_i = \sum_{j=1}^{m} \alpha_{ij} h_{d_j} \tag{4.17}$$

where

$$\alpha_{ij} = \frac{\mathbf{exp}(h_{e_{ij}})}{\sum_{k=1}^{M} \mathbf{exp}(e_{ik})} \tag{4.18}$$

$$e_{ij} = a(h_e, h_{d_j}) \tag{4.19}$$

which produces a probability distribution over the encoder hidden states. In standard machine translation, this is referred to as the alignment of words between languages that is used to inform a projection to a vocabulary space. The pointer network takes a spin on this approach by sampling from this distribution directly to copy items seen by the encoder. For more precise mathematical details on the probabilistic inspirations, please refer to the original paper [92]. The only modification that I have make to this

setup is to include an embedding representation of the desired output, referred to as countdown embeddings, similarly to as is described by [33] with the only difference being that is a learned, multi-dimensional representation instead.

Chapter 5

Evaluation

In this chapter I outline my methodology for collecting the crowd-sourced evaluation dataset used in this thesis. The goal of creating this dataset was to collect a sample set of substantive snippets between April 23rd, 2018 and February 25th, 2019 that: 1) consisted of primarily opinion; 2) were from a representative subsample of the larger U.S. talk radio universe [1]; and 3) emphasized non-syndicated, and thus more likely to be inherently unique to the corresponding station's location as it cannot be seen anywhere else, content as as a way of recording local echoes to nationally trending events. Members of LSM and Cortico have this categorization to be a reasonable proxy. The surveys created for data annotation collection can be found in Appendices B and C.

5.1 Station Subsampling

The first step to creating the evaluation dataset was to identify a representative subsample of the U.S. talk radio universe using the pool of stations available through our ingest. I did this by first identifying metrics by which to compute the difference between a subsample of our stations and the larger universe. I gathered metadata statistics from [1], which consists of an aggregation of federal regulatory filings for almost all radio stations in the U.S. I narrowed these metadata fields down to station format ("Talk," "News/Talk," and "Public Radio"), geography ("South," "Midwest,"

Statistic	Utterance-level	Snippet-level
Count	668	275
Min Length	6 (4)	5
Median Length	38 (15)	31
Mean Length	38 (17)	34
Max Length	79 (63)	121
Min Compression Rate	0.15	0.02
Median Length	0.46	0.08
Mean Compression Rate	0.47	0.11
Max Compression Rate	0.95	0.60

Table 5.1: Rounded basic statistics for the targets of the evaluation dataset collected and used in this thesis. Utterance lengths in parentheses refer to the references and source input otherwise. Compression rates at the utterance-level are averaged over all three available gold references.

"West," and "Northeast"), battleground state[1] ("True" or "False"), and whether the station city type ("Small", "Large"). This reduced the talk radio universe from 17,188 stations to 1,552. The metadata corresponding to the stations in our ingest system stations as of February, 2019 were then pulled totaling 157 stations and compared against the metadata for the full set of 1,552 stations.

In accordance with the number of stations that were selected for ingest when the system was first created, I decided that 50 stations would have representation in the evaluation dataset. To select the most representative subsample of the 157 available, I initially modeled the problem similarly to the Facility Location Problem with a capacitated facility location formulation [25]. This framing is useful for Boolean decisions variables for inclusion constrained by additional scalar variables that represent the fraction of the total demand satisfied by an assignment. It is also useful for when "facilities" (stations) cannot be assigned when they are not open. The translation of this to our context is to include or exclude stations based on how much they contribute to the representation across various features (station metadata) while also accounting for whether the station is being ingested in a particular week.

[1]For my purposes, I considered these to be Nevada, Arizona, Colorado, Iowa, Wisconsin, Michigan, Ohio, Pennsylvania, Virginia, North Carolina, Florida, and New Hampshire.

Since this problem is NP-hard, only approximate solutions are available using popular integer linear programming (ILP) solvers such as Gurobi [30, 4]. Upon analysis of the results, I found the representativeness unsatisfying–features were either heavily over- or under-sampled with respect to the radio universe and the approximate solution was not very good quality as measured by the L1-norm between the two. Although weighting the features occurred to me as an option to improve the solution, I decided against it as it seemed like it would introduce strong bias into the station selection results.

To bypass these issues I ran ten million trials[2] of random subset selection and recorded results that beat the previous mean squared error minimum difference across all metadata features. To make this setup comparable to that of the facility location problem I pruned stations from consideration with less than 30 weeks of ingest history. This turned out to yield a much better solution to the station assignment problem that that of the approximated solution using ILP, so I took this subset as the 50 station subset of our 157 available stations that would be included in the evaluation dataset. See Table 5.2 for more final metadata statistics.

5.2 Week Assignment

With stations selected, the next task was to assign them weeks. Weeks start on Monday and refer to all content up until the previous Sunday. For example, April 23rd, 2018 is the first week of the evaluation dataset and covers all content since 12:01 AM on April 17th, 2018; February 25th, 2019 is the last week and covers all content since 12:01 AM on February 19th. Since I had a subset of 50 stations, I initially thought to assign each six non-overlapping weeks from which I would find evaluation snippets to create a round numbered dataset of 300 total snippets that could be easily divided into validation-test splits.

After visual inspection, I discovered that such a solution did not exist since early

[2]For computational context, ten million trials took only a few hours to run on an 8-core machine. While this is an almost negligibly small portion of the total possible 157 choose 50 (2.048e42) choices, I felt satisfied with its subset selection.

Metadata	Radio Universe	Evaluation Stations	Full Ingest
Number Stations	1,552	50	157
Talk	0.15	0.14	0.17
News/Talk	0.35	0.42	0.51
Public Radio	0.50	0.44	0.32
South	0.32	0.30	0.34
Midwest	0.25	0.20	0.14
West	0.29	0.36	0.31
Northeast	0.14	0.14	0.21
Battleground	0.30	0.30	0.39
Not Battleground	0.70	0.70	0.61
Small Population	0.74	0.64	0.48
Large Population	0.26	0.36	0.52

Table 5.2: Final metadata statistics that were obtained by minimizing feature-wise L1-norms between the Radio Universe and randomly sampled subsamples of 50 stations from our radio ingest. Category groups implied by column have proportions that sum to 1.00 for each column's corresponding rows with the exception of "Number Stations."

on in the ingest several weeks had very few stations being collected. I thus removed the first two weeks of April 2018 from consideration such that the evaluation data considered only weeks from April 23rd, 2018 onward. Given that I imposed a deadline for additional data inclusion on February 25th, 2019[3] so as to aid with my research's version control, the task thus became to assign the six weeks to 50 stations from within these date constraints. I did this using a constrained resource optimization setup.

In my first pass over this problem, I attempted to enforce an exact constraint on six to both the stations in the subset and the weeks available for each of the station's ingest history. I found such a strict solution criterion to be very difficult to optimize. Even after several days of searching on an 8-core machine, a solution had still not

[3]This is the date of the latest monthly radio dump that I received from the ingest's principal contributors before I began finalizing the contents of this thesis.

be found. Instead of parallelizing across more machines I instead opted to relax my constraints. By keeping the exact constraint of weeks being assigned to six stations each but optimizing instead for a minimum of four to a maximum of seven week assignments per station, I was able to find a solution in less than one minute. Since this was only marginally less representative of the talk radio universe upon weighting station contributions by number of weeks represented, I kept these assignments.

The last constraint I applied to the evaluation dataset's week-station assignments had to do with my desire to also capture local echoes to nationally trending events (defined in Section 5.3) in the snippets. To do this I use the same setup as assigning weeks to stations but instead optimized for Boolean assignments to each of the six entries in each of the week groups. I enforced a minimum of one of the weeks have to pertain to a nationally trending event with a maximum constraint of two. An exact constraint was not used here because sometimes these events dominate national media coverage for multiple weeks at a time. Finding solutions under these constraints took only a few milliseconds.

5.3 Content Filtering

Sections 5.1 and 5.2 culminate in a three columns of 300 rows consisting of weeks, each repeated six times, in the first column, stations assignments for the week in its row in the second column, and Boolean values in the third column indicating if that station-week should (two of the six weeks) or should not (four of the six weeks) pertain to a nationally trending event. Provided these assignments, the goal was then to filter the corpus for opinion-related content for these stations and weeks. To do this, we computed locally trending terms for each of stations with respect to that week.

Using a linearly interpolated trigram language model with weights set via expectation maximization [51] and the upstream ASR system's full vocabulary, locally trending terms were computed by taking the `argmax` over the vocabulary for the probabilities of words for the current divided by the sum of their probabilities over

the previous four weeks:

$$\underset{s,t\in|V|,w_i\in W}{\text{argmax}} \frac{P(t|w_1)}{\sum_{i=1}^{4} P(t|w_i)} \tag{5.1}$$

where $|V|$ is the size of the language model's vocab, i is the number of previous weeks w_i for which we compute the probabilities with respect to term t, and W is the set of weeks that radio ingest covers for station s. If the previous four weeks were not all being ingested, then we compute the denominator take all previous available weeks in the four-week range.

The idea behind computing locally trending words in this way is that if a term's probability spikes dramatically in a given week relative to the previous four weeks for a given station, then it is likely being associated in a new way than it was previously. This simple method turns out to be strong qualitative method for identifying events that are surfacing locally, which was a desirable characteristic as I was interested in capturing a strong presence of local issues in the evaluation dataset. Examples of locally trending terms can be found in Table 5.3. I therefore used this as a pruning mechanism for reducing the corpus in order to capture local voices in the evaluation dataset.

WHKT	**WHMQ**	**WHO**
"beans"	"the_terrorist"	"an_angel"
"blah"	"bulldogs"	"anxiety"
"bullying"	"care_act"	"big_finish"
"informal"	"freezing_rain"	"bob_evans"
"problem_finding"	"in_west"	"campagin_finance"
"security_council"	"pittsburgh"	"impeachment"
"watson"	"swatstika"	"violations"

Table 5.3: Unsorted examples of locally trending terms during the week of December 17th, 2018 for three stations. top-terms such as these were used to filter the corpus to narrow the scope of the annotation process.

To find nationally trending events, I normalized these top-terms across all stations in the radio ingest. I did this by first reformulating the vocabulary for each week seen in the evaluation dataset as all seen top-terms across all 157 stations in the radio

ingest. With this newly defined vocabulary of top-term phrases, I then computed week-wise probabilities over their respective counts summed across all stations and recomputed Equation 5.3. Examples of nationally trending phrases can be found in Table 5.4.

Given the transcriptions of the entire radio ingest I filtered for snippets containing at least one locally- (for that station) or nationally-trending (across all stations) term provided the station-week tuple could be found in the evaluation dataset. I then manually curated the corpus for week-station-snippet assignments for snippets with non-trivial conversational dialogue. I considered such dialogue to consist of long exchanges between call-in speakers and radio show hosts, the presence of multiple speakers interacting with one other, monologues that did not fit the speaking style of well-known talk radio hosts such as Rush Limbaugh, and relatively infrequent top-terms. I also tried to limit the overlap of top-term selections between stations so as to diversify the content of the evaluation corpus.

02/04/2019	**02/11/2019**	**02/18/2019**	**02/25/2019**
"roger_stone"	"howard_schultz"	"fairfax"	"declaration"
"venezuela"	"roses"	"lieutenant_governor"	"manafort"
"convington"	"ralph_northam"	"enquirer"	"paso"
"indictment"	"flowers"	"the_lieutenant"	"doppelganger"
"yo"	"treaty"	"sexual_assault"	"ha"
"philips"	"polar_vortex"	"new_deal"	"in_el"
"transgender"	"minus"	"black_face"	"beds"
"nathan"	"starbucks"	"allegation"	"andrew_mccabe"
"nicolas_maduro"	"valetine's_day"	"cosmology"	"mccabe"
"native_american"	"rowe"	"inaugural"	"omar"

Table 5.4: Examples of the top ten trending terms across all stations in our radio ingest for weeks in February 2019. These terms were used to find national events in the corpus during the annotation process. They are ordered from highest normalized score (top) to lowest normalized score(bottom).

During this selection stage, I discovered that several of the assigned station-weeks were not actively transcribing data due to aforementioned network difficulties. This caused 21 snippets to be recorded in the corpus despite being nearly empty, which

bypassed my check for which stations were being recorded in which weeks. Recomputing the assignments for these weeks following the constraints previously imposed proved to be too convoluted and I realized this artifact too far into the process to turn back. As such, I excluded them from the dataset and had the remainder sent to Rev for professional transcription [2]. After receiving back the transcriptions I realized an additional four were not as content-dense as I had initially thought. I opted to exclude these as well for a final evaluation dataset size of 275 snippets.

Curating the corpus for the final 275 snippets satisfying one or more of these criteria required a large time investment that I estimate to have taken about 14 hours, or 3 minutes per snippet, while skimming. During curation, I was not able to discern a pattern between selected snippets during this selection process. Exploring this further would be important future work as it would enable semi-supervised learning on our radio corpus and, although outside the scope of this thesis, brings up interesting questions about which features a strong model emphasizes most. Spending this time with the data influenced many of the preprocessing steps described in Section 4.2, noising methods outlined in Section 4.3, architecture design choices from Section 4.4, and future work suggested in Section 7.3.

5.4 Annotation

Once I had the 275 evaluation snippets transcribed, I began crowdsourcing annotation labels by launching two separate surveys on Amazon Mechanical Turk (MTurk) [47] to find utterances to serve as summaries and compress these utterances by removing words. In this section, I will describe these surveys, the inter-annotator agreement reliability of their collected data, and my thought process behind various user experience decisions. The full surveys can be found in Appendices B and C.

The first of the annotation task consisted of two parts: 1) identifying utterances that contain opinion, and 2) selecting two to three (up to the user to account for variable snippet length) of the identified utterances believed to be most representative the original snippet. I paid annotators $0.85 per task for snippets containing less than

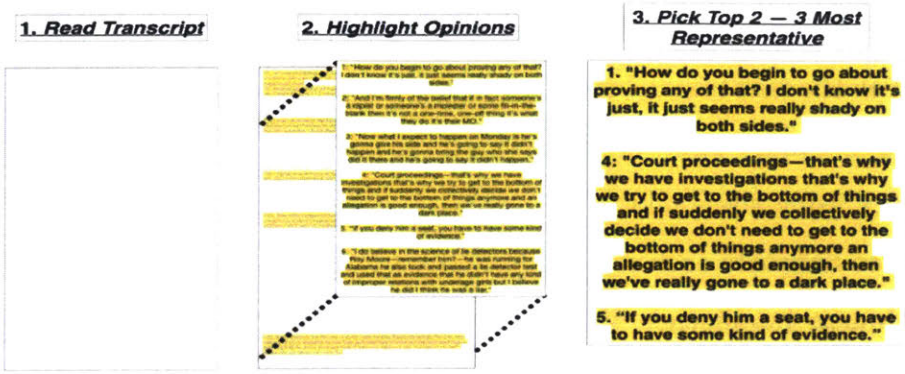
31 utterances, and $1.00 per task for snippets containing more than 31 utterances because this was the median snippet length across the data. These payment values were based on the task taking between four and eight minutes to complete, on average, and three preliminary pilot user studies. After catching all readily identifiable bugs in the survey using these pilots, I then had MTurk automatically assign five unique annotators to annotate each snippet. Annotators were required to have at least 100 previous Human Intelligence Tasks (HITs) with a minimum 93% to participate in the survey. These minimum requirements were set to select for more experienced MTurkers given the difficulty of the task.

To teach MTurk annotators how to do this task, I presented them with the instructions found in Figure 5-2. They then proceeded to take one easy and one medium difficulty attention check to determine if they read the instructions. They were required to answer these questions correctly before proceeding to the next question. If they got it wrong, they were given a short explanation that gave them a strong hint as to the answer. If they failed both of these attention checks, though, the survey was immediately terminated and they were not given a code to receive compensation. I chose to terminate the survey rather than reject their answer so the the workers' ratings were not hurt due to this atypically difficult MTurk task.

After successfully passing these initial attention checks, MTurkers then were instructed to read through a radio snippet transcript and click on any utterances containing to opinion. The definition of what opinion consists of, given in the instructions screen, was repeated for their convenience. Two additional, randomized attention checks were randomly inserted in the transcript to continue quality assurance. These attention checks were not mentioned to the worker beforehand but self-explanatory as to their intent. Examples of randomly inserted attention-check utterances can be found in Table 5.5. Survey participants were required to identify these to proceed to the next step. All workers that attempted to proceed without having all attention-checks marked were flagged for manual review.

The final step of this task was then to pick the top two or three most representative utterances of those selected. Language use to describe the process was copied verbatim

To summarize, you will use the following process:



1. **Read** the transcript.

2. **Highlight** all sentences that contain opinions, interpretations, viewpoints and/or beliefs, regardless if you believe their content to be true or agree with them. This may sometimes include statements on behalf of other people, for example "a lot of people think filing taxes is annoying."

3. **Pick** the top 2 or 3 most conceptually important, representative sentences with respect to the discussion happening in the original transcript.

Figure 5-1: Instructions presented to MTurk crowdsource workers to identify opinion and representative utterances.

Speaker 0: Yeh, well, I have a family that needs the bonus–I can't fail this attention check! Especially not if I read this and know to click on it.
Speaker 0: Please don't give me any of that non-sense and click this to let me know you really read this transcript.
Speaker 0: Yeh? Well I think that you should click this to let us know you are not randomly answering.
Speaker 0: Well who really knows anything anymore, are you even paying attention? If you are, check this box and keep reading.
Speaker 0: Ok, I agree with that but I am not sure about attention checks so you should click this if you wish to be paid.

Table 5.5: Examples of utterances randomly inserted into the transcript for use as attention checks.

throughout the task so as to not confuse the annotator. I allowed annotators to choose the number of answer choices they would like to submit in this part to allow for variable length and redundant snippets. Although given a choice, the number of answer choices was required to be within this range.

The outputs of this annotation process consisted of two sets of binary labels for whether 1) the utterance was identified to contain opinion-related content, and 2) the utterance was representative of the dialogue from the original snippet read. To measure how defined the task was, I computed the Fleiss' κ inter-annotator agreement [35] for both of these annotation-levels. In doing this, it was important to keep in mind the political nature of most of the content being annotated—it has been shown that a person's notion of veracity shifts as ideological beliefs become part of their analysis [86].

To accommodate for this phenomenon as well as the ambiguity of the task and its definitions, I considered all possible combinations of annotators and computed the maximum Fleiss' κ. By looking at the strongest agreement between annotators instead of solely the whole group as an aggregate, I intended to consider the question of how closely a subgroup of n people can get to agreement on what is opinion rather than asking them if they agree with *my* definition of opinion. I also report the κ for the full group ($n = 5$) for full transparency. Using this methodology, I observed the average κ values across the full evaluation dataset listed in Table 5.6.

Nearest n annotators	**Is Opinion κ**	**Is Representative κ**
2	0.65402	0.69880
3	0.49353	0.45897
4	0.37132	0.29436
5 (all)	0.26013	0.18767

Table 5.6: Fleiss' κ inter-annotator agreement values for opinion identification and representative utterance selection.

While Fleiss' κ is difficult to interpret as only rules of thumb are provided in the paper. Nevertheless, the original authors state that $\kappa < 0$ is "poor agreement," $0.01 \leq \kappa \leq 0.20$ is "slight agreement," $0.21 \leq \kappa \leq 0.40$ is "fair agreement," 0.41

$\leq \kappa \leq 0.60$ is "moderate agreement," $0.61 \leq \kappa$ 0.80 is "substantial agreement," and $0.81 \leq \kappa \leq 1.00$ is "almost perfect agreement." Since κ is higher when there are fewer categories [83] and the average snippet length is 34 utterances, these results can be understood to be as at least strongly reasonable, if not more than expected, inter-annotator agreement across both evaluation criteria.

From the top utterance annotations gained from this process, I let the representative utterance be the one with the largest number of annotators labeling it as a top utterance. Representative utterances were iteratively selected in this manner in decreasing count order until no utterances were left, such as in the case of all annotator choices being shared between just two utterances, or the summary vector reached length three. Ties were broken by random sampling and the remaining elements of the count group were returned to the iteration queue to be added until the aforementioned summary conditions were met. Of the 275 snippets, 157 had summaries consisting of three utterances and 118 had summaries with just two utterances.

Notably, single-turn speaker utterances under 8 tokens and over 85 tokens in length after splitting on white-space and before `[.,!?;]` were not considered for top utterance selection. This represented only 3% of the total eligible data and was enforced such that there were no unreasonably short or long utterances in the dataset to be compressed. The final length distribution after roughly matched that of the canonical summarization dataset Gigaword, which I believe to be supportive evidence that such removal was principled.

After this handling, unpruned representative utterances were sent to MTurk to be compressed by three annotator at a rate of $0.25 per utterance. Data were tokenized in the same way as described in Section 4.2 with the exception of utterance discard and token removal, which were skipped. Consecutive numbers were still tokenized in the same way and `[.,!?;-+]` were split into their own tokens. While `[.,!?;-+]` were presented to annotators in the survey to avoid issues with readability stemming from lack of punctuation, they were removed in postprocessing so-as-to match the format of the talk radio training data.

As in the first task, annotators were required to have completed at least 100 HITs

Task: You will shorten a paragraph by highlighting words that are *absolutely necessary* to retain the paragraph's **fluency** (A.K.A. grammaticality) and **original information**. Shortened outputs should be as concise as possible with respect to these criteria.

Punctuation are not considered to be important (such as in the example below), **multiple sentences may be combined**, and **entire sentences may be deleted** if you consider them to be extraneous.

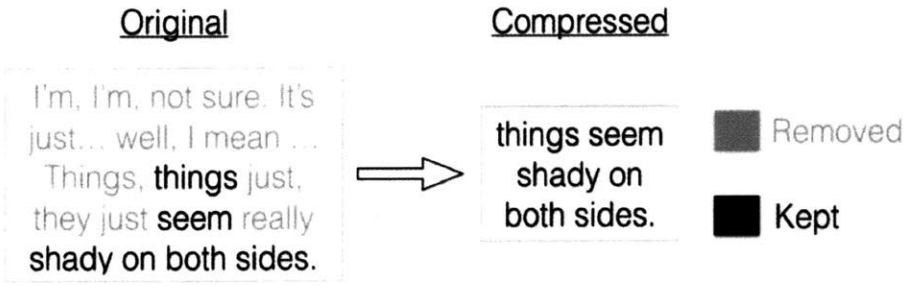


Figure 5-2: Instructions shown to MTurk crowdsource workers for the utterance compression task.

with a success rate of 93% or more to be eligible to participate in the survey. Instructions given to how annotators should complete the task can be found in Figure 5-2. After reading these instructions, annotators were then asked to 1) read the utterance they were about to compress, 2) provide a short (one to three word) summary of the utterance as an attention check, and 3) compress the utterance down by selecting words that were not necessary to retain fluency and grammar. Language use was kept as consistent as possible across the survey just like before.

I analyzed the the Fleiss' κ similarly to for the first survey and found that the κ values between the nearest 2 and 3 (all) annotators were 0.51523 and 0.27307, respectively. This time, I computed the κ for groups to account for the room for interpretation of task instructions. For example, some annotators made the utterances

sound like news titles (e.g. "government notice to appear issue redacted," "images gaza jerusalem illusrates contradictions there") as opposed to the desired outcome of shorter, still fully fluent versions of the original. Budget constraints prohibited me from having these re-annotated. Overall, I found these are suitable values because, similarly to before, there values were computed across 34 categories (utterances) on average.

Chapter 6

Results

In this chapter I will quantitatively evaluate my proposed model on the dataset described in Chapter 5 and benchmark this performance against that of several strong baselines from the literature. Since popular metrics used for evaluation of summarization methods are controversial within the NLP community, I provide qualitative explorations of my model. Examples of model selections and compressions can be found in Appendix A.

6.1 Specifications

The model that I report consists of two separately optimized modules. The first of these is the self-attentive snippet encoder, which greedily scores and snippet utterances to determine how much they contribute to the summary representation. Snippets are passed into this module first to prune candidate input utterances. The second module compresses these utterances down and is the modified encoder-decoder pointer network described in Section 4.4.2. Both models were trained for three full epochs over the December, 2018 training data of 992,301 snippets. During these training runs, I optimized the negative log likelihood of the unnoised data using Adam [53] with a learning rate of 1e-4, default settings, and no custom learning rate scheduler. I did not use early stopping as I did not observe any tell-tale signs of overfitting [78] on the loss or validation set.

Module-specific learning environment hyperparameters are as follows. The self-attentive utterance encoder consisted of a three-layer bidirectional LSTM with hidden dimension size 256; the forward and backward pass concatenation dimensionality thus equaled 512. The utterance-level attention mechanism had a hidden dimension of size 512. The inputs to the utterance encoder were 300-dimensional embeddings of tokens from a vocabulary size of 25,000 and three special `UNK` tokens to map back to for copying to handle out-of-vocabulary (OOV) items. GloVe [69] was used to initialize the embeddings and they were then fine-tuned during training. Vocabulary not found in the original GloVe embeddings such as named entities and station names were randomly initialized. The vocabulary size and number of special tokens were set via a preliminary analysis that observed more than 99% of training corpus utterances did not exceed three OOVs with these settings. Vocabulary items were based on the 25,000 most frequently occurring unigrams in the radio ingest corpus across all months of data collection.

The self-attentive snippet encoder was separately parameterized, jointly optimized, and had the same exact architecture as that of the utterance encoder minus the word embeddings. The utterance and snippet representations acquired from these two encoders were ultimately utilized by the scoring mechanism. The scoring mechanism consisted of one 512-dimensional feedforward linear network for each of its information content and position feature approximators. Utterance position was embedded into a 64-dimensional representation. The other two components, salience and novelty, were 512-dimensional bilinear feedforward networks. All feeforward networks mapped to a scalar output, contained no non-linear activation functions, and had their bias terms omitted from computation. The total number of parameters in the full self-attentive snippet encoder module was 18,374,892.

To noise the self-attentive snippet encoder inputs, I used a noising rate of 7.00 to 10.00 times the original length. Under this rate a snippet of length e.g. five would be noised to length 35 to 50 depending on the value that was uniformly sampled from the noising rate range. Noise was applied by randomly inserting utterances from the training corpus into the snippet so that the original, target utterances were spread

out. Before noising, snippets were truncated by randomly selecting an index between the start and end of the snippet and extracting the index and the next one to four utterances following it. The number of utterances included in the truncated version of the snippet was thus sampled from the integer range [2, 4]. I used a batch size of 20 snippets.

Truncating the snippet was inspired by works such as [77, 33, 5] and intended to encourage the model to learn to output down to the approximate target output length of the validation dataset. The conceptual idea of doing this is that if a model is trained to map noised sequence that were noised to be seven times their original, then the model is learning to compress inputs to $1/7 \approx 0.1429\%$ their original length at inference-time. By letting the noising rate be within [7.00, 10.00], I meant for evaluation snippets to be natively mapped to 3 or less utterances. Since the average snippet length in the training corpus was approximately 34 utterances, I truncated them to an input size of two to five utterances such that the aforementioned noising range would noise them back up to around this length. I found this one to work best while experimenting with different noising rate bounds.

The utterance compression pointer network had the same dimensional sizes as the self-attentive network, namely a 256-dimensional three-layer bidirectional encoder with 300-dimensional fine-tuned GloVe embeddings that mapped to the same vocabulary of size 25,000 plus three special `UNK` tokens. The decoder of the pointer network was a three-layer, 256-dimensional LSTM that had 64-dimensional countdown embeddings concatenated to each of its inputs. These countdown embeddings were regularized with a dropout [84] probability of 0.20, which I found to slightly improve performance. The attention mechanism used to form a distribution over the encoder LSTM states to sample output token was 1024-dimensional. I noised utterances by appending subsampled words from two randomly sampled training corpus utterances until the utterance was 1.60 to 2.50 times its original length. The tokens in the utterance were then completely shuffled. This was done to learn a model that approximately outputs the desired compression rate at inference-time of half or less similarly to [33]. I used a batch size of 128 utterances.

For evaluation, I benchmark across a variety of strong rule- and graph-based methods. The rule-based baselines consist of exhaustive combinations of typical baselines used for extractive and compression-based text summarization to match my hierarchical setup. These baselines are described more fully in Section 3.1. The most notable of these baselines is `Oracle` + `AllText`, which represents the performance ceiling for purely extractive methods that do not perform compression at the utterance level as it provides 100% recall for extracting representative utterances.

Beyond these rule-based algorithms, I further compare against LexRank[32], ClusterRank [38], and multi-utterance compression [18] using open-source implementations that follow the original papers' technical details. I was also interested in comparing against stronger models such as [80] and [37], but found their out-of-the-box implementations to be too highly tailored to their respective domains; modifying them to be compatible with my inputs would have stripped away too many of their components to expect them to still perform well.

6.2 Analysis

In this section I quantitatively evaluate using `ROUGE`[1] [55] and `METEOR`[2] [12]. I also explore the model's outputs and use methodology from Turner and Charniak [90] to qualitatively evaluate the model. Recall from Section 3.2 various n-gram lengths, `ROUGE` measures the following between a generated or output compression (hypothesis) and its gold reference, human annotated compression (reference):

1. `ROUGE-N`: (e.g. 1, 2, 4): their overlap of N-gram tokens.
2. `ROUGE-L`: their longest common subsequence, thereby taking into account a notion of utterance level structure similarity.
3. `ROUGE-SU4`: the skip-bigram plus unigram based co-occurrences statistics.

[1] I use the python-rouge package implementation [87]
[2] I use the nlgeval package implementation [49, 81]

Since ROUGE only considers word overlap and two utterances can communicate the same idea despite having no n-gram word overlap at all (e.g. "it's not right" and "it is wrong" for the unigram case of $n = 1$), its effectiveness for evaluating natural language generation has been heavily debated in the NLP community. At the same time, the task I am aiming to accomplish is purely extractive and a feature of pure extraction is that model outputs can always be mapped back to the reference inputs. This alleviates many of the problems associated with ROUGE as paraphrasing is not possible. Still keeping the considerations of the community in mind, though, I have also included METEOR in the evaluations to get a more holistic picture of quantitative evaluation. METEOR measures the harmonic mean and unigram precision recall with a higher weight for recall than precision. It performs some stemming, synonym and paraphrase matching under the hood in order to account for the space of all possible alignments. These two metrics are the most frequently used in works on automatic text summarization.

Model	R-1	R-2	R-4	R-L	R-SU4	METEOR
Lead N + F8W	0.04632	0.01902	0.00250	0.04586	0.01503	0.04521
Lean N + Random n	0.05194	0.00260	0.00000	0.04483	0.01366	0.04023
Lead N + AllText	0.14033	0.08302	0.04386	0.13755	0.08161	0.13720
Random N + F8W	0.02309	0.00692	0.00052	0.02292	0.00576	0.03482
Random N + Random n	0.03280	0.00155	0.00000	0.02933	0.00862	0.03220
Random N + AllText	0.07380	0.03155	0.01522	0.07119	0.03403	0.08981
Longest N + F8W	0.03171	0.01205	0.00149	0.03135	0.00918	0.03665
Longest N + Random n	0.03660	0.00187	0.00000	0.03270	0.00861	0.03415
Longest N + AllText	0.09000	0.04511	0.02385	0.08683	0.04752	0.11568
Oracle + F8W	0.10643	0.04945	0.00674	0.10601	0.03457	0.14825
Oracle + Random n	0.12774	0.01211	0.00043	0.10670	0.03740	0.12080
Oracle + AllText	0.34587	0.25517	0.15552	0.34362	0.24203	0.52897
LexRank	0.11481	0.07506	0.04217	0.10981	0.05752	0.07429
ClusterRank	0.11001	0.06511	0.01385	0.10304	0.04959	0.06979
Multi-utterance Comp.	0.15202	0.08914	0.03415	0.12106	0.07421	0.12308
Self-attentive + PtrNet	0.22331	0.12455	0.06140	0.20696	0.14316	0.21983

Table 6.1: Performance on the test split (220 snippets) of the evaluation dataset described in Chapter 5. R denotes ROUGE for brevity. The theoretical range of this table is [0.00, 1.00]

As one can see by the reported ROUGE and METEOR scores in Table 6.1, my model outperforms all baselines used for comparison with the exception of Oracle + AllText. This makes sense since this baseline provides 100% recall on selecting representative utterances, which not a human should be reasonably expected to do. At the same time, the ROUGE and METEOR of my model are still quite low. This is due in part to the fact that in the context of my evaluation dataset, these metrics are heavily dependent on the model being able to identify the two to three "representative" target utterances from all utterances in the snippet.

A completely random recall, computed by dividing the number of representative utterances in a snippet by its length, is 10.56%. From my experiments, I observed that Lead N achieved a rounded-up recall of 30%, Longest N a recall of 21%, LexRank 20%, ClusterRank 19%, Oracle 100%, and my model 40%. The recall of multi-utterance compression was not straightforward to compute as it uses multiple utterances to generate its output. I observed during these analyses that LexRank and ClusterRank had large biases toward medium to long length utterances, which may explain their similar performance to Longest N + AllText.

I will consider how the outputs of my model fair in terms of fluency and information retention. To do this, I followed the methodology of Turner and Charniak [90] of human evaluation for natural language generation. In their method they have annotators evaluate the fluency and information retention of summaries with respect to the original. Whereas for fluency only outputs are provided to the annotator, in the latter both the input and the output are shown and in that order. Criterion are evaluated by separate annotators.

I randomly sampled 25 snippets from the test set to have four annotators score each criterion with information being evaluated first and fluency second. For information retention, the annotators were not provided context as to the what the gold indexes. They were asked how well the information in the summary represents the information in the original snippet. For fluency, the second longest (and by extension, second shortest) was what was presented to the reader of the three references available. Results can be found in Table 6.2 for my model compared to the ground truth.

	Fluency	Information
Random words	2.13	–
Random utterances	–	3.33
`Self-attentive + PtrNet`	3.69	4.02
Ground-truth	4.72	4.49

Table 6.2: Average qualitative scoring of the model compared to ground truth for fluency and information by four annotators for 25 randomly sampled test snippets.

Interestingly, my model was shown to represent information better than expected. Nevertheless, my model is still far off from the ground-truth values. This suggests that there is still optimization to be made in terms of the models representational capacity. Interestingly, though, randomly selecting utterances yields a moderate amount of information. I believe this to be due to my emphasis on content density when curating for evaluation snippets, as described in Section 5.3. My model is still far off from both the ground-truth and random benchmarks. This suggests that while the model is certainly capturing there is still optimization to be made in terms of the models representational capacity.

6.3 Discussion

Despite introducing several methods to noise inputs fed into the model, I did not find in any of my experiments a method or combination of methods that outperformed or came close to the performance of shuffle; all other methods, provided they did not reduce to a simple rule such as `F8W` as will be alluded to in the following paragraph, performed several `ROUGE-1` and `ROUGE-2` points lower. This was strongly against my intuition. I expected multinoise to provide the most fluent and robust outputs given that it provides the model the most diversity in terms of what it was viewing. While this result may be due to the constrained resources I had to test various noising sampling settings and learning hyperparameters, it is also possible that the diversity of multinoising is actually a weakness rather than a strength.

As an alternative to the pointer network utterance compression module I used in this thesis, I also experimented with generative and a deletion-based decoders. In the generative setup, the output distribution from the decoder's feedforward network is over the vocabulary instead of the encoder's hidden states as in pointer networks. While the generative setup produced sensible outputs, it had much more difficulty with long sequences and was occasionally nonsensically abstractive. To a degree this makes sense as the target is always extractive in the evaluation dataset; the hypothesis space of possible options is inherently better captured by a pointer network as its outputs will always land there. This is not the case for generative modeling.

I optimized deletion-based models using binary cross entropy to predict Boolean probabilities over each of the encoder hidden states. These were then sampled from to determine whether a token should be output or removed. This setup is as opposed to in pointer networks and generative modeling, which optimize for multi-class cross entropy.[3] A deletion-based approach seemed to most closely match that of how human annotators remove verbosity. Unfortunately, I observed that the denoising model rapidly reduces itself to be the N token equivalent of `F8W` under intersperse noising. Since the deletion setup is not compatible with many of the noising methods proposed e.g. shuffling due to its inability to modify ordering, I was unable to find a noising schema that bypassed this issue. I observed the same problem for pointer networks trained without shuffling. Generative modeling did not succumb to this same bias but still under-performed other methods.

Finally, I would like to note that I spent significant time trying to build an end-to-end version of my final model. The issue that I ran into was that the self-attentive snippet encoder and the pointer network at first seemed to prefer different batch sizes and learning rates. I tried the following methods to get the overall system to still work: only computing compression loss for utterances found in the original, truncated snippet; truncating snippets to contiguous blocks of length seven so a batch size of 20 utterances would pass a approximately 140 utterances to be compressed, thereby

[3]In my learning environments I actually optimize negative log likelihood, which is more numerically stable than cross entropy although its outputs are utilized in the same way.

mimicking the individually successful optimizations of batch sizes described above; setting temperature values for the scoring and compression contributions to the total loss to be backpropagated [28]; annealing the contributions of the loss of the scoring and compression losses across training [65]; and tying weights between the modules [71]. None of these ultimately worked.

Upon visual inspection of the magnitudes of the gradients backpropagated to the network, I noticed that the compression module received much smaller updates ($<5\%$) relative to the snippet-level encoder. Unfortunately, I did not have enough time to experiment with regularization techniques to marginalize this differential. Ultimately, I would argue that the performance of my final model outweighed my concerns with it not being end-to-end. While end-to-end systems are useful to avoid dependencies between independently run components, end-to-end systems are not absolutely necessary for successful modeling despite what is preferred by the literature.

Chapter 7

Conclusion

In this chapter I summarize the contributions of this thesis, cite limitations of my approach and the data source, and make suggestions for future work for analyzing and better modeling talk radio content. I then finish with a reflection on the unsupervised framing of this thesis and provide the rationale that motivated me to pursue unsupervised summarization of public talk radio.

7.1 Summary

In this thesis I presented a novel, compression-based method for neural unsupervised summarization of long-form conversational dialogue by training on unlabeled outputs from a radio ingest system's ASR model. I evaluated my proposed models on a novel speech summarization dataset of spoken opinion that I created via crowdsourcing, and then benchmarked my model's performance against strong rule- and graph-based approaches. Since the evaluation metrics used are surrounded in controversy regarding what they measure, I also had annotators qualitatively evaluate my model's output for fluency and information retention.

7.2 Limitations

Supervised learning for task-based natural language processing is overwhelmingly dominant in recent machine learning literature. There is a relatively small pool of related work on neural unsupervised methods from which to gain inspiration as a byproduct. While this thesis nevertheless demonstrates the potential of unsupervised NLP, it is difficult not to think that the results laid out herein would be even better if a large, high-quality labeled dataset were used for training instead given the current contents of the literature. I imagine that a model trained on a labeled dataset of e.g. one million snippets would allow for much more direct modeling of of representativeness and to what extend this task is truly feasible.

At the same time and so as not to downplay the unsupervised nature of this thesis, it is crucial to remember that the training data used were of poor quality–the input were transcribed speech from an automatic speech recognition system with a non-trivial word-error-rate of 8-30%. Machine learning is not magic; what one puts into such a system is almost certainly what one will get out. Using the outputs of another model as inputs into my own also make it difficult to disentangle the effects of the radio ingest system on my own. Although it is outside the scope of this thesis, it would have been interesting to test my methods on standard summarization datasets such as Gigaword [40, 67] and CNN-Daily Mail [20] to get a better sense of how my model performs on highly-structured data both with and without labels. There is an abundance of extremely interesting work on summarization that I regret not being able to test more thoroughly.

On the topic of resources, the total budget for this thesis was only a few thousand dollars budgeted over the course of a couple of months. Given more time, compute, and funds, I would have liked to extensively explore the effect of hyperparameters on model performance, further expand the evaluation dataset, and fully incorporate methods I had considered from current state-of-the-art summarization and generative modeling papers. I would have also liked to consider modeling on the full dataset, which I opted not to do for reasons described in Section 4.2, and I would have liked to

explore making my model explicitly opinion-specific. It can be easily argued that my model is a general unsupervised summarization model applied to an opinion-based evaluation dataset.

Another important consideration is that neither the radio ingest used to collect data as input to my models nor the data stream quality were under my jurisdiction. As a result aspects of the data such as station ingest selections, ASR quality, speaker diarization, metadata collection, and audio preprocessing and fingerprinting were out of my direct control. I used what I had and requested some additional features that I thought would help me in this study, but macro-level modifications to this system were not logistically feasible for me to execute myself in a timely manner.[1] This limitation has implications in controlling the bias of the model with respect to geographic location.

Lastly, I would like to note the difficulty of unsupervised NLP and evaluating on NLP tasks in general. The trained models proved to often be finicky to small changes in hyperparameters, such as a learning rate change of 0.0002 with all other variables remaining the same and a preset initialization seed. The validation set used only helped to an extent; models that did well on measured evaluation metrics sometimes read worse than more "poorly" performing models.

7.3 Future Work

Ultimately, the emphasis of this thesis was on establishing principled methodology for cleaning, summarizing, and analyzing data from a talk radio ingest system. Moving forward, I invite the reader to recall the beauty—and perhaps curse—of machine learning: there are a combinatorial number of possible variations that can be applied to and tested for the problems considered in this thesis. These take the form of model architecture, loss setup, hyperparameters, regularization procedures, data curation, and so-on. They could also take the form of making my model more specific to

[1]This limitation could also be considered a strength since the resultant model was not extensively fine-tuned.

opinion, perhaps by feeding it solely opinion-based content.

Additional promising avenues for future research prospects include: characterizing the linguistic structure of opinion and conversational speech "in the wild;" creating ways to systematically identify local echoes to national events using information and feature extraction; analyzing host and call-in interactions as well as the reverberant effects of syndicated content at-scale; studying framing shifts on mainstream issues and what causes them; modeling the lexical chains of entities that are mentioned in conversations and monologues; semantic segmentation of transcripts rather than segmenting based on audio clip size; analyzing how bias and automatic speech recognition quality affect model performance and robustness; enhancing speaker diarization and modeling via learned speaker embeddings for hosts and call-ins; integrating speech characteristics such as prosody into data representations; developing a probabilistic framework for incorporation of other media data streams e.g. Twitter into our model; adding stronger mechanisms of interpretability to model decision-making; extension of methods laid herein to multi-snippet summarization; combining the strengths of the compared baselines for better handling of noisy ASR input; a thorough study of how regularization impacts performance with respect to unsupervised NLP models.

7.4 Reflection

While full supervision can be effective for highly-structured NLP tasks, it imposes limitations on generalizability by presupposing the possible ways meaning can be expressed. During my time at MIT I found that setting aside conceptions of how language is "allowed" to be used lets us better capture its nuances.[2] As such, I continue to be interested in developing less constrained approaches to NLP by using unsupervised learning, generative modeling, and information extraction. Whereas supervised learning effectively overlooks an individual sentence's subtlety in favor of its nearest neighbors, these proposed frameworks enable interpretations of meaning

[2]Admittedly, this may be due to survivor bias caused by my personal interest in unsupervised methods.

by avoiding assumptions regarding language's complexity and variability. In NLP, we have mastered ascribing intent to text. I want to focus on the next step: letting it speak for itself. This thesis is an early step toward such a philosophical framing and my conviction for it has only been strengthened through the work I have done that culminated in this document.

Appendix A

Model Outputs

A.1 Standalone Utterance Compressions

These are compressions of randomly sampled utterances from the radio ingest.

Utterance	Compression
yesterday kim and south korea 's president met in the d._m._z. was president moon saying kim remains committed to a face to face	saying yesterday kim remains committed to a face to face in d._m._z.
children are together and with family but they are suffering they really need their mom the older boy is in school and has already been referred for mental health treatment because he 's just not doing as a result of the trauma	children are together and with family but they are suffering they really need their mom the older boy is in school
looking but senior care for your mom or dad but do n't know where to start	do n't know where to start senior care do
dude it 's ## degrees today sunny and hot evening scattered showers and storms expected a high of ## that the lake i ## inland from the w._t._o. breaking news center i 'm tony	breaking news center i 'm tony inland from the lake center i 'm tony showers and storms expected
kenny warner with many day schedule throughout europe peter skins newest album	newest album with many kenny warner album
start point guard tirade irving 's exodus to boston there were injuries there was a roster overhaul in february prompting ## n._b._a. analysts to call the cabs at that point a dumpster fire meaning guard j._r. smith ## game suspension for throwing soup at an assistant coach	start point guard tirade irving 's exodus to boston there were injuries there was a roster overhaul in february prompting ## n._b._a. analysts
## coach saluted <unk> worse student athletes aaron masters for basketball and shame prior for baseball press get high school ## or they 're ## graduates with distinction articles	## press get high school graduates with ## or ## coach basketball and shame they
## of the actors kept spinning you 're near phased by accident and god really that will he was he was the punk	you 're ## of the actors kept spinning that will near you
weekend south korean president moon jan met with kim jong noon and says the north korean leader still hope to meet	weekend south korean president moon jan met with kim jong noon

Table A.1: These are compressions for randomly sampled utterances from the validation split of the evaluation corpus.

A.2 Variable Length Compressions

Rate	Compression
0.1	do n't follow
0.2	problem here is even more corrupt
0.3	do n't see this is the most people thought
0.4	do n't see this is the problem here most people still take thought
0.5	problem here is the fbi but most people do n't follow it than i thought seriously
0.6	do n't see this is the problem here most people do n't follow it is even more corrupt than
0.7	do n't see this is the problem here most people do n't follow it than i thought they were but most people thought
0.8	problem here is the fbi most people do n't see this but most people do n't follow it than i thought they were but more corrupt
0.9	do n't see this is the problem here most people do n't follow it than i thought they were but most people still take that closely still take follow
1.0	problem here is the fbi wiki seriously some people do n't see this but most people do n't follow it than i thought they were but most people still take that closely

Table A.2: Compressions of the utterance `some people still take wiki seriously see this is the problem here the fbi is even more corrupt than i thought they were but most people do n't follow it that closely` for varied compression rates.

Rate	Compression
0.1	unlike his predecessor defense
0.2	unlike his predecessor defense secretary jim mattis who 's
0.3	unlike his predecessor defense secretary jim mattis said we ca n't leave syria behind
0.4	unlike his predecessor defense secretary jim mattis said we ca n't leave syria such as the president 's
0.5	unlike his predecessor defense secretary jim mattis said we just ca n't leave syria on some of the president 's decisions was quite
0.6	unlike his predecessor defense secretary jim mattis said we ca n't leave syria we just questioned some of the president 's decisions behind the scenes what we questioned
0.7	unlike his predecessor defense secretary jim mattis said we ca n't leave syria we just questioned some of the president 's decisions behind the scenes such as what we do know shanahan
0.8	unlike his predecessor defense secretary jim mattis said we just ca n't leave syria on some of the president 's decisions was quite forceful on what we do know is that the president 's decisions was quite
0.9	unlike his predecessor defense secretary jim mattis said we ca n't leave syria we just ca n't leave syria behind the scenes such as removing the president 's decisions was quite forceful on what we do know some of his predecessor shanahan
1.0	unlike his predecessor defense secretary jim mattis said we ca n't leave syria we just ca n't leave syria such as the president 's decisions was quite forceful on what we do know some of the president 's decisions was quite forceful on what we questioned

Table A.3: Compressions of the utterance `what we do know is that shanahan is unlike his predecessor defense secretary jim mattis who questioned some of the president 's decisions was quite forceful on some areas behind the scenes such as removing troops from syria mattis said we just ca n't leave syria` for varied compression rates.

A.3 Snippet Compressions

Table A.4: Outputs for compression of every utterance in a snippet with the selected (unpruned) utterances bolded. A compression rate of 0.50%. The crowdsourced representative utterance labels were 1, 6, and 7 (100% recall).

Index	Utterance	Compression
0	**it 's a very challenging job that is often the end of political careers and she was the longest serving home secretary since the second world war**	**it 's a very challenging job that is the end of political careers**
1	so i think her reputation was definitely of a safe pair of hands if you 'd like when the referendum happened in ## and produced this unlikely result	referendum was definitely unlikely result so i 'd like her reputation when the referendum happened
2	you spoke to many people who could talk about her as a leader meaning just being in the room where decisions are made and what she is like what did they tell you	decisions are made as a leader meaning in the room where people who tell you spoke

Continued on next page

Table A.4 – continued from previous page

Index	Utterance	Compression
3	people who worked with theresa may over the years defend her kind of extremely stoutly and are extremely loyal to her because she is fundamentally straightforward she has enormous stamina you ask her to be somewhere ## o'clock in the morning she will be there ready to go	enormous stamina are extremely loyal to her because she has worked with over the years defend her kind she is fundamentally straightforward ## o'clock
4	her performance in the house of commons trying to sell this deal over the last few months has been even for people who might passionately disagree with it no one disputes her honesty and her desire to do the right thing by the country	it has been in the house of commons trying to do the right thing by people who disagree with her honesty deal
5	what i members of the cabinet and the whole government are doing is working to ensure that we leave the european union with a deal and that is the way crosstalk	crosstalk leave the european union with a whole government is working to ensure the cabinet

Continued on next page

Table A.4 – continued from previous page

Index	Utterance	Compression
6	**however she 's not someone who 's comfortable with lots of voices in the room making contesting points she likes to have a very small group of people who she trusts implicitly**	**someone who 's comfortable with lots of voices in the room making a very small group**
7	**looking back on it one of the things that 's clear of her handling of brexit a phrase that really stayed with me during my reporting was she bunkered it**	**looking back on it during my reporting of the things that really stayed with brexit**

Table A.5: Outputs for compression of every utterance in a snippet with the selected (unpruned) utterances bolded. A compression rate of 0.50%. The crowdsourced representative labels were 6 and 14 (0% recall). Note that the bolded utterances are informative in spite of zero recall.

Index	Original	Compression
0	she said in august in the lawsuit of that year ## she recounts the incident and alleges that she told trump i 've been on the road for you since march away from my family to which he replied you 're doing an awesome job	lawsuit that she recounts the incident and she said he replied my family in the lawsuit of doing an awesome job alleges

Continued on next page

Table A.5 – continued from previous page

Index	Original	Compression
1	go in there and kick some ass right so then the accusation is that he allegedly grabbed her by the hand and leaned in she described the moment to the washington post and as she describes the moment it 's actually kinda funny	it 's actually kinda funny is that she describes the accusation so then she grabbed her hand and as she described
2	the picture she describes you can see president trump then candidate trump going into slow motion you know just coming in for that kiss in slow motion you can just hear her saying ohh noo	slow motion you can see president trump going into the picture coming in for that she describes
3	you know i do n't believe any of this happened but whatever so she 's describing the moment saying oh my god i think he 's going to kiss me he 's coming straight for my lips	he 's coming straight for the moment i think my god i do n't believe any of saying
4	so she says i turn my head and he kisses me right on the corner of my mouth and still holding my hand the entire time then he walks out that 's it folks	that 's right on the corner of my head and she says i turn my mouth walks

Continued on next page

Table A.5 – continued from previous page

Index	Original	Compression
5	and for this she ca n't sleep she 's riddled with guilt i mean she 's bringing a lawsuit i just i do n't know to me it sounds rather silly	**she ca n't sleep she 's bringing a lawsuit to me it sounds silly**
6	but again let 's not forget what the left has been trying to do to president trump since the day he was elected now let 's not forget that we know that the mueller investigation we 're supposed to get that report at some point this week	let 's not forget that the left has been trying to get that report at some point this week since the mueller investigation
7	now the folks at the mueller investigation came out and said no that 's not gonna be the case so then what happens for the democrats the mueller report nothing comes out	democrats came out and said no that 's not gonna be the case for the mueller
8	and we also know that james clapper and some other folks were out there talking about how well the mueller report do n't hang your hopes on the mueller report folks because that thing could be anticlimactic phil talked about that of course	talked about that thing and we also know that james folks were out there on the mueller report do n't hang

Continued on next page

Table A.5 – continued from previous page

Index	Original	Compression
9	so now the democrats are in the situation where they do n't know what to do because the mueller report is gonna be anticlimactic you 've got all these other lawsuits that are swirling around but we know that the unemployment rate is looking really good	swirling around but we know that the democrats are in the situation where they do n't know what these other lawsuits is looking
10	there 's a lot of really good things going on in america right now all the democrats now have at their disposal is these kinds of accusations you know whether it 's the whole jussie smollett kind of a scenario or this lawsuit from this woman	**there 's a lot of really good things going on right now it 's the whole jussie smollett in america is these accusations**
11	it 's all about name calling that 's what it 's all about so is it to anybody 's surprise that as we found out that the mueller report was not gonna come out this week that here we are on monday talking about a lawsuit in the washington post	that 's all about so is it 's not gonna come out that 's all about this week in the washington post is it 's

Continued on next page

Table A.5 – continued from previous page

Index	Original	Compression
12	about some woman some former trump presidential campaign worker who was saying that the president attacked her battery is what she 's saying when even if it was true it 's just a little inaudible which i do n't think it 's true	it 's just a little inaudible which is what she 's saying when even if it was true it was saying
13	i 'm just saying i do n't think it 's true anyway trump then allegedly grabbed her by the hand leaned in blah blah according to the lawsuit johnson is a highly accomplished african american woman who served as a senior staffer for trump 's presidential campaign	according to the lawsuit i 'm just saying it 's true anyway i do n't think it 's true anyway i do n't
14	she was the suit claims an integral part of the campaign 's success and was repeatedly recognized for her contributions in the moment the defendant trump forcibly kissed her miss johnson was a highly successful and widely released respected campaign staffer	integral part of the suit repeatedly recognized she was a highly successful contributions in the defendant and widely released respected

Continued on next page

Table A.5 – continued from previous page

Index	Original	Compression
15	the lawsuits say she felt reduced to just another object according to the suit which refers to the incident as a humiliating violation which amounts to common law battery so that 's what they 're saying	according to the lawsuits felt reduced incident as a humiliating violation which refers to just say she felt
16	johnson 's attorney hasan a zavareei released a statement on monday noting that after trump 's victory miss johnson tried to move on with her life i wonder if she wanted to get a job within the trump administration	attorney hasan released a statement after trump administration she wanted to get a job with her life i miss
17	'cause this has all the earmarks of a disgruntled employee who left the campaign for whatever reason maybe she was fired we really do n't know why was she not given a job within the trump administration	fired for whatever reason maybe she was not given a job within the trump administration 'cause this 'cause
18	'cause typically you know a lot of these folks do end up getting a job with the administration now look i 'm just spitballin 'on this but the whole thing at least to me is highly suspect it really is	i 'm just spitballin now look at least a lot of these folks do end up getting a job 'cause

Continued on next page

Table A.5 – continued from previous page

Index	Original	Compression
19	but the attorney is comin 'out swingin ' he says but when she saw what her work on the campaign had rot a president who mocks the metoo movement and undermines the dignity of the office with his sexist and racist behavior	attorney is sexist and he says she saw a president who mocks the office with his work on swingin ' racist
20	she decided to seek justice for herself and the many other women victimized by this sexual predator so clearly we know that this is somebody who absolutely hates president trump calling him a sexual predator	calling him a sexual predator so she decided to seek herself and the many other women victimized
21	the attorney says that she is a brave woman and he says i am proud to represent her in this important lawsuit so you know we 'll have to wait and see what happens	woman and see what happens in this important lawsuit so you 'll have a brave to represent
22	i 've not seen this story go especially widespread as of yet but we 'll have to wait and see what happens my name is name redacted in for phil valentine who johnny just so you know he is on his way in	johnny just so i 've not seen this story but we 'll have to see what happens on his widespread name

Continued on next page

Table A.5 – continued from previous page

Index	Original	Compression
23	so incase you 're wonderin 'what in the h e double toothpicks what in the hell i 'm doin ' here well phil had a bit of a emergency i 'll let phil fill you in if you will if he wants to	phil fill you will double toothpicks in the hell i 'm so incase you 're in the doin 'what i 'll
24	myself i 'll be right back right here on super talk ## ## wtn	myself i 'll be right back ##
25	inaudible from the genesis diamonds wtn traffic center an accident from earlier ## westbound and hickory hollow pkwy looks like they 're still tryin 'to deal with that and it 's causin ' some minor delays	they 're still tryin diamonds wtn from earlier and hickory and it 's causin genesis ## westbound minor
26	## southbound near old hickory blvd some debris is in the roadway now the big debris that would be the mudslide of course from this weekend and that has things shut down	## this weekend and that would be the big roadway now shut down in the course
27	## eastbound old hickory blvd and briley pkwy and right out old hickory blvd that is where traffic is being diverted traffic every ## minutes for ya every morning on super talk	## eastbound old hickory blvd traffic is being diverted and briley and briley every ## minutes

Continued on next page

Table A.5 – continued from previous page

Index	Original	Compression
28	is your existing alarm system tryna tell you something is there a trouble beep or a special light lit up on that keypad of yours my name is name redacted president of nca please do not ignore your alarm system it 's tryna tell you something important	there 's tryna please do not ignore your existing alarm lit up on that tell you something is name is yours my name
29	it could be that you have a low backup battery or maybe that it 's not communicating right now with central station if that is the case it is time to get your system fixed nca was founded in ## right here in middle tennessee	it 's not the case or maybe it could be that you have a low battery or right here in middle tennessee
30	we 're family owned and operated and would love to help your family with your existing alarm system nca offers landline monitoring starting at just ## ## a month if you do n't have a landline nca offers wireless monitoring starting at just ## ## a month	we 're starting at just ## a month if you do n't have a family owned and operated monitoring starting to help nca

Continued on next page

Table A.5 – continued from previous page

Index	Original	Compression
31	we also can install a new system in your home or business and all monitoring agreements are just month to month give nca a call at ## ## 7nca that 's ## ## ## or visit our website at nca cc	nca monitoring agreements are just a new ## or ## 7nca and all that 's at ## we also visit
32	mention our license number ## and you can choose between ## months of free monitoring or a free cell dialer	mention our license number ## months of free cell dialer
33	message and data rates may apply	data rates may
34	when did it become ok for men to be lazier softer fatter we need to bring the men of this country back to greatness with ageless male max a patent pending formula with an ingredient that helps boost your total testosterone	it did it when did it become ok for men of this country that helps back to bring the fatter
35	promoting greater increases in muscle size and twice the reduction of body fat percentage than exercise alone plus an amazing ## increase in nitric oxide which can be handy in the gym and in the bedroom	promoting greater increases in the bedroom fat percentage than exercise alone plus an amazing ## reduction of body

Continued on next page

Table A.5 – continued from previous page

Index	Original	Compression
36	take your manhood to the max by trying your first ## day bottle free not ## days not ## days but a full ## day supply free when you text the word quick to ## just pay shipping and handling finally a	text supply free not ## days but a full ## day supply free not ## day supply free when you pay

Appendix B

Annotation Survey: Opinion Identification and Representative Utterance Selection

Figure B-1: Selection Pane 1: Captcha validation check.

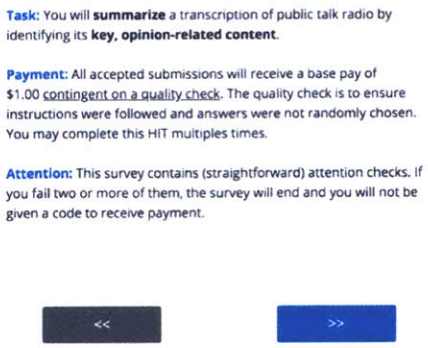
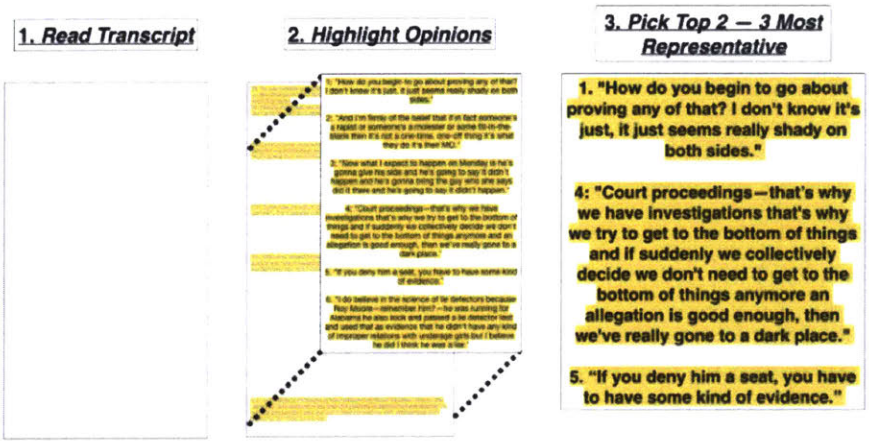


Figure B-2: Selection Pane 2: Basic information regarding the opinion identification and representative utterance selection tasks.

To summarize, you will use the following process:

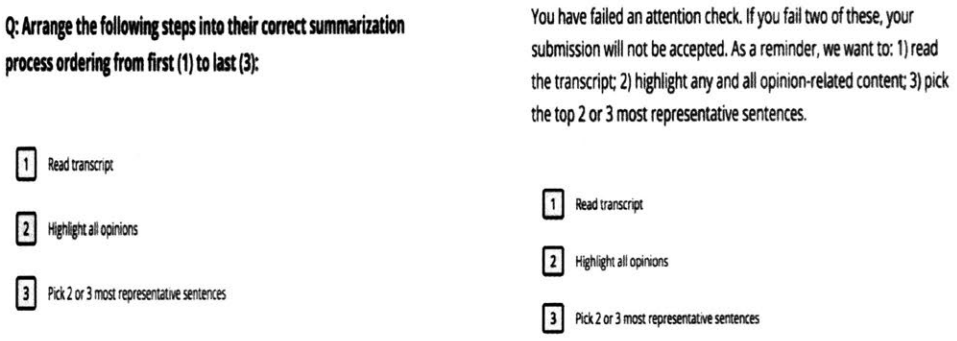


1. **Read** the transcript.

2. **Highlight** all sentences that contain opinions, interpretations, viewpoints and/or beliefs, regardless if you believe their content to be true or agree with them. This may sometimes include statements on behalf of other people, for example "a lot of people think filing taxes is annoying."

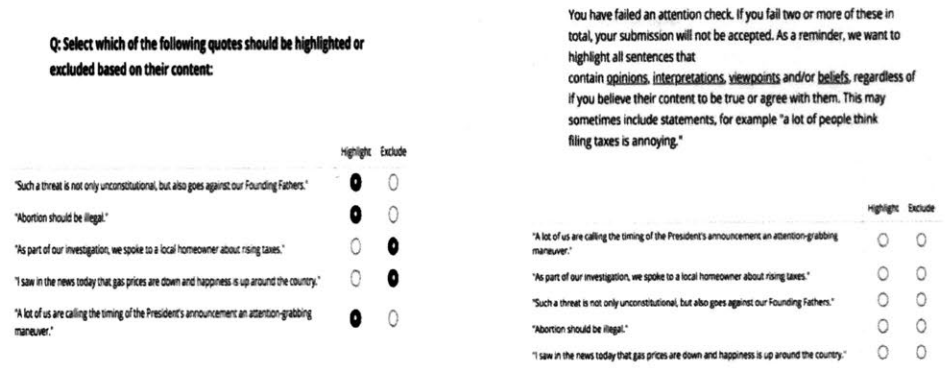
3. **Pick** the top 2 or 3 most conceptually important, representative sentences with respect to the discussion happening in the original transcript.

Figure B-3: Selection Pane 3: Instructions for identifying opinion and selecting representative utterances.



(a) Base Question (b) Incorrect Answer

Figure B-4: Selection Pane 4: Attention check 1 (easy) to verify their understanding of the overall annotation process.



(a) Base Question (b) Incorrect Answer

Figure B-5: Selection Pane 5: Attention check 2 (hard) to verify their understanding of opinion identification criteria.

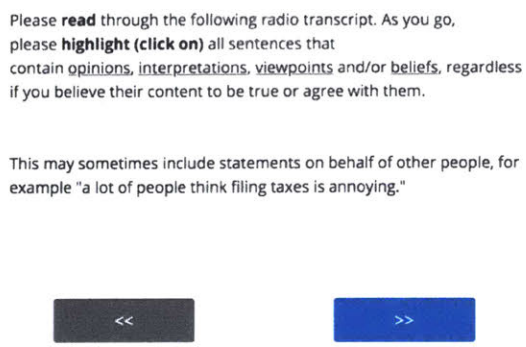


Figure B-6: Selection Pane 6: Reminder of instructions for identifying opinon.

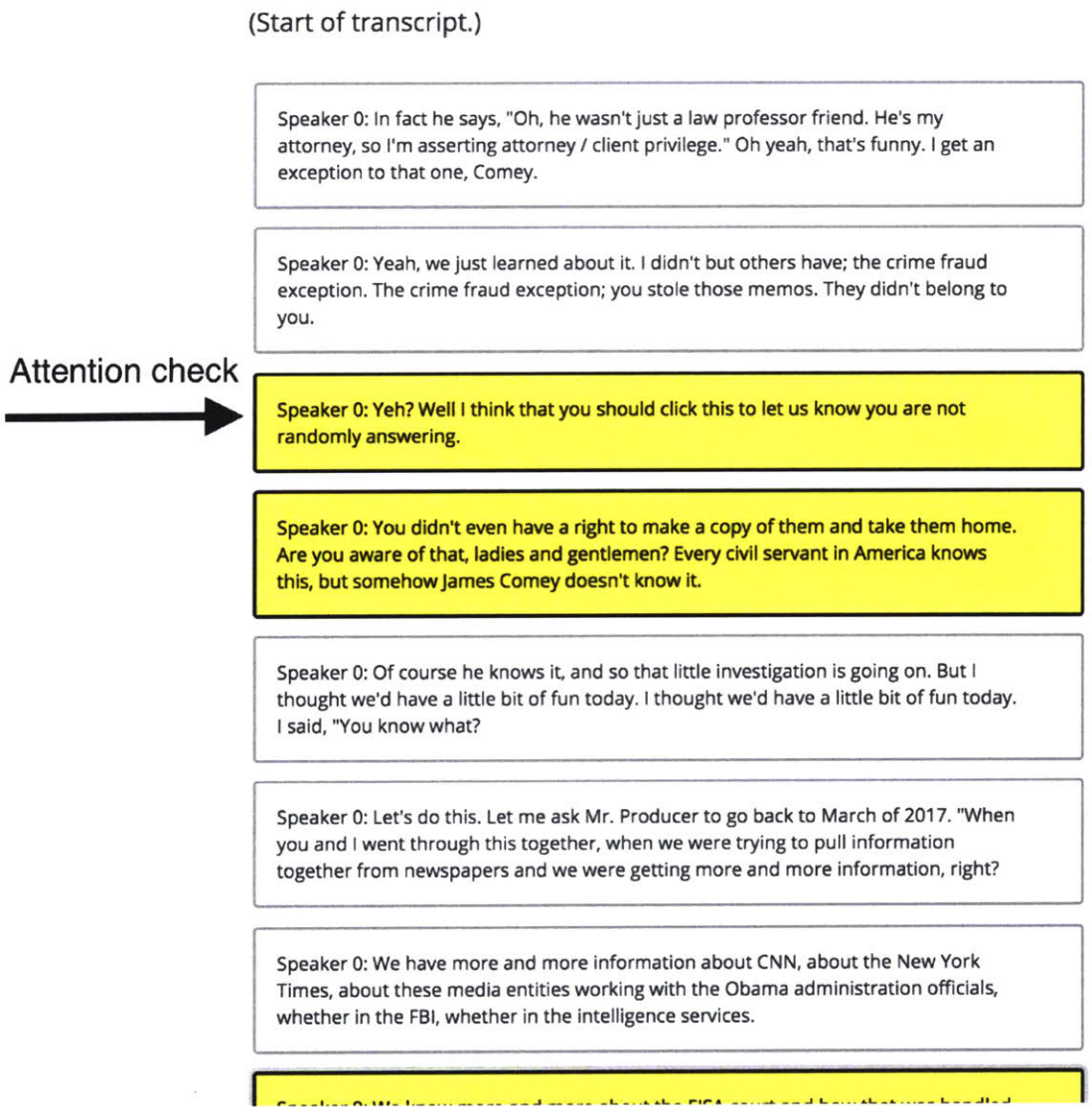


Figure B-7: Selection Pane 7: Opinion identification screen with a randomized attention check.

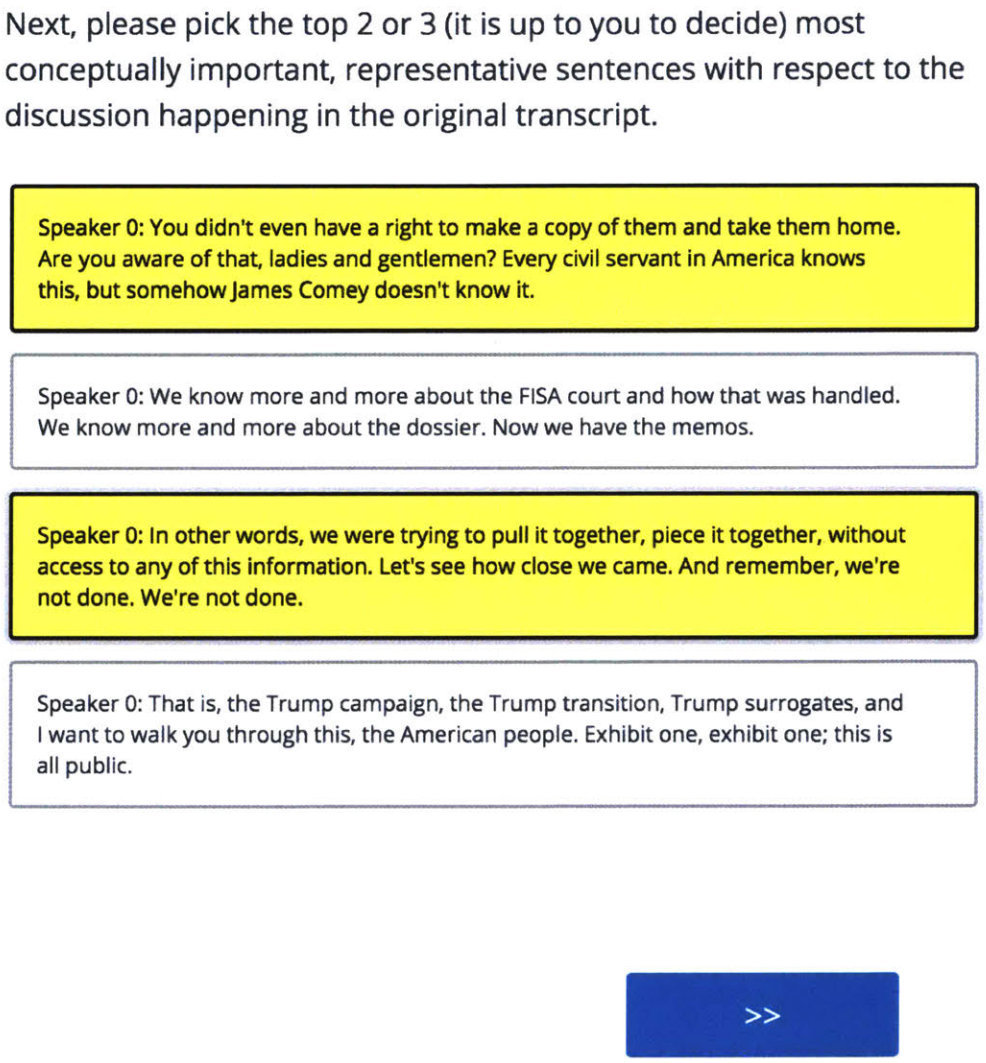


Figure B-8: Selection Pane 8: Representative utterance selection.

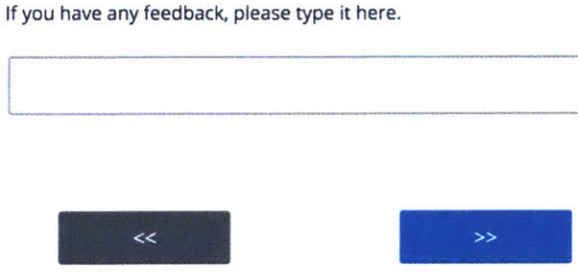
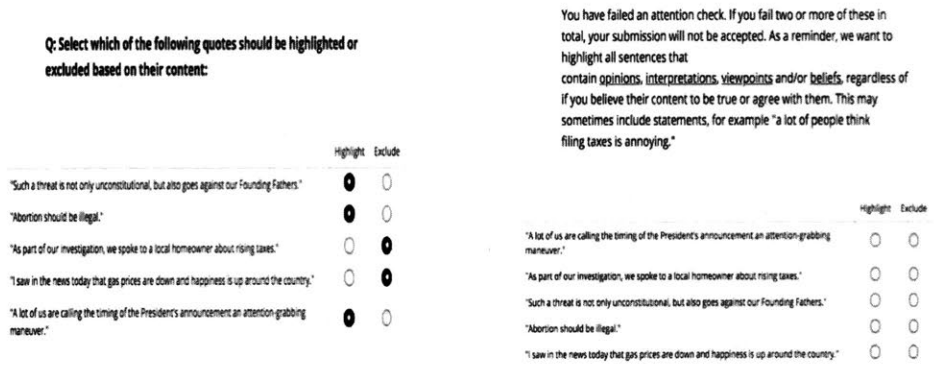


Figure B-9: Selection Pane 9: Optional feedback forum.



(a) Successful Completion Question

(b) Attention Failure

Figure B-10: Selection Pane 10: End of survey message.

Appendix C

Annotation Survey: Utterance Compression

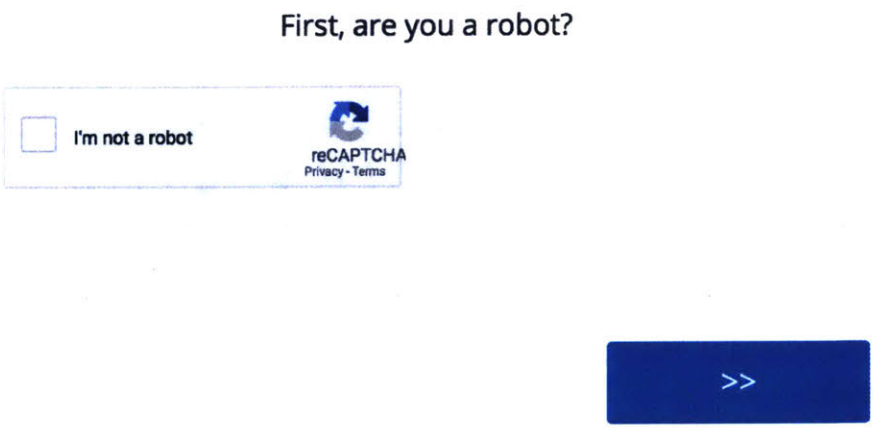


Figure C-1: Compression Pane 1: Captcha validation check that the annotator is human.

Task: You will shorten a paragraph by highlighting words that are *absolutely necessary* to retain the paragraph's **fluency** (A.K.A. grammaticality) and **original information**. Shortened outputs should be as concise as possible with respect to these criteria.

Punctuation are not considered to be important (such as in the example below), **multiple sentences may be combined**, and **entire sentences may be deleted** if you consider them to be extraneous.

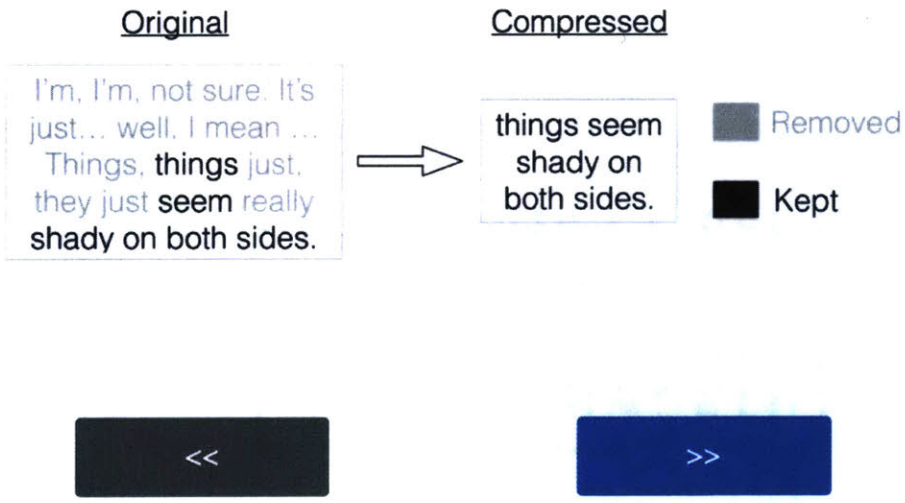


Figure C-2: Compression Pane 2: Information and instructions for performing utterance compression.

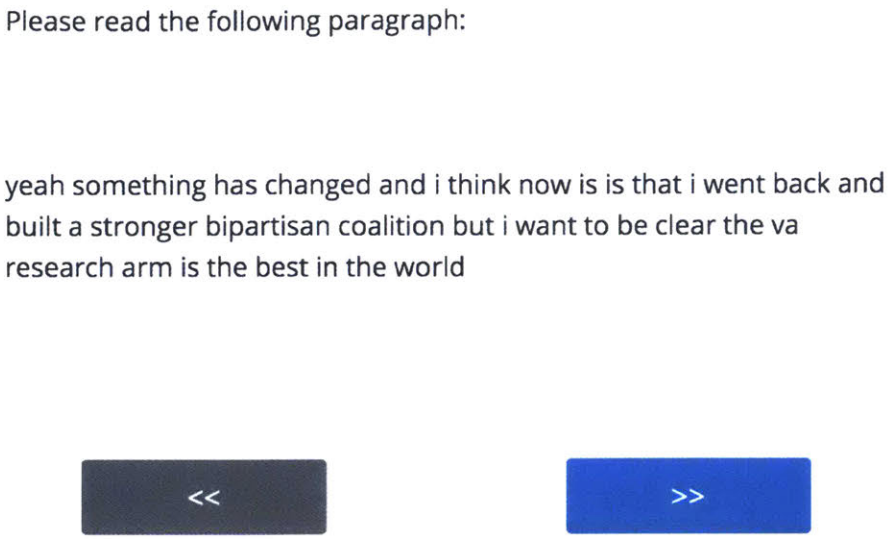


Figure C-3: Compression Pane 3: Utterance presentation.

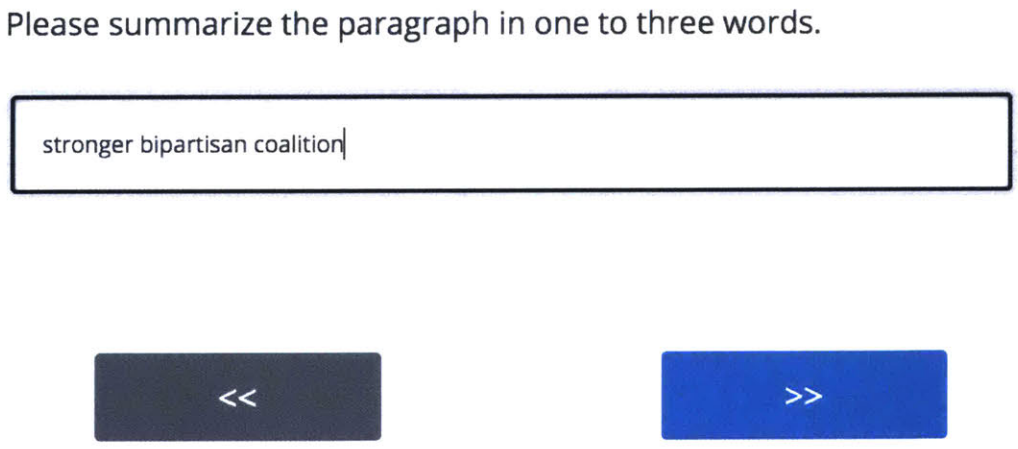
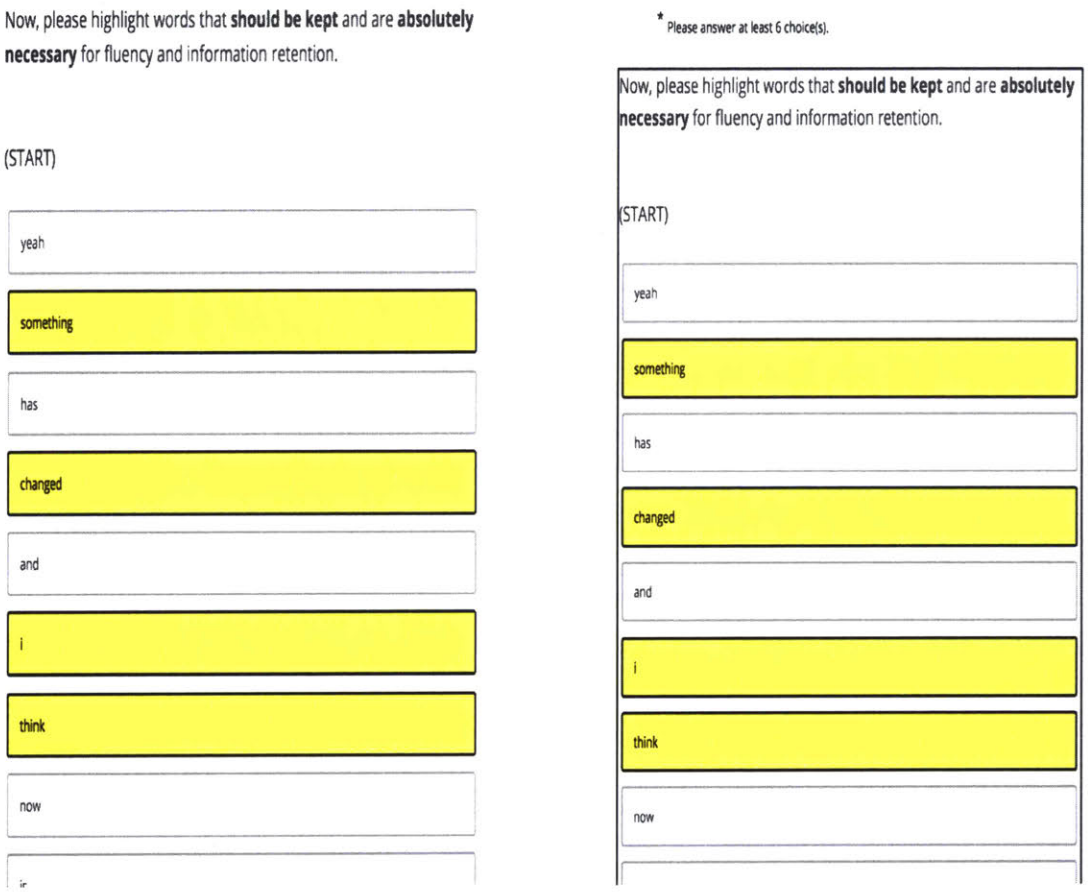


Figure C-4: Compression Pane 4: Attention check (easy) asking what the utterance was about in 1-3 words.



(a) Base Question

(b) Incorrect Answer

Figure C-5: Compression Pane 5: Utterance compression and error throw if less than six tokens are chosen.

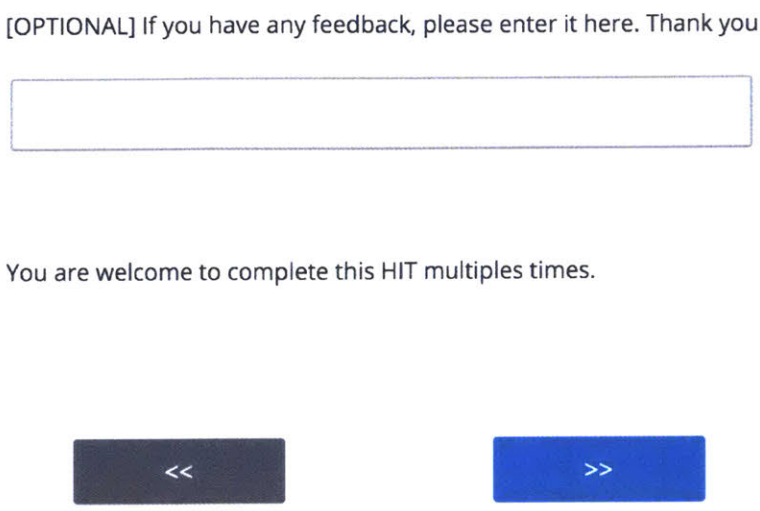


Figure C-6: Compression Pane 6: Optional feedback forum.

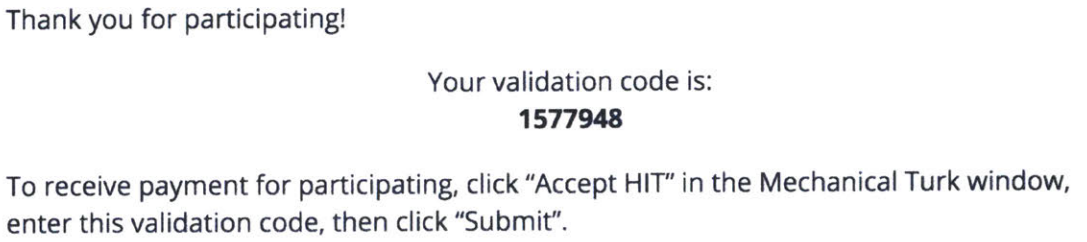


Figure C-7: Compression Pane 7: End of survey.

Bibliography

[1] Radio locator. https://radio-locator.com/, 2018.

[2] Rev transcription faq. https://www.rev.com/transcription/faq, 2019.

[3] Segmentation and diarization using lium tools. https://cmusphinx.github.io/wiki/speakerdiarization/, 2019.

[4] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

[5] Kamal Al-Sabahi, Zhang Zuping, and Mohammed Nadher. A hierarchical structured self-attentive model for extractive document summarization (hssas). *IEEE Access*, 6:24205–24212, 2018.

[6] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, and Krys Kochut. Text summarization techniques: A brief survey. In *arXiv*, arXiv, 2017.

[7] Stefanos Angelidis and Mirella Lapata. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *EMNLP*, 2018.

[8] Cameron B. Armstrong and Alan M. Rubin. Broken contract? changing relationships between americans and their government. *Journal of Communication*, 39(2):84–94, 1989.

[9] Yossi Asi, Einat Kermany, Yonaton Belinkov, Ofer Lavi, and Yoav Goldberg. Finegrained analysis of sentence embeddings using auxiliary prediction tasks. In *International Conference on Learning Representations*, ICLR, 2017.

[10] Joonhyun Bae and Sangwook Kim. Identifying and ranking influential spreaders in complex networks by neighborhood coreness. In *Physica A: Statistical Mechanics and its Applications*, volume 395, pages 549–559, 2014.

[11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations (ICLR)*, 2014.

[12] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization.* 43rd Annual Meeting of the Association for Computational Linguistics (ACL), 2005.

[13] David Barker and Kathleen Knight. Political talk radio and public opinion. *The Public Opinion Quarterly*, 64(2):149–170, 2000.

[14] Valentin Barrière, Chloé Clavel, and Slim Essid. Opinion dynamics modeling for movie review transcripts classification with hidden conditional random fields. In *INTERSPEECH*, 2017.

[15] Doug Beeferman, William Brannon, and Deb Roy. Radiotalk: a large-scale corpus of talk-radio transcript, 2019.

[16] Steven Bird, Edward Loper, and Ewan Klein. Natural language processing with python. In *O'Reilly Media Inc*, 2009.

[17] Jordan Body-Graber and David Blei. Syntactic topic models. In *Advances in neural information processing systems*, NIPS, pages 185–192, 2009.

[18] Florian Boudin and Emmanuel Morin. Keyphrase extraction for n-best reranking in multi-sentence compression. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT, pages 298–305. Association for Computational Linguistics, 2013.

[19] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasillis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. The ami meeting corpus. In *Second international conference on Machine Learning for Multimodal Interaction (MLMI)*, pages 28–39, 2005.

[20] Danqi Chen, Jason Bolton, and Christopher D. Manning. A thorough examination of the cnn/daily mail reading comprehension task. In *Association for Computational Linguistics (ACL).*

[21] Danqi Chen, Jason Bolton, and Christopher D. Manning. 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 2016. Association for Computational Linguistics.

[22] Chirp. Audio fingerprinting: what is it and why is it useful? https://blog.chirp.io/audio-fingerprinting-what-is-it-and-why-is-it-useful/, 2018.

[23] The Neilsen Company. News/talk audience share trend in 2017. https://www.nielsen.com/us/en/insights/news/2017/news-radio-reach-surges-during-september-hurricanes.html, 2017. Accessed 11-02-2018.

[24] The Neilsen Company. How america listens: The american audio landscape. https://www.nielsen.com/us/en/insights/news/2017/news-radio-reach-surges-during-september-hurricanes.html, 2018. Accessed 10-31-2018.

[25] Michele Confortio, Gerard Cornuejols, and Giacomo Zambelli. Integer programming. In *SpringLink Graduate Texts in Mathematics)*, 2014.

[26] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[27] Cortico. Earshot: What are people saying in the public sphere? https://earshot.cortico.ai/about, 2018–2019.

[28] Bin Dai and David Wipf. Diagnosing and enhancing vae models. In *International Conference on Machine Learning (ICLR)*, 2019.

[29] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[30] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[31] Gravem Edouard, Armand Joulin, and Nicolas Usunier. Improving neural language models with a continuous cache. In *International Conference on Learning Representations*, ICLR, 2017.

[32] Günes Erkan and Dragomir R. Radev. Lexrank: Graph-based centrality as salience in text summarization. 2004.

[33] Thibault Fevry and Jason Phang. Unsupervised sentence compression using denoising auto-encoders. In *Proceedings of the SIGNLL Conference on Computing Natural Language Learning.* Association for Computational Linguistics, 2018.

[34] Katja Filippova. Multi-sentence compression: Finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 322–330, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[35] Joseph L. Fleiss. Measuring nominal scale agreement among many raters. In *Psychological Bulletin*, pages 378–382, 1971.

[36] Lee Francis. Talk radio listening, opinion expression and political discussion in a democratizing society. *Asian Journal of Communication*, 17(1):78–96, 2007.

[37] Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics*, Coling, pages 340–348, 2010.

[38] Nikhil Garg, Benoit Favre, and Korbinian Reidhammer. Clusterrank: a graph based method for meeting summarization. In *Tenth Annual Conference of the International Speech Communication Association*, 2009.

[39] Demian Golipour Ghalandari. Revisiting the centroid-based method: A strong baseline for multi-document summarization. In *Conference on Empirical Methods in Natural Language Processing Workshop on New Frontiers in Summarization*, 2017.

[40] David Graff and Christopher Cieri. English gigaword ldc2003t05. https://catalog.ldc.upenn.edu/LDC2003T05, 2003. Accessed 05-01-2019.

[41] Intelligence Advanced Research Projects Activity Group. Automatic speech recognition in reverberant environments (aspire) challenge. https://www.iarpa.gov/index.php/working-with-iarpa/prize-challenges/306-automatic-speech-in-reverberant-environments-aspire-challenge, 2015.

[42] Martin Hassel. Evaluation of automatic text summarization: A practical implementation. In *KTH Numerical Analysis and Computer Science*, Licentiate Thesis, 2004.

[43] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[44] Richard C. Hofstetter, David Barker, James T. Smith, Gina M. Zari, and Thomas A. Ingrassia. Information, misinformation, and political talk radio. *Political Research Quarterly*, 2(52):353–369, 1999.

[45] Richard C. Hofstetter, Mark C. Donovan, Melville R. Klauber, Alexandra Cole, Carolyn J. Huie, and Toshiyuki Yuasa. Political talk radio: A stereotype reconsidered. In *Political Research Quarterly*, volume 2, pages 467–479, 1994.

[46] Ian Hutchby. Confrontation talk: Arguments, asymmetries, and power on talk radio. *Routledge*, 2013.

[47] Amazon.com Inc. Amazon mechanical turk. https://www.mturk.com/, 2019.

[48] Google Inc.

[49] Maluuba Inc. Evaluation code for various unsupervised automated metrics for natural language generation. In *GitHub*, 2018.

[50] A Janin, D Baron, J Edwards, D Ellis, D Gelbart, Nathaniel Morgan, B Peskin, Thilo Pfau, Elizabeth Shriberg, A Stolcke, and Chuck Wooters. The icsi meeting corpus. pages I–364, 05 2003.

[51] Daniel Jurafsky and James H. Martin. Chapter 4: Ngrams. In *Speech and Language Processing*, pages 1–28. Stanford University, 2014.

[52] Urvashi Khandelwal, He He, Peng Qi, and Daniel Jurafsky. Sharp nearby, fuzzy far away: How neural language models use context. ACL, 2018.

[53] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR)*, 2015.

[54] Xin Li, Lidong Bing, Piji Li, and Wai Lam. A unified model for opinion target extraction and target sentiment prediction. *arXiv preprint arXiv:1811.05082*, 2018.

[55] Chin-Yew Lin. Rouge: a package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out.* 42nd Annual Meeting of the Association for Computational Linguistics (ACL), 2004.

[56] Chin-Yew Lin and Eduard Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the Human Language Technology Conference.* North American Chapter of the Association for Computational Linguistics (HLT-NAACL), 2003.

[57] Hui Lin. Submodularity in natural language processing: algorithms and applications. In *Master's thesis*, University of Washington, 2012.

[58] Hui Lin and Jeff Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 912–920, 2010.

[59] Rada Mihalcea and Paul Tarau. Long story short - global unsupervised models for keyphrase based meeting summarization. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004.

[60] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *27th Conference on Neural Information Processing Systems (NIPS)*, pages 3111–3119, 2013.

[61] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *AAAI*, 2016.

[62] Takeshi Onishi, Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. Who did what: A large-scale person-centered cloze dataset. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2230–2235, Austin, Texas, November 2016. Association for Computational Linguistics.

[63] Diana Owen. Who's talking? who's listening? the new politics of radio talk shows. In *Broken Contract? Changing Relationships Between Americans And Their Government*, 2017.

[64] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web, 1999.

[65] Hengyue Pan and Jiang Hui. Annealed gradient descent for deep learning. In *Association for Uncertainty in Artificial Intelligence (AUAI)*, 2015.

[66] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[67] Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. English gigaword fifth edition ldc2011t07. https://catalog.ldc.upenn.edu/LDC2011T07, 2011. Accessed 05-01-2019.

[68] Vijayaditya Peddiniti, Guoguo Chen, Vimal Manohar, Tom Ko, Daniel Povey, and Sanjeev Khudanpur. Jhu aspire system : Robust lvcsr with tdnns, ivector adaptation and rnn-lms. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 539–546. IEEE Signal Processing Society, 2015.

[69] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[70] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The kaldi speech recognition toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December 2011. IEEE Catalog No.: CFP11SRW-USB.

[71] Ofir Press and Lior Wolf. Using the output embedding to improve language models. In *EACL*, 2017.

[72] National Public Radio. Talk of the nation. https://www.npr.org/programs/talk-of-the-nation/, 2008–2013.

[73] National Public Radio. Morning edition. https://www.npr.org/programs/morning-edition/, 2013–2019.

[74] Anand Rajaraman and Jeffrey Ullman. Data mining. In *Mining of massive datasets*, pages 1–17, 2011.

[75] Korbinian Riedhammer, Benoit Favre, and Dilek Hakkani-Tur. Long story short - global unsupervised models for keyphrase based meeting summarization. In *Speech Commun*, volume 52(10), pages 801–815, 2010.

[76] Sebastian Ruder. Nlp progress. https://github.com/sebastianruder/NLP-progress/blob/master/english/summarization.md, 2018.

[77] Chopra Sumit Rush, Alexander M. and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT, pages 298–305. Association for Computational Linguistics, 2013.

[78] Shaeke Salman and Xiuwen Liu. Overfitting mechanism and avoidance in deep neural networksg. In *arXiv: 1901.06566*, pages 1–8, 2019.

[79] Paddy Schannell. Broadcast talk. In *Broadcast talk*, volume 5. Sage, 1991.

[80] Guokan Shang, Wnesi Ding, Zekun Zhang, Antoine J.-P. Tixier, Polykarpos Meladianos, Michalis Varzirgiannis, and Jean-Pierre Lorre. Unsupervised abstractive meeting summarization with multi-sentence compression and budgeted submodular maximization. In *Proceedings of Association for Computational Linguistics*, 2018.

[81] Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *CoRR*, abs/1706.09799, 2017.

[82] The Rush Limbaugh Show. Premiere radio networks. https://www.rushlimbaugh.com/, 2014–2018.

[83] J. Sim and C. C. Wright. The kappa statistic in reliability studies: Use, interpretation, and sample size requirements. In *Physical Therapy*, pages 257–268, 2005.

[84] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[85] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *28th Conference on Neural Information Processing Systems (NIPS)*, pages 3104–3112, 2014.

[86] Briony Swire, Adam Berinsky, Stephan Lewandowsky, and Ullrich K. H. Eck. Processing political misinformation: comprehending the trump phenomenon. *Royal Society of Open Science*, 2017.

[87] Paul Tardy. A full python implementation of the rouge metric (not a wrapper). In *GitHub*, 2017.

[88] Antoine Tixier, Fragkiskos Malliaros, and Michalis Vazirgiannis. A graph degeneracy-based approach to keyword extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 1860–1870. Association for Computational Linguistics, 2016.

[89] Antoine Tixier, Polykarpos Meladianos, and Michalis Vazirgiannis. Combining graph degeneracy and submodularity for unsupervised extractive summarization. In *Conference on Empirical Methods in Natural Language Processing Workshop on New Frontiers in Summarization*, pages 48–58, 2017.

[90] Jenine Turner and Eugene Charniak. Supervised and unsupervised learning for sentence compression. In *Proceedings of the Association for Computational Linguistics*, pages 290–297, 2005.

[91] Joseph Turow, Joseph Cappella, and Kathleen Jamieson. Call-in political talk radio: Background, content, audiences, portrayal in mainstream media. *Report Series*, (5), 1996.

[92] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2692–2700. Curran Associates, Inc., 2015.

[93] Yaushian Wang and Hung-yi Lee. Learning to encode text as human-readable summaries using generative adversarial networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4187–4195, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[94] Sam Wiseman and Alexander M. Rush. Sequence-to-sequence learning as beam-search optimization. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1296–1306, Austin, Texas, November 2016. Association for Computational Linguistics.

[95] Itzhak Yanovitzky and Joseph N. Cappella. Effect of call-in political talk radio shows on their audiences: Evidence from multi-wave panel analysis. *International Journal of Public Opinion Reearch*, 13(4):377–399, 2001.

[96] David Zajic, Bonnie J. Door, Jimmy Lin, and Richard Schwartz. Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. In *Information Processing and Management*, pages 1549–1570. ScienceDirect, 2007.