Liquid News - A Semantic-Relational Model for Enhanced Understanding

by

Dagmawi Samuel Haile

S.B. Computer Science and Engineering
Massachusetts Institute of Technology (2022)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2023

Authored by: Dagmawi Samuel Haile
Department of Electrical Engineering and Computer Science
May 12, 2023

Certified by: Andrew B. Lippman, Ph.D.
Senior Research Scientist and Associate Director of the MIT Media Lab
Thesis Supervisor

Accepted by: Katrina LaCurts
Chair, Master of Engineering Thesis Committee

Liquid News - A Semantic-Relational Model for Enhanced Understanding

by

Dagmawi Samuel Haile

Submitted to the Department of Electrical Engineering and Computer Science
on May 12, 2023, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

The landscape in which society interacts with news has evolved due to the advent of the internet and modern communication platforms. Although this evolution has led to greater diversity and accessibility of news media, it has also created challenges regarding selective news coverage, bias, and fake news. This work proposes a novel news platform called Liquid News that aims to enhance people's understanding of news by leveraging machine-learning-based analysis and semantic navigational aids. Semantic segmentation and unsupervised clustering are the core machine-learning tasks underpinning Liquid News. Thus far, many state-of-the-art (SoTA) large language models provide building blocks for both tasks. However, more research needs to be done on combining large language models and their application to analyzing video news. Liquid News addresses this domain gap by intersecting semantic segmentation, unsupervised clustering, and video processing in application to video news. Furthermore, Liquid News investigates solutions to overcoming the challenges of anisotropy in semantic embedding and clustering of text.

Thesis Supervisor: Andrew B. Lippman, Ph.D.
Title: Senior Research Scientist and Associate Director of the MIT Media Lab

Acknowledgments

I thank Andrew Lippman for his guidance and support throughout this thesis. His over thirty-five years of academic and research experience, particularly in the context of technology and news, provided invaluable insight into this work.

I thank my Viral Communications labmates: Mike Jiang, Kevin Dunnel, Erick Oduniyi, and Trudy Painter, for their feedback and help in developing this work. I also want to acknowledge Alessandra Davy-Falconi for all her efforts in assisting the Viral Communications group in conducting our research.

I thank all my friends who provided motivation and feedback while completing my thesis and reminding me to take a break from my research and live a little. In particular, I want to thank Omoruyi Atekha and Nebyu Haile for their insights into the Liquid News UI/UX. Additionally, I would like to thank Azariah Beyene and Mikael Nida for their insight into the system architecture of Liquid News. Lastly, I want to acknowledge Christian Belser and Julian Manyika for collaborating on the RPCSE model.

I thank the Rogers Foundation for supporting my undergraduate and graduate studies, allowing me to focus on my research and intellectual growth.

Lastly, I want to thank my loved ones for their continued support throughout my time at MIT. In particular, my parents, Samuel and Emebet, and my siblings, Nebyu and Tsedenya, for their love and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

For much of the 20th century, news media was constrained to a handful of media outlets and simple publication mediums (i.e., print, radio, and television)[7]. The Federal Communication Commission (FCC) used a set of strict regulations under policies such as the Communications Act of 1934 [15] and Fairness Doctrine [41] to ensure media outlets broadcasted content that was in the public's best interest, presented a well-rounded view of crucial issues and offered unbiased news reporting. However, in the 1980s, a remarkable transformation occurred in the United States media landscape. Politics and policy changes spurred the rise of cable television, which ushered in a new era of political communication known as the post-broadcast democracy era. The enactment of notable policies such as the Cable Act of 1984 [14] and repealing the Fairness Doctrine in 1987 [41] set the stage for the evolution of the news media landscape. The relaxation of media regulation synchronized with the rise of internet technology, leading to the proliferation and expansion of media outlets, each with its perspective on important issues.

Over the subsequent three decades, media outlets continued to evolve at a rapid pace embracing modern platforms such as websites (Google), news aggregators (Apple News), social media (Facebook), and messaging apps (WhatsApp) into their distribution. Today, over 50% of people surveyed in the United States received some portion of their news from online platforms [7, 43]. However, although there has been a significant shift towards online platforms, traditional news still plays a vital role in news

consumption, particularly TV news (cable and broadcast) [43].

The news landscape's current state offers significant improvements compared to the 1980s regarding accessibility and diversity of thought. However, the sheer amount of news content has made parsing the news significantly more difficult. The post-broadcast democracy era has seen broadcast media digital distribution increase channel capacity by over six-fold. Recent statistics show that over 30,000 hours of content are uploaded per hour to the internet. Reports suggest that the growth of video distributed over the internet may be exponential [34]. The growth of news content in conjunction with the relaxation of regulations in the 1980s has pushed the landscape away from one that obeys Hotelling's law - it is rational for producers to make their products as similar as possible in many markets. Many news outlets now selectively filter information and unequivocally distort facts, often to push political agendas [7].

This distortion of the news is made more dangerous by the decrease in people's ability to identify attribution. Studies show that less than 50% of users can identify the source of their information when found from online sources [34]. Another study also found that only 9% of U.S. adults are confident that they can tell if a news organization does its reporting [7]. The fragmentation of the media landscape has also led to growing concerns over the diversity of perspectives people are exposed to. More than half the U.S. population prefer algorithmic media suggestion from social media such as Facebook, YouTube, and TikTok compared to editors. These algorithms introduce the possibility of creating media echo chambers by exposing users to only content and perspectives they agree with.

These factors have culminated in a news landscape in which viewers distrust and struggle to comprehend modern issues. The degradation of trust between viewers and media outlets is shown by recent studies, which found only 26% of people in the US trust the news [43]. Studies have also found increased disconnection and disengagement with news in the US. One possible reason is the increased difficulty in comprehending the news, with 15% of younger US viewers citing trouble understanding the news [43].

News is, therefore, now at a convergence of niche market profitability, blurred lines

between factual news and editorial, and the ability to design the presentation for a reaction rather than reflection. This new landscape motivates Liquid News, which aims to improve interaction and comprehension of the news by leveraging machine-learning-based analysis and semantic navigational aids.

1.1 Proposed Work

This thesis proposes Liquid News and RPCSE - Relative Placement Contrastive Learning of Sentence Embeddings. These two models provide a system to help people better parse and understand modern news while exploring more perspectives. More specifically, RPCSE expands on the existing SimCSE model to reduce the anisotropy behavior seen in the word embedding space of large language models. This work directly contributes to clustering and semantic segmentation tasks, which Liquid News addresses. Building on the work of RPCSE, Liquid News implements a new unsupervised model for semantic segmentation and clustering whose output is used to generate a clip-based semantic relational platform for news.

Chapter 2

Background & Related Work

2.1 Background

2.1.1 Word Embeddings

Semantic segmentation and unsupervised clustering are the key tasks underpinning the work done in this thesis. These tasks are related to embedding natural text into a vector space from which notions of relative relation and semantic similarity can be extracted. The embedding of natural language into a vector space is the basis for more complex natural language processing (NLP) tasks such as segmentation, clustering, part-of-speech tagging, etc. There has been extensive work in the language embedding domain, but two major paradigms have emerged Latent Semantic Analysis (LSA) and Neural Net Language Models (NNLMs) such as CBOW [32], BERT [17], RoBERT [26], and GPT-3[8].

Latent Semantic Analysis

Latent Semantic Analysis (LSA) is a computational approach that uses statistical computations on a large text corpus to extract and represent the contextual-usage meaning of words. The critical insight behind LSA is that the collective data from the contexts in which a word appears and where it does not appear provides constraints that reliably determine the similarity in meaning between words and sets of words

[25]. The first step in LSA is to create a Term-Document-Matrix (TDM), which represents text as a matrix where rows represent distinct words and columns refer to the document of origin. The value of a cell in the matrix is the frequency of word R in doc C. Cell values then undergo a Term Frequency - Inverse Document Frequency (TF-IDF) and normalization transformation to weight words as a function of their importance to a given document. Finally, singular value decomposition (SVD) is applied to the matrix to derive a latent semantic structure model. The SVD of the TDM derives a set of uncorrelated factors, k, by which each term and document can be represented. Therefore a vector in the k-space can be used to define natural language. Furthermore, SVD is computationally effective, as in most cases, $k < N$ [16]. It is important to note that LSA is limited to its training corpus, and in practice, LSA is used on a smaller, more fine-tuned corpus, and it gains its contextual information strictly from this corpus.

Neural Net Language Models

Neural Net Language Models (NNLM) are an approach to word embeddings that takes a neural network approach to represent words as dense vectors in high-dimensional space. These models vary from LSA because they gain semantic and syntactic relationships between words by training on a large dataset of text corpora [20]. One of the first neural models for learning word embeddings was the Continuous Bag of Words (CBOW) which predicts the probability of a word given a continuous distribution of context [32]. This model represents each word as a high dimensional vector and then trains a feed-forward NNLM on a large corpus to learn the representation of words [32]. Following the NNLM approach to word embedding, the seminal transformer architecture introduced the self-attention mechanisms, which significantly improved the ability of NNLMs to learn the relationships between words in a given context. Self-attention enabled NNLMs to better weigh the importance of words in a given context when learning embeddings, which was crucial in long-range context dependencies. Furthermore, multi-head attention enabled different types of relationships to be understood between words [46]. Learning word embeddings via the transformer

architecture leads to state-of-the-art (SoTA) performance on downstream NLP tasks such as machine translation, sentiment analysis, and question answering. Building on this work, Dandekar et al. improved learned word embeddings using a contrastive objective function [33].

2.1.2 Anisotropy

Although word embeddings derived from Neural Net Language Models (NNLMs) have led to SoTA performance on downstream NLP tasks, anisotropy has been identified as a limitation in models such as GPT-2[39], BERT[17], RoBERTA[26], where it has been discovered that their embeddings space shows a degenerated structure that causes reduces semantic expressiveness. Anisotropy is a phenomenon in which embeddings exist within a narrow cone of vector space. As a result, embeddings for disparate, unrelated, or contradicting entities can have similar or identical cosine similarities [40, 18]. One proposed solution to the anisotropy issues is the whitening transformation, which transforms the embeddings into a standard normal distribution with a mean vector of zero and a covariance matrix as the identity matrix [28]. This transformation has been shown to reduce anisotropy and improve the performance of language models on various NLP tasks. Recently, work has shown that a contrastive learning approach can regularize pre-trained word embedding, making the embedding space more uniform [33]. One such model is SimCSE [19], which is explored in 2.2.

2.1.3 Clustering

The k-means algorithm is a well know unsupervised method for solving the following problem. Given a dataset $V = [v_1; : : : ; v_n]$ of d-dimensional datapoints and a set of clusters $C = [c_1; : : : ; c_t]$. We want to produce a matrix a V x C binary matrix A, such that A_{ij} indicates if v_i belongs to cluster c_j. The problem is optimized via the objective function in Eq. 2.1 [44, 22]

$$O(A; C) = \sum_{i=1}^{n} \sum_{k=1}^{t} A_{ik} \, k v_i \quad c_k k^2 \tag{2.1}$$

Although k-means is considered unsupervised, the value of k is known a priori. This poses a challenge as k is unknown beforehand in most applications. To resolve the issue of not knowing k a priori, many cluster validity methods have been presented to validate clustering results with known knowledge or intrinsic data information [44]. More recently, the X-means and U-k-means methods enable clustering utilizing a range of cluster numbers or no cluster numbers, respectively [44, 36].

2.1.4 Large Language Models

Recently, a new class of neural net language models, known as large language models (LLMs), has emerged. These models refer to neural net models that have been scaled to a large set of parameters and training datasets, often on the scale of billions. LLMs such as GPT-4 [35], GPT-3 [8], Claude [21], and LaMDA [37] have shown impressive results in understanding and generating natural language text, particularly in complex scenarios in which earlier models such as BERT [17], RoBERTa [26], and GPT-2 [39] failed. These models achieve state-of-the-art (SOTA) results in various distinct NLP tasks, which showcases their increasingly task-agnostic abilities and a shift from fine-tuning pre-trained models. LLMs have also been shown to contain emergent properties that they were not designed or trained to do beforehand.

Emergent properties are heavily tied to reinforcement learning from human feedback (RLHF) [35] or Constitutional AI [21]. RLHF focuses on aligning AI with humans through human feedback. In contrast, Constitutional AI focuses on aligning AI through principles that minimize harmful action while maximizing performance. In both cases, reinforcement learning has demonstrated that LLMs can be even more performant via few-shot or one-shot learning models [8], in which the pre-trained model is given a few or one demonstration of tasks beforehand. Furthermore, combining reinforcement learning with human feedback or constitutional principles with few-shot learning has enabled LLMs to invoke task-specific behavior with natural language instructions.

This paradigm has been a significant motivator in the growth of work related to language models in dialogue applications such as PaLM [27], ChatGPT, and Claude

[21]. However, although LLMs have shown to be robust in many emergent and NLP-specific tasks, they are still limited by their tendency to produce untrue or nonsensical content, often termed hallucinations [35].

2.2 Related Work

2.2.1 SECTOR

One core task of this thesis is to apply topic classification and segmentation to the news. There has been extensive work in text segmentation, topic modeling, and text classification within this domain, but there is less work in models that combine all three. One project that addresses topic classification and segmentation is SECTOR, a neural model that segments a large corpus of text into segments with assigned topic labels [6]. The SECTOR paper defines the WikiSection machine reading tasks. Given a document $D = \langle S; T \rangle$ consisting of N consecutive sentences $S = [S_1; \ldots; S_N]$, split D into a collection of distinct sections $T = [T_1; \ldots; T_M]$, where each section $T_i = \langle S_i; y_i \rangle$ contains a sequence of sentence $S_i \quad S$ and topic label y_i that describes the sentences [6]. The model uses a supervised approach in which processed Wikipedia articles are used as ground truth for segments T and topic labels y. SECTOR solves the problem by breaking the problem down to a sentence-level topic assignment because $|T|$ is unknown beforehand.

$$p(\bar{y}_1; \ldots; \bar{y}_N \mid \mathbf{D}) = \prod_{k=1}^{N} p(\bar{y}_k \mid \mathbf{s}_1; \ldots; \mathbf{s}_N) \tag{2.2}$$

Given this new problem definition, the SECTOR model implements a four-stage pipeline. The first stage embeds each sentence using the word2vec neural model [32]. The second stage uses sentence embeddings to create a distributional representation of latent topics using an LSTM architecture [23]. The third stage assigns each sentence to its most probable label y_k with a simple feed-forward neural net with a softmax activation trained to maximize matching to headings in the dataset, yielding a dense topic embedding matrix $E = [e_1; \ldots; e_N]$. The final stage of the model uses princi-

ple component analysis (PCA) and Gaussian smoothing to implement segmentation boundaries based on the embedding deviation per sentence [6].

2.2.2 SimCSE

As stated in the 2.1, anisotropy has been challenging in word embeddings produced by neural net language models (NNLMs). Recent work has shown that a contrastive learning approach can mitigate the issue [33, 19]. One model in particular, Simple Contrastive Learning of Sentence Embeddings (SimCSE), uses a contrastive learning framework to finetune word embeddings from large language models (LLMs) [19]. Supervised SimCSE utilizes the entailment and contradiction pairs from the Natural Language Inference dataset (NLI) [30] as positive and hard negatives for its contrastive learning task. An entailment or positive datum is a pair of sentences $(x_i; x_i^+)$ where x_i is an anchor sentence and x_i^+ is an entailed or semantically similar sentence. A contradiction or hard-negative datum is a pair of sentences $(x_i; x_i)$ where x_i is a semantically different sentence. Using this dataset, SimCSE extends the single datum to be $(x_i; x_i^+; x_i)$, combining the entailment and contradiction pairs for a given anchor, shown in Figure 2-1.

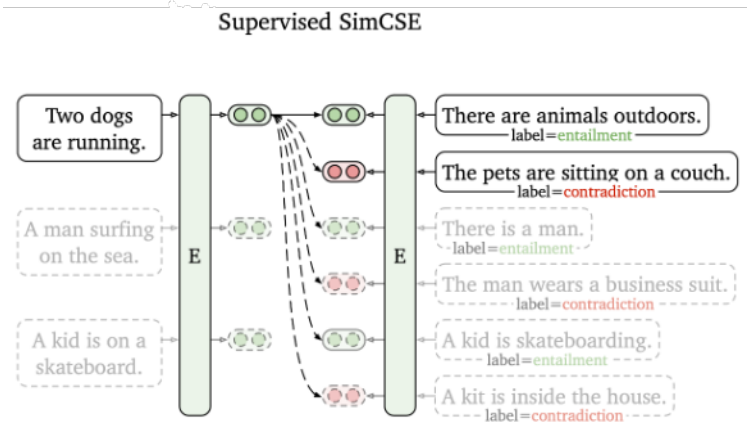

Figure 2-1: Shows the structure of the supervised SimCSE learning model [11]

Given the extended datum $(x_i; x_i^+; x_i)$, SimCSE trains on the SNLI dataset using loss defined in Eq. 2.3, taking the cross entropy objective with n-batch negatives in a mini-batch with N pairs, where is temperature, sim is cosine similarity, and

$(h_i; h_i^+; h_i)$ is the embedded datum $(x_i; x_i^+; x_i)$ from a pre-trained LLM such as BERT [17] or RoBERTa [26].

$$L_{CE} = -\log \frac{e^{\text{sim}(h_i;h_i^+)}}{\sum_{j=1}^{N} e^{\text{sim}(h_i;h_j^+)} + e^{\text{sim}(h_i;h_j)}} \tag{2.3}$$

2.2.3 Loss Functions

Triplet Margin Loss

Sentence-BERT (SBERT) [42] is a modified variant of the BERT architecture introduced by [17] that leverages a siamese and triplet network structure to improve the semantic meaningfulness of sentence embeddings of the original BERT architecture, setting SoTA performance on classification and regression tasks.

The SBERT model proposed three objective functions: classification, regression, and triplet. In the triplet objective function defined in Eq. 2.4, three datums are described: the anchor sentence s_a, a positive (semantically similar) sentence s_p, and a negative (semantically different) sentence s_n. Eq. 2.4 tunes the SBERT network such that the semantically similar sentences s_a and s_p are pushed together and the semantically different sentences s_a, s_n are pushed apart.

$$\max(\|s_a - s_p\| - \|s_a - s_n\| + \epsilon, 0) \tag{2.4}$$

DiffCSE

DiffCSE [13] is an unsupervised contrastive learning framework for learning sentence embeddings that builds on the work done in the SimCSE model [19]. DiffCSE builds on top of SimCSE by adding a difference prediction objective function to the standard contrastive learning object defined in SimCSE. This combination of these losses results in the objective function defined in Eq. 2.6

$$L^{x}_{\mathrm{RTD}} = \sum_{t=1}^{\bar{X}} \begin{array}{l} 1 \quad x^{00}_{(t)} = x_{(t)} \quad \log D \quad x^{00}; \mathbf{h}; t \\ 1 \quad x^{00}_{(t)} \neq x_{(t)} \quad \log \quad 1 \quad D \quad x^{00}; \mathbf{h}; t \end{array} \tag{2.5}$$

$$L = L_{\mathrm{contrast}} + L_{\mathrm{RTD}} \tag{2.6}$$

2.2.4 News Segmentation Models

Systems for segmenting news using machine learning have been studied before. This work can be divided into two categories [24]:

- **Metadata-dependent** methods use metadata provided by broadcasters, such as audio-video watermarks and closed captioning
- **Presentation style** methods use presentation styles of the broadcasts and use features defined by domain experts to segment video through rule-based systems or machine learning algorithms

A notable early presentation style method used a two-level, multi-modal framework to segment news videos into single-story semantic units. The shot level analysis classifies video shots into predefined categories using low-level and high-level features. In contrast, the scene-level analysis employs Hidden Markov Models (HMM) to identify story boundaries [10]. A more recent system segments TV broadcast videos into four granularities: broadcast, program, story, and shot. The system is designed to identify and categorize three types of shows based on their presentation formats and metadata [24].

Chapter 3

RPCSE - Relative Placement Contrastive Learning of Sentence Embeddings

One contribution of this thesis is to explore and improve on existing work to further reduce the presence of anisotropy in word embeddings derived from NNLMs, improving the performance of LLMs on downstream tasks, particularly in semantic textual similarity (STS) and which is core to the Liquid News system. As mentioned in 2, contrastive learning has shown promise in reducing anisotropy while pushing state-of-the-art performance (SoTA) on NLP tasks. Specifically, the SimCSE model described in 2.2.2 shows SoTA performance on STS and SICK-Relatedness tasks. Due to this performance, I present a model (built in conjunction with Julian Manyika and Christian Belser - MIT EECS) that builds on SimCSE to improve performance and reduce anisotropy using a modified objective function to leverage the entire SNLI dataset [12].

3.1 Dataset Modification

The SimCSE model was trained on the Stanford Natural Langauge Inference (SNLI) corpus [12]. For a datum within the SNLI dataset, x, we are provided an entailment

Text	Judgments	Hypothesis
A man inspects the uniform of a figure in some East Asian country.	contradiction C C C C C	The man is sleeping
An older and younger man smiling.	neutral N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	contradiction C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	entailment E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	neutral N N E C N	A happy woman in a fairy costume holds an umbrella.

Figure 3-1: Describes how entailment, negatives, and neutral sentences pairs are defined with the SNLI Corpus

pair $(x; x^+)$, contradiction pair $(x; x^-)$, and a neural pair $(x; x^?)$. These relations are examples in figure 3-1. The SimCSE model only leverages entailment pairs and contradiction pairs as parameters of its loss function. RPCSE utilizes neutral sentences in addition to positive and negative sentences. The role of the neutral sentences is detailed in section 3.2.1.

As mentioned in section 1, a significant issue with the large transformer-based models used as the sentence encoder for SimCSE is the presence of anisotropy. Therefore, in this paper, we also test the performance of the standard SimCSE model using a whitened BERT pre-trained model [29, 49].

3.2 RPCSE Architecture

3.2.1 Loss Function Modification

To account for neutral sentences in the SNLI dataset [12], two alternative loss terms that can be used in isolation or as a linear combination to generate the overall loss were developed.

SimCSE-n Loss

SimCSE-neutral (SimCSE-n) loss mirrors the original SimCSE supervised loss function shown in Eq. 3.1. The only difference is the addition of the neutral sentences in

the cross-entropy calculation:

$$L^{0}_{\text{CE}} = \log \frac{e^{\text{sim}(h_i;h_i^+)=}}{\sum_{j=1} \left[e^{\text{sim}(h_i;h_j^+)} + e^{\text{sim}(h_i;h_j)} + e^{\text{sim}(h_i;h_j^?)} \right]} \tag{3.1}$$

This cross-entropy loss pulls the anchor sentence embedding and its corresponding entailment sentence embedding closer together. It pushes apart the similarity between the anchor sentence and the entailment, contradiction, and neutral sentences for different anchor sentences. SimCSE-n was attempted in [19], but it performed worse than the cross-entropy objective that considered only the entailment and contradiction sentences (Eq. 2.3). The hypothesis is that SimCSE-n can improve performance when paired with an auxiliary objective considering neutrals, such as the objective detailed in the following section.

Relative Placement Loss

Relative placement (RP) loss encourages the neutral sentence to be represented as less similar to the anchor sentence than the entailment sentence but more similar to the anchor sentence than the contradiction sentence, as shown in Figure 3-2. This means that the cosine similarity between the embeddings of the anchor sentence (h_i) and the neutral sentence ($h_i^?$) must be less than that of the anchor sentence and the positive sentence (h_i^+), but greater than that of the anchor sentence and the negative sentence (h_i).

To reflect the relative placement of neutral sentences in the embedding space, we defined the relative placement auxiliary loss function in Eq. 3.2 (N is the mini-batch size).

$$L_{RP} = \sum_{j=1}^{N} `_{\text{TMS}}(h_j; h_j^+; h_j^?) + `_{\text{TMS}}(h_j; h_j^?; h_j) \tag{3.2}$$

The $`_{\text{TMS}}$ term is a triplet margin similarity function, which calculates triplet margin loss using cosine similarity as the distance metric and treating the three parameters

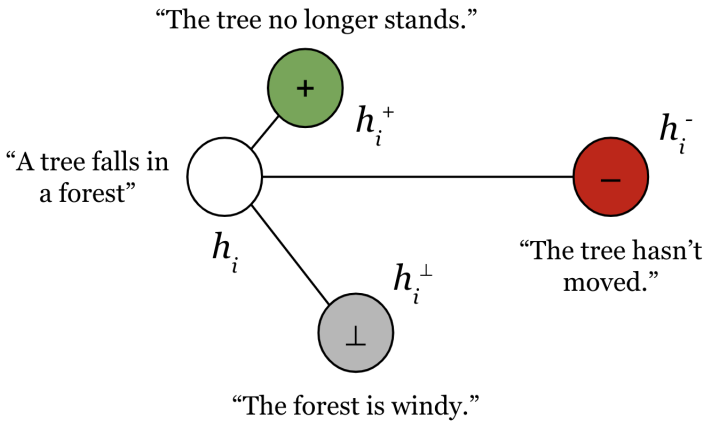

Figure 3-2: Shows the desired relative placement of the neutral sentence with respect to the anchor, positive and negative sentences

as the anchor, close and far sentences, respectively:

$$`_{TMS}(h_a; h_c; h_f) = \max\{\text{sim}(h_a; h_f) \quad \text{sim}(h_a; h_c) + m; 0\} \tag{3.3}$$

In Eq. 3.2, $`_{TMS}(h_j; h_j^+; h_j^?)$ will be positive if the cosine similarity between $h_j$ and $h_j^+$ is not greater than the cosine similarity between $h_j$ and $h_j^?$ by at least a margin of $m$. Similarly, $`_{TMS}(h_j; h_j^?; h_j)$ will be positive if the cosine similarity between h_j and $h_j^?$ is not greater than the cosine similarity between h_j and h_j by at least a margin of m, where m is positive. The linear combination of these triplet margin losses serves as a way of penalizing cases when the similarity between the anchor and neutral sentence representations is greater than the similarity between the anchor and positive sentence representations or less than that of the anchor and negative sentence representations.

Chapter 4

Liquid News

4.1 Problem

As outlined in 1, Liquid News aims to build a platform that helps users parse the news and expose users to different perspectives on current topics. At the system level, Liquid News aims to solve two technical tasks - segment news in an unsupervised manner and cluster news segments based on semantic content. These tasks closely mirror the k-means task and the WikiSection machine reading task described in 2.1.3 and 2.2.1, respectively; thus, we use them as a guide for framing the Liquid News tasks. More formally, the objectives of Liquid News can be defined as a segmentation task and a clustering task.

4.1.1 Clustering Task

Given a collection of videos $V = [v_1; : : : ; v_n]$, assign each video v_i to a cluster from an unknown collection $C = [c_1; : : : ; c_k]$, where jCj is not known a priori. The solution to this task will be $k = jCj$ and a $jV j$ x jCj binary matrix A, such that A_{ij} represents if video v_i belongs to cluster c_j.

Liquid News solves this task at the topic and subtopic level. At the topic level, the goal is to cluster a collection of videos V into clusters C_{TOP} with size $k_{TOP} = jC_{TOP}j$. At the subtopic level, the goal is for every topic cluster C_{SUB} to cluster a collection

of video segments S into clusters C_{SUB} with size $k_{SUB} = |C_{SUB}|$.

4.1.2 Segmentation Task

Given a video $V = \langle T; S = ; i$, and a collection of consecutive sentences $T = [T_1; : : : ; T_N]$, split V into a collection of disjoint segments $S = [S_1; : : : ; S_M]$. Each segment $S_i = \langle T_i; l_i i$ contains a sequence of sentences $T_i \quad T$ and a subtopic label l_i that describes the topic of the sentences.

Liquid News solves this task in an unsupervised manner as there is no ground truth for subtopics labels l_i and segments S.

4.2 User Interface

The primary end-user objective of the Liquid News platform is to help users parse the news and expose them to different perspectives. Solving the clustering and segmentation tasks in 4.1 provides us with the data to address this objective, but the solution is primarily a user-interface problem. The following sections outline the initial approaches to the Liquid News interface and the system's final interface, outlining the shortcomings that lead to the final design.

4.2.1 Initial Design

The initial interface takes a graphical approach to representing the topic and subtopic clustering. Each topic cluster is associated with its own $x; y$ plane shown in 4-1. Segments are arranged within each $x; y$ plane based on their word embedding. Note that a word-embedding is a dense high-dimensional vector that can not be expressed in two dimensions. The interface uses principle component analysis (PCA) to reduce the dimensionality to two dimensions. This interface does not explicitly use the notion of subtopics as it predates the second level of clustering described in 4.1. Each point on the $x; y$ plane corresponds to a segment (termed a clip in the user domain). When a clip is selected, the user can watch it in the top right corner of 4-2. The left-hand

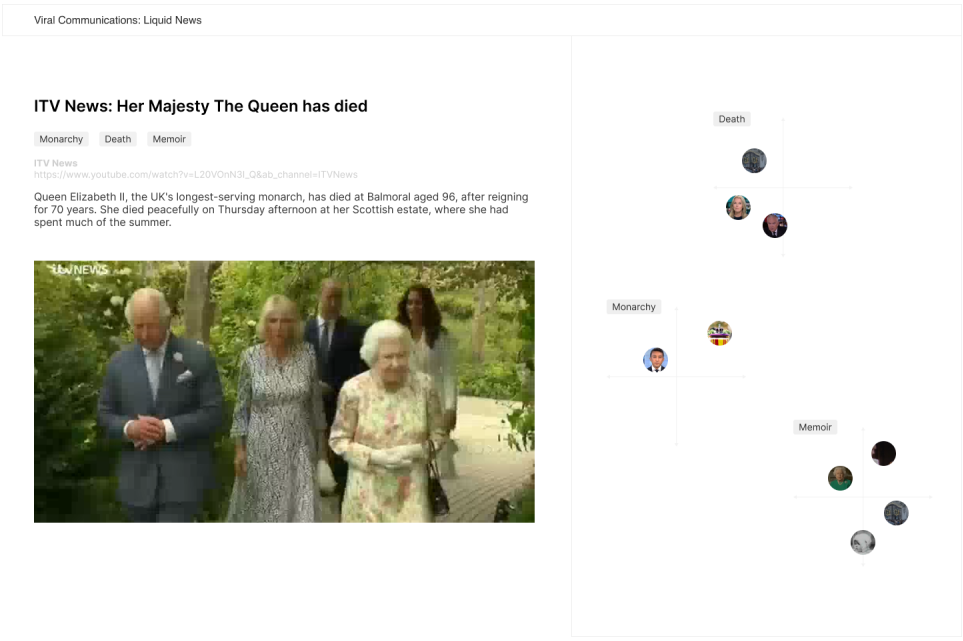

Figure 4-1: View A of the initial interface. View A shows the topical clustering, representing the first layer of the relation between videos identified by Liquid News.

side of 4-1 is a collection of all the original videos. The original, full-length videos are provided for the possibility that users want to watch the full video associated with a clip. User testing indicated major shortcomings in the initial interface. Although PCA reduction allowed the spatial clustering of the clips to be shown in two dimensions, it led to many questions regarding the meaning of each axis. Labeling each axis is impractical because the ability to express these latent variables as natural language has yet to be discovered. This fault motivated the second clustering layer at the subtopic level described in 4.1. Another shortcoming was including the full-length video within a single page. Many users found it unnecessary as they preferred the clips themselves and wanted more space allocated towards them. This representation failed because the platform's purpose was to provide more explainability of how news segments are related, and this UI made the task more difficult.

4.2.2 Final Design

Leveraging the insights from the initial design, the final UI for the platform was overhauled to a grid-based approach that mirrors many existing video platforms, such

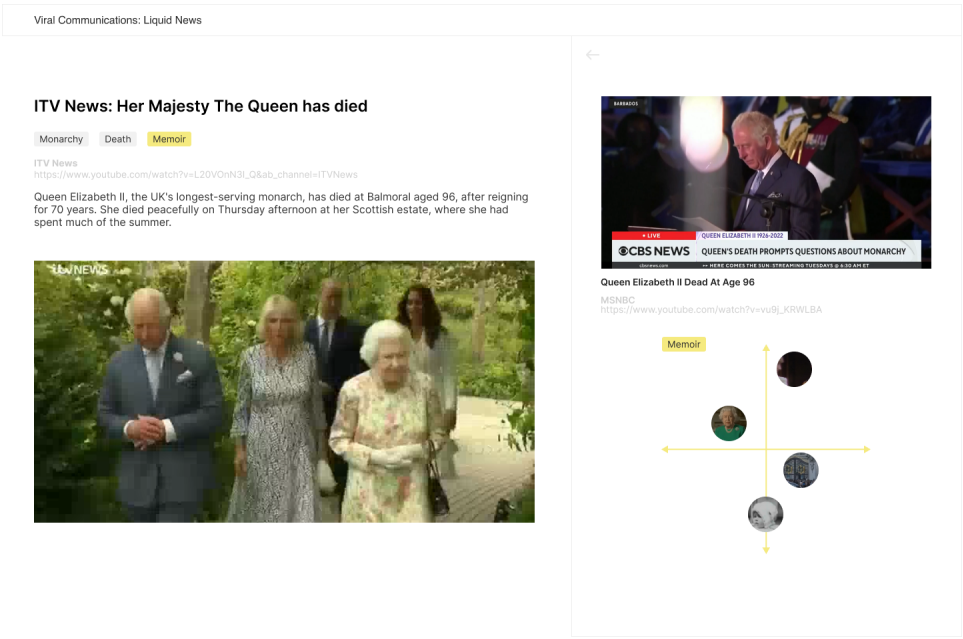

Figure 4-2: View B of the initial interface. View B shows the intra-topic relation between clips identified by Liquid News. All clips for a given topic are reduced into a 2-dimensional plane.

as YouTube and Vimeo. A critical factor in this decision was to provide a familiar user experience (UX), reducing the learning curve needed to use Liquid News. The final design is split into four key sections: topic selector, subtopic selector, clip selector, and clip player. The topic selector is the collection of buttons seen in the top left-hand side of Fig. 4-3. The subtopic selector is the button collection on the top right-hand side. When a topic button is selected, the subtopic-selector presents the associated subtopics. The clip player in the bottom-right presents a video player to view each clip and a description containing the following essential data: title, source, and summary. The clip selector is the large gird on the right. For each subtopic, a grid of associated clips is presented. The clips are sorted in the semantic manner described in 4.3.2. When a clip for the gird is selected, it updates the viewing with the data associated with the clip. The ability of this UI/UX to accomplish the objective of Liquid News is addressed in 5.2.2.

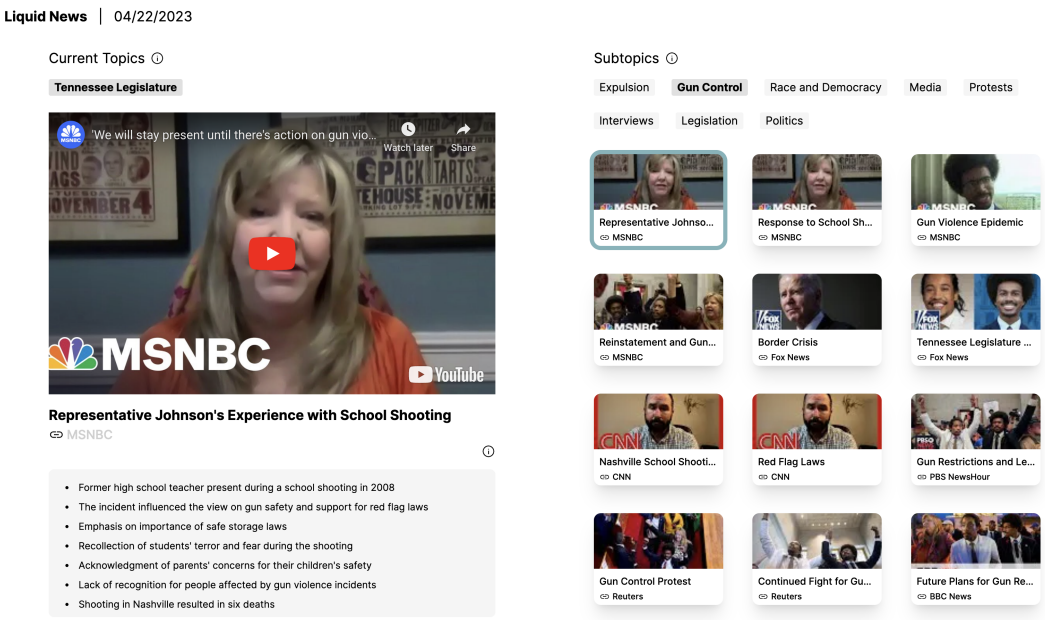

Figure 4-3: Final user interface of the Liquid News platform that was launched to end users. The design builds on the shortcomings of the initial designs and implements the structure of many modern video platforms for user familiarity and ease of use.

4.3 System Architecture

The Liquid News architecture contains three planes: backend, frontend, and storage. The back-end and storage planes solve the segmentation and clustering tasks. The front-end and storage plane operate together, leverage the solution to the tasks, and present end users with a platform that aligns with the goals of providing helpful parsing of the news and UI that exposes users to a broader range of perspectives.

4.3.1 Storage

The storage plane of the Liquid News system is responsible for maintaining the intermediate data for each video as the segmentation and clustering tasks are solved and must maintain the solutions until a new job is initiated. This plane contains a long-term MongoDB database that stores metadata and a short-term Amazon S3 database for video storage. The Amazon S3 database is only used for small-volume, short-lifespan videos for evaluation. Therefore, the primary data store for the system is the MongoDB database. The system operates on a large volume of videos, so to

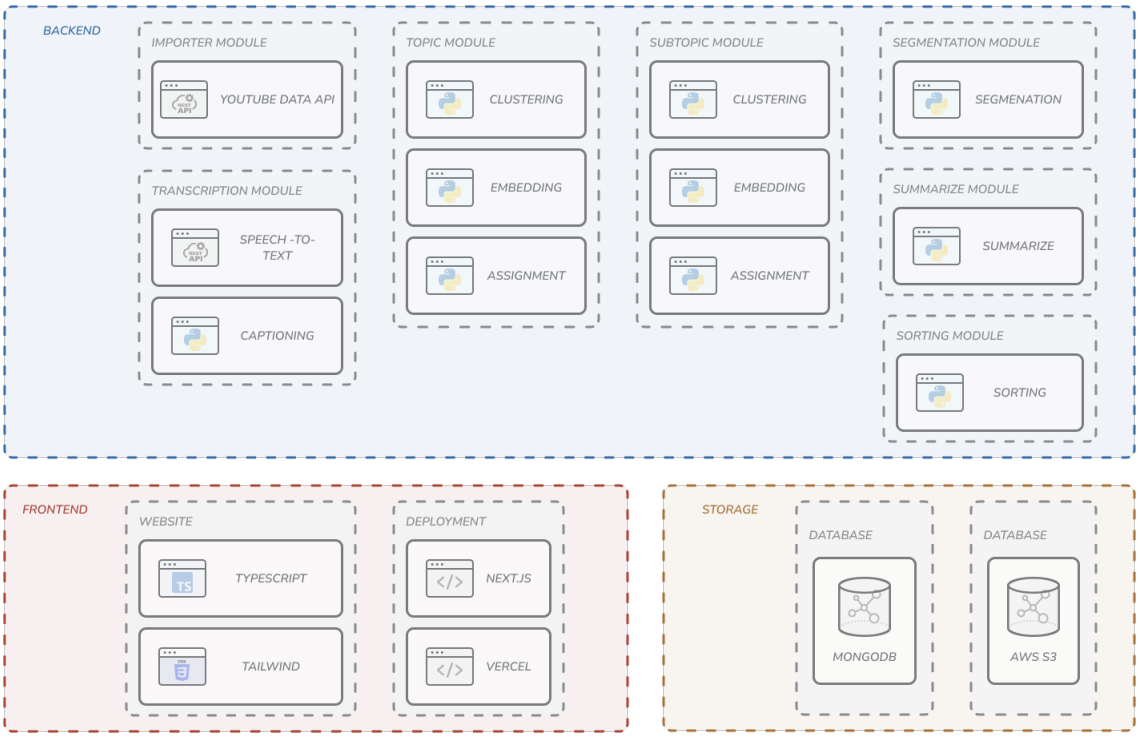

Figure 4-4: A high-level module based on the Liquid News System representation. The system comprises three planes: back-end, front-end, and storage. Each container contains modules that operate in orchestration or a pipeline.

optimize for space, no video data is maintained in the MongoDB database. Instead, videos are indexed using video identification strings provided by the YouTube API. The primary database for the system contains three collections: channels, metadata, and topics. The channel collection contains documents corresponding to the YouTube channels that serve as video sources for the systems. The schema for a document in this collection is shown in Table 4.1.

The metadata collection contains all data for a video. Many fields are metadata provided by YouTube, as shown in the full schema in 4.2. In addition, each video is

Field	Data Type
`channel`	String
`channel_id`	String
`latest_pull_status`	String
`timestamp`	Number

Table 4.1: MongoDB channel collection document schema, the `channel_id` is the index used by the importer module to identify the correct channel

Field	Data Type
publishedAt	String
channelId	String
title	String
description	String
channelTitle	String
liveBroadcastContent	String
publishTime	String
videoId	String
transcription	String
sentence_captions	Array
topic	String
segments	Array

Table 4.2: Schema for a doc in the metadata collection. The bold fields topic and segments are solutions to the clustering and segmentation task

Field	Data Type
title	String
shift_phrase	String
start_timestamp	Number
end_timestamp	Number
start_index	Number
end_index	Number
segment_transcription	String
subtopic	String
embedding	Array

Table 4.3: Schema for an object in the segments field of a doc in the metadata collection. The bold field subtopic is the solution to the clustering task at the segment level.

indexed in the database and on YouTube using a unique video identification string called `videoID`. Furthermore, each document contains two key fields generated by the Liquid News backend: `transcription` and `sentence_captions`. These fields are the text corpus data used to solve the segmentation and clustering task. Finally, each doc contains the `topic` field, which defines the cluster a video belongs to, and a `segments` array, which contains the segmentation for a given video. Table 4.3 shows the schema for each object in the segment array. The `topic` and `subtopic` fields are the solutions to the clustering task, and the `segments` is the solution to the segmentation task.

Table 4.4 defines the schema of the topics collection, a summarized and reorganized version of the metadata collection used to serve video data to the front end. Video data is aggregated by `topic`, with a document for each topic. Each topic aggregates associated video segments by `subtopic`. All segments for a given subtopic are sorted via their word embedding and stored in the `clips` field.

Field	Data Type
topic	String
topic_description	String
topic_embedding	Array
timestamp	String
subtopics_processed	String
subtopics	Object

Table 4.4: Topics collection document schema

4.3.2 Backend

Importer Module

The first module in the Liquid News backend pipeline is the importer. This module retrieves the metadata from the channels defined in the MongoDB database. The module makes a REST API call to the YouTube Data API for each channel in the channel collection. It requests the metadata for the `max_results` most recent videos, where `max_results` is a user-defined argument.

Transcription Module

The downstream modules in the pipeline rely on accurate transcription and captioning of every video. Initially, the transcription module relied on captioning provided by the YouTube Data API to generate transcriptions and sentence-level captioning. However, this failed due to unreliable and error-prone captioning. Instead, the transcription module utilizes the SoTA Conformer speech-to-text model that builds on the well-known Conformer architecture [38]. Rather than running a local model, the system utilizes the latest Conformer-1 model from AssemblyAI. The Python YouTube-DL library is used to download MP3 audio for each video via `videoIDs` in parallel, in batches of 32. Then in parallel, each audio file is transcribed and captioned at the sentence level using the AssmeblyAI Conformer-1 API. The `transcription` and `sentence_captions` fields in Table 4.2 are updated.

Role	Content
System	You are a helpful assistant who strongly prefers being unbiased and concise.
System	Return a list of unique key topics for these titles along with a description of the topic. The number of topics must be significantly shorter than the number of titles listed below.
User	`prompt`

Figure 4-5: Few-shot chaining prompt used to guide the GPT-4 to identify the underlying clustering for the list of videos and decoding the latent representation of each cluster in text. Where `prompt` is the list of video titles joined with their respective descriptions.

Topic Module

The topic module is responsible for assigning each video in the metadata collection a topic. The list of topics that can be assigned to each video is the smallest clustering that expresses the latent clustering of all the video titles. We leverage the GPT-4 large language model to identify this set of topics. Rather than building or fine-tuning an existing language model, the topic module uses prompting as a form of few-shot learning to solve the clustering tasks. Recent work signals that few-shot learning is more robust than fine-tuning [45]. For each video in the collection, we form a sub prompt tp_i = `title + description` from metadata shown in Table 4.2. The concatenation of all the sub prompts $\sum_{i=1}^{n} tp_i$ = `prompt`. We then prompt the model with a primer chain that expresses the clustering task in natural language, as shown in Fig. 4-5.

Liquid News uses GPT-4 due to its longer context window and superior responses. The response from the prompting in Fig. 4-5 solves the clustering task in natural language. We will denote this solution as $C_{TOP}; jC_{TOP}j = k_{TOP}$. Although the cluster size k_{TOP} and the cluster meanings C_{TOP} are known, the natural language solution does not assign the titles to their cluster. The topic module solves this issue by assigning each cluster to the topic C_{TOP_i} with the largest cosine similarity by converting each sub prompt tp_i and topic C_{TOP_i} into a word embedding using the `text-embedding-ada-002` embedding model from Open-AI, shown in Alg. 1.

Algorithm 1 Topic Assignment

```
procedure AssignTopicToVideos(topic_embeddings, titles)
    topics_to_videos ← {}
    for title in titles do
        best_topic ← arg max_topic cosine_similarity(
                        get_embedding(title; model); topic_embeddings[topic])
        if best_topic in topics_to_videos then
            topics_to_videos[best_topic]:append(title)
        else
            topics_to_videos[best_topic] ← [title]
        end if
    end for
    return topics_to_videos
end procedure
```

Role	Content
System	You are a helpful assistant who strongly prefers being unbiased and concise.
System	Identify when there is a major topic change and return the segment in the following JSON format: `{"0": {"title": "", "shift_phrase": ""}}`. The `title` is a word/phrase to describe the segment. The `shift_phrase` must be a single unique sentence in which the shift occurs. The `shift_phrase` must be verbatim what is said in the transcript and must appear in the order they come in the text. The segmentation must be broader and have fewer segments when possible.
User	`transcript`

Figure 4-6: Few-shot chaining prompt used to guide the GPT-4 to identify segment boundaries in a video transcription.

Segmentation Module

The segmentation module is responsible for solving the segmentation tasks defined in 4.1.2. Following the intuition from the topic module and leveraging the task-agnostic ability of GPT-4, we define the segmentation task as a chain of natural language prompts. First, the model is provided with the transcription for a video. Once given the transcription, it is asked to identify sentences with a topic shift. This mirrors SECTOR's approach in defining the segmentation problem at the sentence level, as shown in Eq. 2.2 and identifying edges in the latent space. The chain of prompting for this natural language framing for the task is shown in Fig. 4-6.

Remember the solution to the segmentation class is a segmentation of the video $S = [S_1; ...; S_M]$, $S_i = \langle T_i; l_i \rangle$ where T_i is a subset of all sentences $T = [T_1; ...; T_N]$.

The response from the model only gives us the $T_i[0]$ $\forall$ i. T_i, is reconstructed using the sentence level captioning and each T_i as boundaries as shown in Alg. 2.

Algorithm 2 Segmentation - the core functionality of the GetSegments function. The algorithm iterates through the segments and captions, comparing the shift phrases in segments to the text in captions. The edge and base cases for the start and end of the list are excluded from this representation.

```
procedure GetSegments(segments; captions)
    for i  0 to len(segments)  2 do
        first_topic_shift  segments[i][shift_phrase]
        second_topic_shift  segments[i + 1][shift_phrase]
        start; start_index; end; end_index  None; None; None; None
        for j  0 to len(captions)  1 do
            if string_comparer(first_topic_shift; captions[j][text]) then
                start  captions[j][start]
                start_index  j
                break
            end if
        end for
        for k  0 to len(captions)  1 do
            if string_comparer(second_topic_shift; captions[k][text]) then
                end  captions[k][start]
                end_index  k
                break
            end if
        end for
            segments[i][segment_transcription]
        get_segment_transcript(start_index; end_index; captions)
    end for
end procedure
```

Subtopic Extraction Module

The subtopic module solves the clustering tasks at the segment level. The clustering task at the segment level requires solving the task for each topic T_i to identify subtopics $S_i = [S_{1i}; : : : ; S_{ki}]$ for all i in set T. The methodology by which the clustering is solved for each topic is identical to the topic level with one minor change. Due to the restriction of the context window of the GPT-4 (appx. 6k words), the sub-prompt tp is simplified to $tp =$ `title`. The subtopic module runs into a context window limit because $jtp_{SUB}j >> jtp_{TOP}j$. This is apparent given the formulation of the following tp_{SUB}. Given a topic cluster T_i, the set of all titles that must be

clustered into S_i is defined as:

$$tp_{SUB} = \bigcup_{j \in I} S_j$$

Where I is the set of videos belonging to topic cluster T_i and S_j is the set of segments for each video j. Each segment is assigned to its corresponding subtopic using the assignment algorithm in Alg. 2.

Sentiment Sorting Module

The sentiment sorting module orders all segments corresponding to a given subtopic S_{ij} (where i indexes the topic and j indexes the subtopic) relative to their semantic meaning. The module creates this ordering by converting each segment into the word embedding space, creating a high dimensional dense vector. Then, similar to the methods seen in 4.3.2, the segment is ordered from greatest to least cosine similarity to the average word embedding vector. The intuition is that the latent vector space will reflect complex properties such as perspective, bias, and tone. Thus, this sorting method will cause videos that share similar complex properties to lie near each other in the sorting.

Algorithm 3 Embedding Sort

```
1:  procedure SortClipsViaAvgEmbedding(subtopic)
2:      embeddings = []
3:      for clip in subtopic.clips do
4:          if "embedding" in clip then
5:              embeddings:append(clip:embedding)
6:          end if
7:      end for
8:      avg_embedding = np:mean(embeddings; axis = 0)
9:      for clip in subtopic.clips do
10:         if embedding in clip then
11:             clip_embedding = clip:embedding
12:             clip:cosine_similarity = cosine_similarity(avg_embedding; clip_embedding)
13:         end if
14:     end for
15:     subtopic:clips:sort(key = lambda x : x:get(cosine_similarity; [0])[0])
16:     return subtopic
17: end procedure
```

4.3.3 Frontend

The Liquid News frontend is a lightweight single-page site built on the Next.JS framework and Typescript. The data is served by the `.../api/topics` endpoint that reads from the topics collection in the MongoDB database. The `liquid` component parses the JSON response into the interfaces defined in Fig. 4-7.

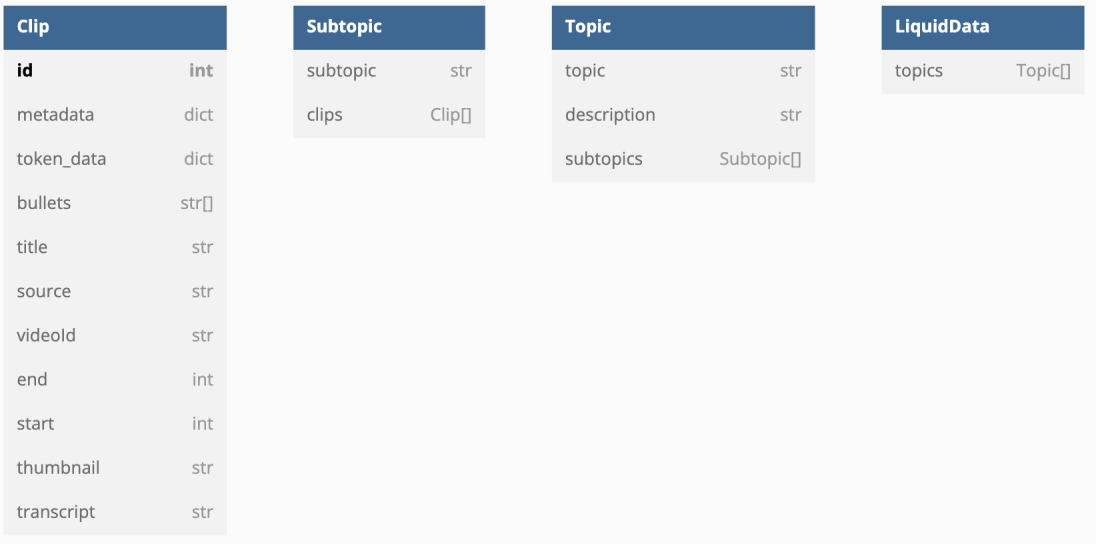

Figure 4-7: TypeScript Interfaces

Chapter 5

Evaluation

5.1 RPCSE - Relative Placement Contrastive Learning of Sentence Embeddings

5.1.1 Experiments

We hypothesized that pairing the RP objective in Eq. 3.2 with the contrastive objective shown in Eq. 2.3 or the cross-entropy objective shown in Eq. 3.1 from SimCSE-n, would improve sentence embeddings and performance on downstream textual similarity tasks. Eq. 5.1 and 5.2 show a pairing of the SimCSE or SimCSE-n and RP objectives can balance the contrastive and relative placement objectives for the embedding space (where is a hyperparameter controlling the weight given to the RP objective):

$$\text{SimCSE RP Loss} = L_{CE} + \quad L_{RP} \tag{5.1}$$

$$\text{SimCSE-n RP Loss} = L^{0}_{CE} + \quad L_{RP} \tag{5.2}$$

SimCSE-n and RP objectives were also tested separately to understand each objective's effect. The SimCSE model was a baseline for the RPCSE models with varying objective functions against various semantic similarity tasks.

Metrics

The evaluation focused on classifying the entailment relation between sentence pairs and the STS task of quantifying the similarity between two sentences. There are three relations. (1) Entailment sentence pairs have the same continuity in meaning. (2) Contradiction sentence pairs contradict one another. (3) A Neutral sentence pair neither entails nor contradicts one another. The sentence similarity is judged on a 6-point scale (0 being unrelated and 5 being the same). Figure 5-1 shows a few basic examples for each point on the scale.

5	*The two sentences are completely equivalent, as they mean the same thing.*
	The bird is bathing in the sink. Birdie is washing itself in the water basin.
4	*The two sentences are mostly equivalent, but some unimportant details differ.*
	Two boys on a couch are playing video games. Two boys are playing a video game.
3	*The two sentences are roughly equivalent, but some important information differs/missing.*
	John said he is considered a witness but not a suspect. "He is not a suspect anymore." John said.
2	*The two sentences are not equivalent, but share some details.*
	They flew out of the nest in groups. They flew into the nest together.
1	*The two sentences are not equivalent, but are on the same topic.*
	The woman is playing the violin. The young lady enjoys listening to the guitar.
0	*The two sentences are completely dissimilar.*
	The black dog is running through the snow. A race car driver is driving his car through the mud.

Figure 5-1: Similarity scores with example sentences for each one.

We have seven subtask datsets. STS12 [4] and STS13 [5] datasets come from machine translation evaluation and newswire headlines. STS14 [2] and STS15 [1] use tweets, newswire headlines, and image descriptions and introduces datasets in different languages, such as Spanish. STS16 [3] and STS-Benchmark [9] datasets focus on

Hyperparameter	Value
batch-size	128
learning-rate	5 10 ^{5}
epochs	3
Pooling method	CLS
Temperature	0.05

Table 5.1: Hyperparameters used for testing RPCSE.

plagiarism detection, machine translation, and question-answering. SICKRelatedness [31] datasets included examples of lexical syntactic and semantic phenomena.

Spearman correlation evaluates how well the supervised learning model performed against each test dataset, as done in the SimCSE [19] paper. Spearman correlation was chosen because it measures rankings instead of actual scores, which better suits the needs of evaluating sentence embeddings.

5.1.2 Configuration

A mix of BERT-Base-Uncased, BERT-Large-Uncased, and whitening transformations was used for word embeddings. Parameters were fine-tuned according to the optimal setting given by [11], shown in Table 5.1.

5.1.3 Results

Table 5.2 shows that two trends occurred for the best configuration of SimCSE RP Loss which has = 0:1 and m = 0:1. (1) SimCSE RP Loss follows a similar trend as SimCSE Bert-Base model, and (2) a decrease in performance by 4.09 % from the SimCSE BERT-Base model. The general decrease in performance suggests neutral data points don't significantly influence STS tasks. The SimCSE baseline model does not use neutral data points, while neutral points had a significant role in SimCSE RP Loss.

Like the SimCSE model, STS13, STS15, and STS-Benchmark remained the best-performing tasks. STS13 [5] focuses on similarity in English pairs for news headlines and glosses. STS15 [1] focused on many question-and-answer forums, news head-

lines, and image descriptions. STS-Benchmark [9] had mainly image captions, news headlines, and user forums. STS12, STS14, and SICK-Relatedness were amongst the worst-performing tasks. STS12 [4] had news, videos, and machine translation evaluation. STS14 [2] contained news headlines and summaries, forum posts, glosses, and tweet-news pairs. SICK-Relatedness [31] contained image description data and video description data. Tasks with videos in their data sets had worst performance than those without.

Specific STS tasks had a more significant decrease in performance than others. STS13 experienced a decrease in Spearman correlation by 4.62%, STS14 by 6.84%, STS15 by 5.40%, and STS-Benchmark by 5.35%. While STS12 only experienced a decrease of 1.51%, STS16 by 2.62%, and Sick-R by 2.25%. These performance differences are related to the content of the data sets and sub-tasks in each STS task. A noticeable difference in the dataset is that STS14, STS15, and STS-Benchmark are tasks in which the evaluation data set contains image descriptions. For each, at least 20% of their data were image descriptions. It's an interesting coincidence that these tasks had the most drastic decline in performance compared to SimCSE. STS13 and STS14 were also the only ones that contained glosses in their data sets, while they had the largest difference in performance compared to the SimCSE baseline. The change in the SimCSE model to incorporate neutrals is greatly affected by tasks that contain these aspects of the origin of the data set.

5.1.4 Analysis

To see the qualitative impact on the embedding space of the RPCSE models, similar to [11] a small-scale retrieval experiment using SimCSE-BERT-base, RPCSE-n RP Loss (BERT-base), and RPCSE RP Loss (BERT-base) on a 1000 sentences sampled from STS-B dataset was conducted. Table 5.3 shows the quality of the top three most similar sentences to the query string for both the RPCSE-n RP Loss and RPCSE RP Loss are very similar to those of SimCSE. However, the cosine similarities are 0.0363 and 0.0385 less on average.

We further expanded these tests to analyze the three least similar examples shown

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
SimCSE Baseline								
BERT-Base	75.30	84.67	80.19	85.40	80.82	84.26	80.39	81.58
BERT-Large	75.78	86.33	80.44	86.06	80.86	84.87	81.14	82.21
BERT-Base-Whitened	67.96	77.96	72.95	84.76	79.96	81.63	77.90	77.59
BERT-Large-Whitened	68.89	83.44	75.14	84.22	80.73	82.43	74.65	78.50
RPCSE Baseline								
SimCSE RP Loss (5.1)								
= 0.1; m = 0.05	74.34	79.73	73.13	79.44	77.60	78.62	78.26	77.30
= 0.1; m = 0.1	73.79	80.05	73.35	80.00	78.20	78.91	78.14	77.49
= 0.1; m = 0.5	73.81	79.84	73.18	79.79	78.13	78.57	77.84	77.31
= 0.2; m = 0.1	73.43	79.85	72.87	79.84	77.82	78.80	77.16	77.11
SimCSE-n RP Loss (5.2)								
= 0.1; m = 0.1	73.52	79.43	73.35	80.76	78.62	79.18	78.35	77.60
= 0.1; m = 0.5	73.47	79.24	73.30	80.70	78.62	79.13	78.16	77.52
= 0.2; m = 0.1	74.10	76.72	72.16	79.97	77.68	79.00	77.63	76.75
SimCSE-n (3.1)								
= 0.1; m = 0.1	73.91	79.40	73.18	81.14	78.55	79.12	78.71	77.72
RP Loss (3.2)								
= 0.1; m = 0.1	63.45	52.67	54.47	69.12	68.86	63.66	61.64	61.98

Table 5.2: Performance of the RPCSE model with both the SimCSE RP Loss and the SimCSE-n RPLoss, as well as the baseline SimCSE model with varying BERT pre-trained sentence embeddings. The SimCSE RP Loss (Eq. 5.1) experiment and the SimCSE-n RP Loss (Eq. 5.2) experiments used a BERT-Base uncased model for sentence embeddings. Note that represents the weight given to the L_{RP} and m is equal to the margin defined in 3.3.

Query: a man is playing music			
	Supervised SimCSE-BERT	RPCSEn RP Loss - BERT	Similarity
1	a guy is playing an instrument	a guy is playing an instrument	0.9185, 0.8625
2	a man playing the guitar	a man is playing his guitar	0.8463, 0.8141
3	a man is playing his guitar	a man is playing guitar	0.8342, 0.8133

Table 5.3: Top three similar sentences to the query sentence for SimCSE-BERT model compared to RPCSE-n RP Loss model.

Query: a man is playing music			
	Supervised SimCSE-BERT	RPCSE RP Loss - BERT	Similarity
1	a guy is playing an instrument	a guy is playing an instrument	0.9185, 0.9353
2	a man playing the guitar	a man playing the guitar	0.8463, 0.7933
3	a man is playing his guitar	a man is playing his guitar	0.8342, 0.7883

Table 5.4: Top three similar sentences to the query sentence for SimCSE-BERT model compared to the RPCSE RP Loss BERT model.

in Table 5.5, and we found that the cosine similarities were, on average, 0.0024 and 0.001 less for RPCSE-n RP Loss and RPCSE RP Loss to SimCSE. These results suggest that the RPCSEn-RP Loss and RPCSE-RP Loss models lead to a greater dispersion in the embedding space for entailment pairs because although in a different order, the top three similar sentences in Table 5.3 and Table 5.4 are the same, but on average have a lower cosine similarity. Furthermore, the results in Table 5.5 and 5.6 suggest that the embedding space for both our models presents less anisotropy as the cosine similarity for the least similar sentences decreases, suggesting greater separation between entailment and contradiction pairs.

Query: a man is playing music			
	Supervised SimCSE-BERT	RPCSEn RP Loss - BERT	Similarity
1	the lady cracked an egg into a bowl	black and white cows behind a fence	0.005, 0.0020
2	ocean liner close to coast with houses in the background	three dogs running in the dirt	0.004, 0.0020
3	two men standing in grass staring at a car	wo cats, one ginger, the other white laying on a bed	0.004, 0.0018

Table 5.5: Least three similar sentences to the query sentence for SimCSE-BERT model compared to RPCSE-n RP Loss model.

Query: a man is playing music			
	Supervised SimCSE-BERT	RPCSE RP Loss - BERT	Similarity
1	the lady cracked an egg into a bowl	a woman is frying ground meat	0.005, 0.0036
2	ocean liner close to coast with houses in the background	our ladies in swimsuits play sand volleyball on the beach	0.004, 0.0033
3	two men standing in grass staring at a car	tan cows look closely at the camera	0.004, 0.0031

Table 5.6: Least three similar sentences to the query sentence for SimCSE-BERT model compared to RPCSE RP Loss model.

Table 5.7: Clustering Accuracy

Metric	Value
Videos Sampled	319
Unique Videos Sampled	271
Mean Watch Time	0:07:32
Topics Accuracy	0.818
Subtopics Accuracy	0.824
Joint Accuracy	0.721

5.2 Liquid News

5.2.1 Clustering Accuracy

Methodology

The clustering and segmentation ability of Liquid News was tested against a test dataset. The dataset is comprised of 20 videos from 8 channels on varying ends of the political spectrum: CNN, Fox News, MSNBC, BBC, The Washington Post, The New York Times, Bloomberg, and CBS. The videos were then processed using Liquid News to generate a topic clustering for all the videos and subtopic clusters for all the segments. There were 800 segments generated across all topics. Clustering and segmentation assignments were validated using a survey group of 319 US respondents. Respondents were given a randomly sampled segment and asked if the assigned topic and subtopic matched the content.

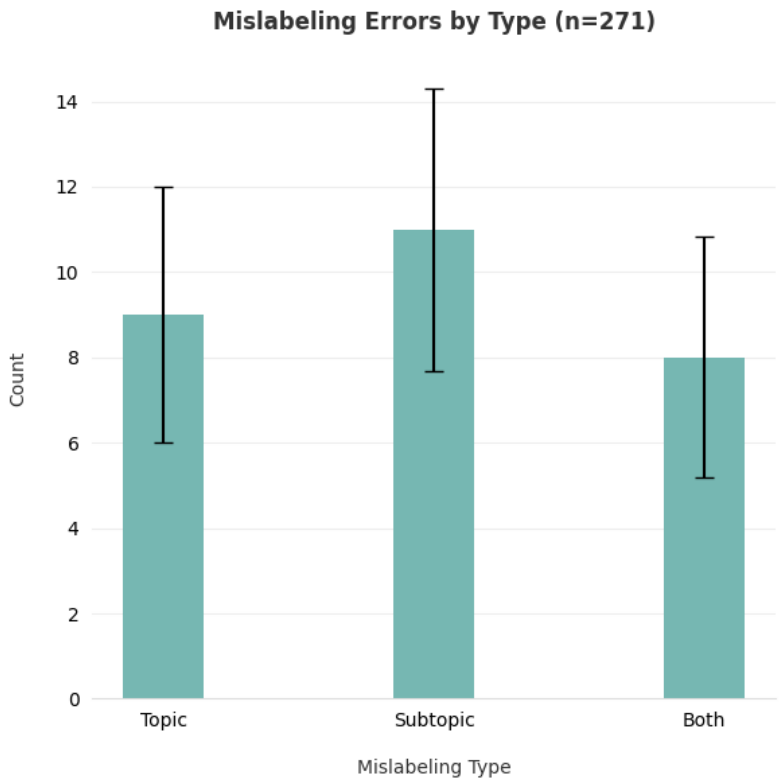

Figure 5-2: Mislabeling type distribution for mislabeling errors in clustering

Results

Out of 319 randomly sampled clips, 271 were unique. This means 33.88% of the full segments set were evaluated. Table 5.7 show the system performs well in clustering topics and subtopics when analyzed in isolation at 81.8% and 82.4%. However, when the accuracy of the clustering is expanded to check if video labels were correct at the topic and subtopic level, it drops significantly to 72.1%. To better understand why there is a significant drop-off, Fig 5-2 analyzes the distribution of the faults. The data shows 37.1% of errors are caused by an inaccurate topic, 34.8% by an inaccurate subtopic, and 28.1% by both being inaccurate. These numbers reveal that a single mislabeling causes the majority of errors. One hypothesis for the topic inaccuracy, when the segment cluster is correct, is because the segment is loosely related to the topic of the entire video it came from. A hypothesis for the subtopic inaccuracy is the segment lies close to two cluster boundaries. When both occur, a combination of both beliefs could occur.

5.2.2 User Interface

5.2.3 Methodology

For this evaluation, a dataset on a single current issue, the expulsion of two legislators from Tennessee, was created. The dataset contains 15 videos from six sources on different ends of the political spectrum: MSNBC, Fox News, CNN, Reuters, PBS News, and BBC. The videos were selected so that the total number of minutes per source was approximately 150. This ensured no one source dominated. These videos were then parsed using Liquid News to create the clustering and segmentation, and on YouTube, they were added to a playlist. A two-part survey was conducted on 112 users to evaluate users on both platforms. The first part of the survey was three questions to understand users' news consumption habits. The second half of the survey analyzed users' behaviors in parsing the dataset using both platforms. The questions presented to users in the first half of the survey are shown in Fig. 5-3.

- What platforms do you get your news from (select all that apply)?
  - Television
  - Internet News
  - Social Media Platform
  - Print Newspaper
  - Radio News
- Which platform do you prefer to get your news from?
  - Television
  - Internet News
  - Social Media Platform
  - Print Newspaper
  - Radio News
- How many hours a week do you engage with the news?
  - Less than 1 hour per week
  - 1-2 hours per week
  - 3-5 hours per week
  - 6-10 hours per week
  - More than 10 hours per week

Figure 5-3: First half of the user interface survey aims to identify user news consumption habits. The first two questions are presented in a randomized order.

The second half of the survey was focused on understanding how each platform helps users parse the news and if they are exposed to more perspectives using Liquid News compared to YouTube. Users were asked: *Use the website linked below to interact with a collection of news videos for 10-15 minutes. Then return to this survey and answer the following questions. Do not use another website during the interaction, and don't close out the survey. (Please make sure your browser is in full screen)* and were directed to YouTube or Liquid News at random. Following their interaction, users were asked questions to gauge how well they understood the content they watched, how well they identified the importance of topics, and how diverse the perspectives they engaged with were. The full set of questions is shown in Fig. 5-4. Note that for the second and fifth questions, the options displayed were limited to the subset of options the user selected for the first question.

Results

When analyzing the user interface of Liquid News, we take it in the context of the system's objectives - help users parse the news and expose them to more perspectives. When looking at the subtopics identified by the survey group, users who used Liquid News on aggregate identified the subtopics - racism, legislation, protests, gun control, and democracy more often - as seen in Fig. 5-5.

The difference in aggregate signals that Liquid News helps users identify subtopics compared to YouTube. Furthermore, when looking at how significant each subtopic is in Fig. 5-6, Liquid News users found the topics of race, legislation, and democracy far more critical than YouTube users. However, Fig. 5-7 indicates the average score for each subtopic across both platforms is nearly identical. This implies the range of topic importance scores of Liquid News users was greater than YouTube, suggesting that Liquid News helped some users parse the news better than others.

The fact that Liquid News users scored legislation, racism, and democracy higher on aggregate is significant because those topics represented a larger portion of the 800 segments. This indicates that Liquid News enabled some users to identify more significant talking points when compared to the YouTube test subjects. Liquid News

- Select all of the topics that correspond to the collection of videos.
  - Climate Change
  - LGBQT Rights
  - Geo Politics
  - Gun Control
  - Racism
  - Democracy
  - Legislation
  - Protests
- For each topic below, give the topic a score based on how relevant it was to the content you watched (7 being the most).
- How many videos did you watch (fully or partially)?
  - 1-2 videos
  - 3-4 videos
  - 5-6 videos
  - 7 or more videos
- Select all of the sources you viewed videos from.
  - BBC
  - CNN
  - Fox News
  - Reuters
  - PBS News
  - MSNBC
- For each of the topics, answer the following question: Did you notice di□erent viewpoints on this topic?

Figure 5-4: The second half of the user interface survey analysis how well users parse the news using Liquid News compared to YouTube and the diversity of the news they engage.

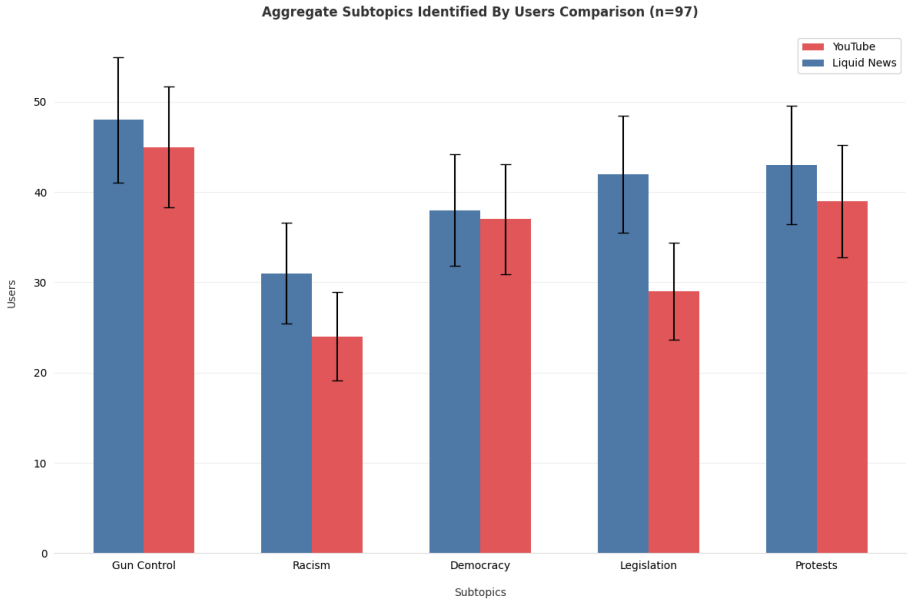

Figure 5-5: Breakdown of the aggregate count of users identified one of the five subtopics that corresponded to the content of the videos in the dataset.

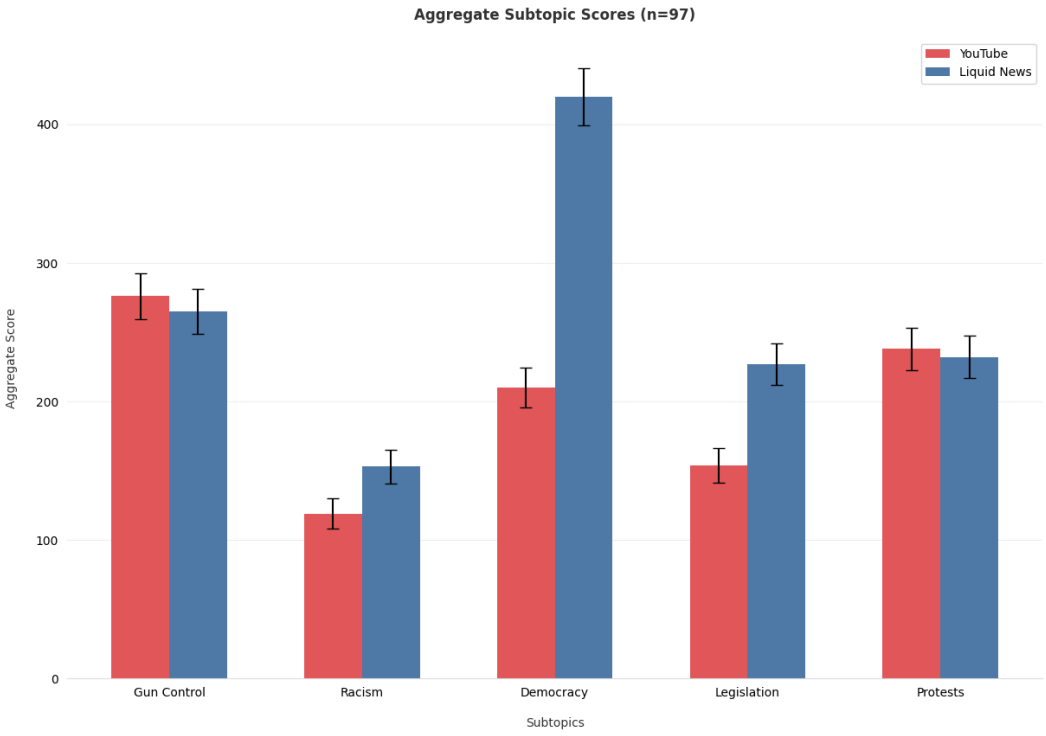

Figure 5-6: Comparison of the aggregate relative importance score of the five subtopics relating to the videos in the dataset across all surveyed users. The important rating ranged from 1-7 in increasing level of importance.

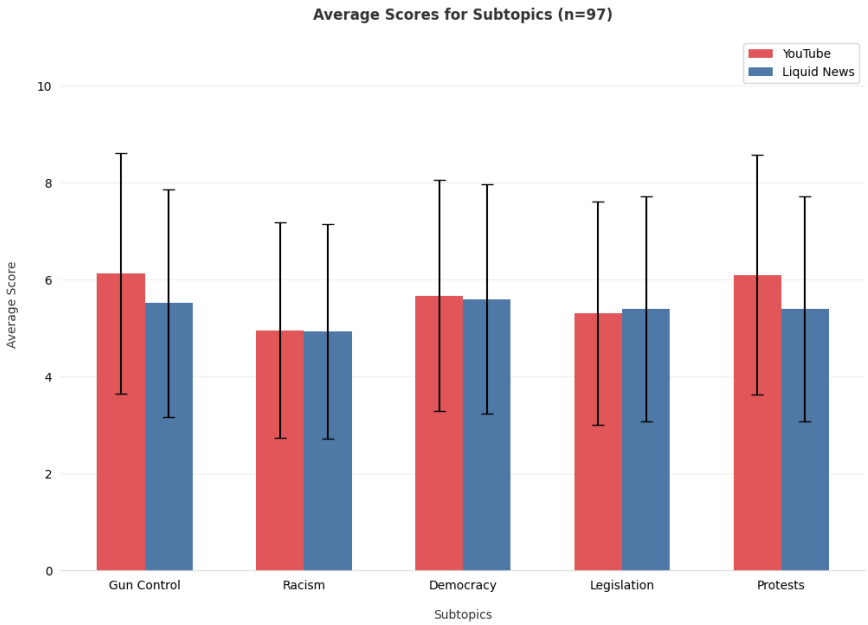

Figure 5-7: Comparison of the average relative importance of the five subtopics relating to the videos in the dataset. The important rating ranged from 1-7 in increasing level of importance.

and YouTube surveyors answered approximately the same for gun control and protest subtopics.

Regarding the goal of exposing users to more perspectives on the news, we look at the number of videos and unique sources surveyors watched. Furthermore, for the subtopics users identified, we analyze if they identified different perspectives. The distribution of the number of videos watched is shown in Fig. 5-8; the average for Liquid News and YouTube users were 3.786 and 3.93, respectively.

These numbers indicate that Liquid News did not lead to users viewing more videos on average. However, given that the median survey time was 12 minutes, Liquid News users likely saw more complete videos than YouTube users because the average YouTube video is significantly longer than a Liquid News clip. Furthermore, Fig. 5-9 shows that, on aggregate, Liquid News users watched a broader range of sources but, on average, watched a similar number of sources at 2.517 and 2.630, respectively.

No platform does an overwhelmingly good job of offering users different perspectives, as shown by Fig. 5-10. Furthermore, the difference between platforms is not

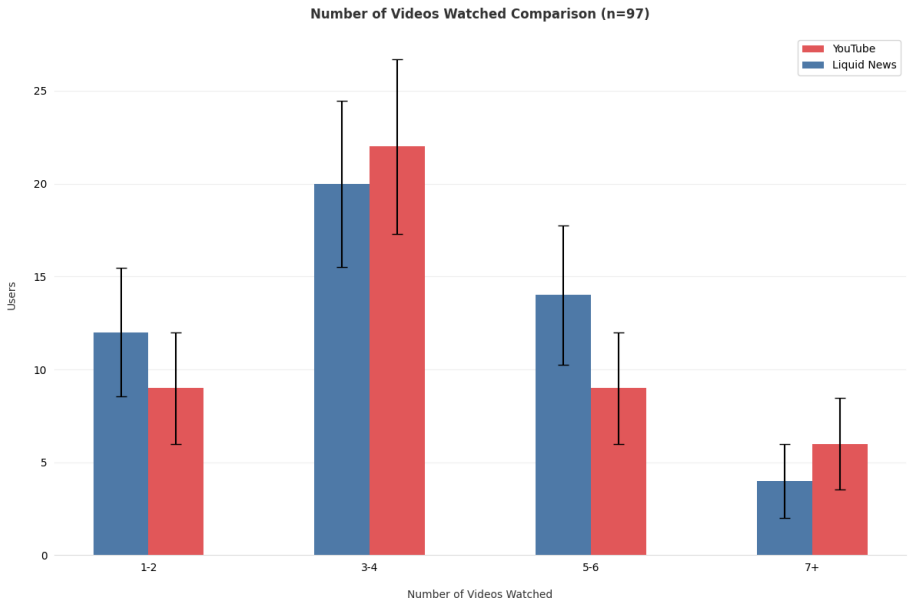

Figure 5-8: Distribution of the number of videos watched by participants who completed the survey. The average number of videos watched by the platform was 3.768 for Liquid News and 3.93 for YouTube.

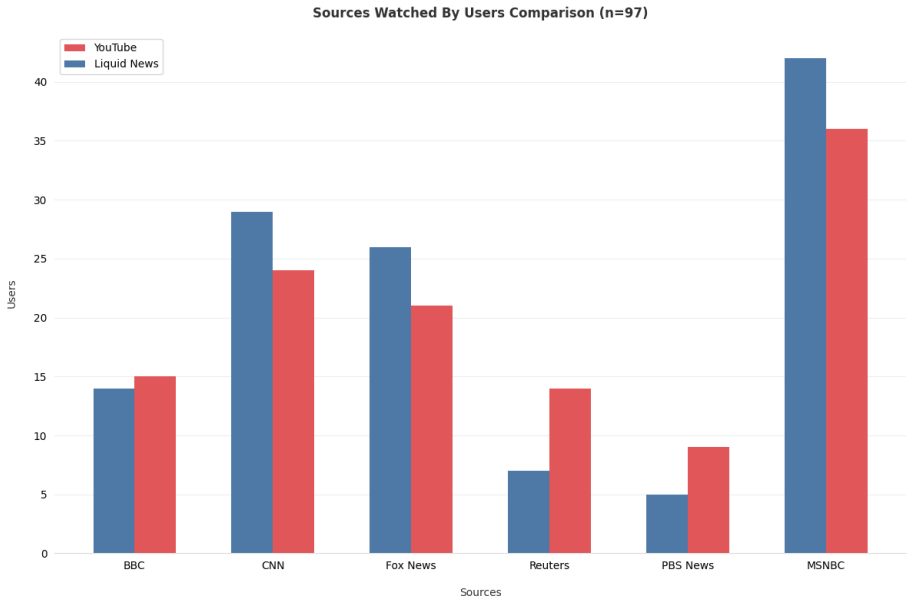

Figure 5-9: Distribution comparison between Liquid News and YouTube on the sources viewed during surveying. The average unique sources viewed for Liquid News and YouTube were 2.517 and 2.630, respectively.

substantial in cases where many users identified different perspectives, such as legislation and gun control. The one exception is the democracy topic, where Liquid News had substantially more users identifying different perspectives.

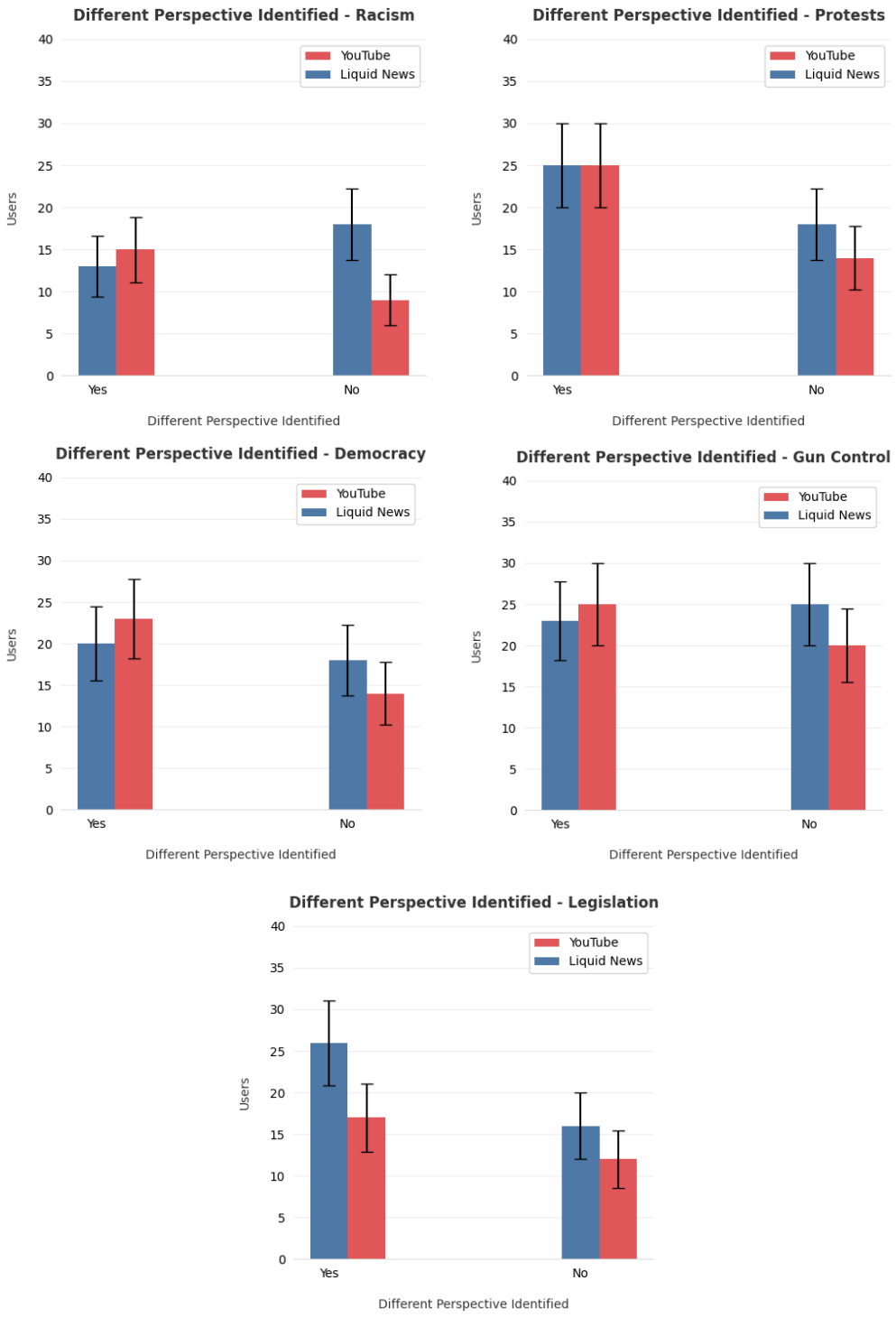

Figure 5-10: Each plot compares the numbers of surveyed users who identified different perspectives on each subtopic.

Chapter 6

Discussion

6.1 RPCSE

As highlighted in 5.1.4, although RPCSE shows slightly lower results compared to SimCSE in STS benchmarks, a closer look at the embedding space and quality of both RPCSE models suggests that RPCSE takes a step in the right direction for creating more meaningful sentence embeddings by decreasing anisotropy as suggested in 5.1.4. A critical next step is to quantitatively support these observations via conducting an alignment vs. uniformity study motivated by work done in [11, 48, 47].

Although RPCSE shows decreased performance on STS benchmarks, there remains room to explore as RPCSE also aimed to improve the meaningfulness of sentence embeddings, similar to whitened models. An essential next step is to evaluate RPCSE embeddings on more fine-grained tasks, such as text classification and sentiment analysis, and compare their performance to that of SimCSE. These additional tests would help determine the robustness of RPCSE, relative placement loss, and neutral pairings in learning embeddings.

Other improvements include exploring newer LLMs, such as GPT-3, GPT-4, and Claude, as the backbone of our RPCSE model. Since neutrals don't have the effect initially hypothesized on STS tasks, giving neutrals scaled negative weights in future iterations could lead to better results.

6.2 Liquid News

Section 5.2.1 show that Liquid News performs reasonably well compared to existing methods regarding segmentation and clustering. The most similar is SECTOR 2.2.1. Liquid News outperforms SECTOR by 8-9 % in isolation at the subtopic and topics level. Furthermore, when analyzed against the joint topic and subtopic clustering, Liquid News achieved 72.1% accuracy, comparable to the 72.3% accuracy achieved by SECTOR in the multi-label noisy data experiment, which closely mirrors the Liquid News test setting. These promising results show that the task-agnostic LLM can compare with more complex task-specific architectures using few-shot learning.

Furthermore, it shows that an unsupervised approach can compete with supervised methods, meaning the overhead of developing and maintaining datasets can be removed. Although the results are promising, better results may have been achieved with more exploration. LLMs with few-shot performance are heavily dependent on the prompting model, and variants of prompts shown in Fig. 4-5 and Fig. 4-6 may yield better results for segmentation and clustering. Additionally, LLMs such as GPT-4 are improving via RHLF, and their ability on these tasks is evolving rapidly. Therefore, the performance and results may improve without a fundamental change in the prompting.

The most apparent direction for future work regarding clustering and segmentation is to explore more prompting patterns to see how variations in wording can influence the outcome of the clustering and segmentation. Furthermore, rather than relying on the most significant cosine similarity, explore if the LLM can identify the clusters and place the videos and segments into their clustering. Additionally, it would be interesting to see how LLMs perform on the same dataset as the SECTOR model.

The results are mixed when applying the clustering and segmentation task to help users parse the news and expose users to more perspectives. As explored in 5.2.2, Liquid News makes the news easier to parse as Liquid News users could identify the important topics concerning the dataset more frequently. However, the advantages were not by as large of a margin as we would have hoped. The major success of Liquid

News was seen in its ability to help users identify the importance of more nuanced or subtle topics, as shown by Fig. 5-6, where Liquid News users outperformed YouTube users in ranking the importance of issues such as democracy, racism, and legislation which were heavily related to the dataset. Thus, in terms of making the news easier to parse, that data support that, to some extent, Liquid News does help users parse the news. Switching focus to the objective of exposing users to more perspectives, Liquid News fails to make substantial improvements. As noted in 5-9, the distribution of sources watched showed no clear emphasis.

Furthermore, the average amount of videos and sources observed was even across both platforms. Additionally, when specifically asked if they identified different perspectives for a given topic, Liquid News did not show an improved ability to identify issues and, in comparison, showed no advantage to YouTube. Overall, this signals that Liquid News failed to achieve its goal of introducing users to more perspectives. The performance results in clustering and segmentation tasks indicate that this failure is due to the user interface rather than the data. The success in helping users parse the news but failure in identifying different perspectives suggests that the part of the system that failed was sorting clips based on semantic relation in a grid. One potential reason for this failure is that although the grid ordering helps group clips together when there are many clips for a given subtopic, many perspectives are only shown if users scroll and explore the grid. Users may not have been scrolling or exploring due to a lack of interest or not realizing the difference in perspectives shown later in the grid. This is one shortcoming of the UI/UX for not making that feature clear, which could have impacted the survey results. This failure presents an obvious direction for future UI testing. Additionally, more designs can be created to represent the data and can be A/B tested to identify a design pattern that best suits the objective.

Another area of improvement that can be explored for Liquid News is the system architecture. Although not an immediate issue in a research context, the Liquid News pipeline takes a substantial amount of time to run. For example, a dataset of 160 videos took approximately 1.5 hours to run. This is a challenge if the system wants to be used for real-time or near-real-time analysis. A heuristic-based updating

technique can be used in which new videos and segments are assigned to the nearest semantically related topic and subtopic based on the previous clustering. When a set threshold is crossed, the pipeline can be rerun. However, this approach will lead to stale data, and new data can be missed during pipeline execution.

Switching to a serverless architecture is one approach to solve the system's speed. The pipeline methods can be refactored to be fine-grained and run on small compute nodes such as AWS-Lambda that operate on unique IPs. Furthermore, the database schema can be reconfigured to resolve atomic read/write issues arising from the parallel workflow of serverless architecture. This approach would circumvent the API limits set by Open-AI from one IP address. Furthermore, greater throughput can be achieved in transcription. The faster system speed would allow more frequent pipeline runs, pushing it closer to near real-time operation. End users would benefit by getting more fresh data akin to social networks.

Lastly, prior work within the Viral Communications group at the MIT Media Lab explored a multi-modal approach to news processing. The SuperGlue news processing pipeline combines text analysis with image analysis. Liquid News focuses strictly on text data, but including image analysis, namely emotion and body language detection, may improve the clustering of news videos and segments.

Bibliography

[1] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado, June 2015. Association for Computational Linguistics.

[2] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland, August 2014. Association for Computational Linguistics.

[3] Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California, June 2016. Association for Computational Linguistics.

[4] Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics.

[5] Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.

[6] Sebastian Arnold, Rudolf Schneider, Philippe Cudré-Mauroux, Felix A. Gers, and Alexander Löser. SECTOR: A Neural Model for Coherent Topic Segmen-

tation and Classification. *Transactions of the Association for Computational Linguistics*, 7:169–184, 2019. Place: Cambridge, MA Publisher: MIT Press.

[7] Michael Barthel, Amy Mitchell, Dorene Asare-Marfo, and Courtney Kennedy. Measuring news consumption in a digital era.

[8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[9] Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[10] L. Chaisorn, Tat-Seng Chua, and Chin-Hui Lee. The segmentation of news video into story units. In *Proceedings. IEEE International Conference on Multimedia and Expo*, pages 73–76, Lausanne, Switzerland, 2002. IEEE.

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020.

[12] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: a deep quadruplet network for person re-identification, 2017.

[13] Yung-Sung Chuang, Rumen Dangovski, Hongyin Luo, Yang Zhang, Shiyu Chang, Marin Soljačić, Shang-Wen Li, Wen-tau Yih, Yoon Kim, and James Glass. Diffcse: Difference-based contrastive learning for sentence embeddings, 2022.

[14] Federal Communications Commission. Cable act of 1984. *Federal Register*, 1984.

[15] United States Congress. Communications act of 1934. *United States Statutes at Large*, 1934.

[16] Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, September 1990.

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

[18] Kawin Ethayarajh. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China, November 2019. Association for Computational Linguistics.

[19] Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple Contrastive Learning of Sentence Embeddings, May 2022. arXiv:2104.08821 [cs].

[20] Gavagai. A brief history of word embeddings | gavagai. https://www.gavagai.io/text-analytics/a-brief-history-of-word-embeddings/, 2023. Accessed 12 May 2023.

[21] Dylan Hadfield-Menell, Ilya Sutskever, Paul Christiano, and Anca Dragan. Constitutional ai: Harmlessness from ai feedback, 2022.

[22] J. A. Hartigan and M. A. Wong. Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979. Publisher: [Wiley, Royal Statistical Society].

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

[24] Raghvendra Kannao and Prithwijit Guha. A system for semantic segmentation of TV news broadcast videos. *Multimedia Tools and Applications*, 79(9):6191–6225, March 2020.

[25] Thomas K Landauer, Peter W. Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25(2-3):259–284, January 1998.

[26] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[27] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Palm: Scaling language modeling with pathways, 2022.

[28] Ziyang Luo. Analyzing the Anisotropy Phenomenon in Transformer-based Masked Language Models.

[29] Ziyang Luo. *Analyzing the Anisotropy Phenomenon in Transformer-based Masked Language Models*. PhD thesis, 2021.

[30] Bill MacCartney and Christopher D. Manning. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 521–528, Manchester, UK, August 2008. Coling 2008 Organizing Committee.

[31] Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA).

[32] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space, September 2013. arXiv:1301.3781 [cs].

[33] Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. Text and Code Embeddings by Contrastive Pre-Training, January 2022. arXiv:2201.10005 [cs].

[34] Nic Newman, Richard Fletcher, David A. L. Levy, and Rasmus Kleis Nielsen. Digital news report 2017. Technical report, Reuters Institute for the Study of Journalism, 2017.

[35] OpenAI. Gpt-4 technical report, 2023.

[36] Dan Pelleg and Andrew W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the 17th International Conference on Machine Learning*, pages 727–734, San Francisco, June 2000.

[37] Gabriel Pereyra, Noam Shazeer, Ciprian Chelba, Niki Parmar, Naman Goyal, Manjunath Kudlur, Patrick Nguyen, Danqi Chen, Ashish Vaswani, Zhifeng Dai, Ankur Bapna, James Bradbury, Yu hsin Chen, Peng Xu, Junjie Li, Jiahui Yu, Denny Zhou, Quoc V. Le, Christopher Olah, and Jakob Uszkoreit. Lamda: Language models for dialog applications, 2022.

[38] Yan Qian, Yi Zhang, Yuxian Wang, Wei Chen, Dong Yu, Jiajun He, and Kai Yu. Conformer: Convolution-augmented transformer for speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7290–7294. IEEE, 2020.

[39] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.

[40] Sara Rajaee and Mohammad Taher Pilehvar. An Isotropy Analysis in the Multilingual BERT Embedding Space. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1309–1316, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[41] Ronald Reagan. Reagan administration repeals fairness doctrine. *The Ronald Reagan Presidential Library and Museum*, 1987.

[42] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084, 2019.

[43] Reuters Institute. Digital News Report 2022. https://reutersinstitute.politics.ox.ac.uk/sites/default/files/2022-06/Digital$_N$*ews* *Report*$_2$022*.pdf; 2022.*

[44] Kristina P. Sinaga and Miin-Shen Yang. Unsupervised K-Means Clustering Algorithm. *IEEE Access*, 8:80716–80727, 2020. Conference Name: IEEE Access.

[45] Lifu Tu, Caiming Xiong, and Yingbo Zhou. Prompt-tuning can be much better than fine-tuning on cross-lingual understanding with multilingual language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 401–411, 2022.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, December 2017. arXiv:1706.03762 [cs].

[47] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere, 2020.

[48] Zhenghao Wang, Jiaxin Li, and Yang Liu. Pre-trained language models for textual similarity measurement. *arXiv preprint arXiv:2004.14072*, 2020.

[49] Yuxian Zhang, Zhenghao Wang, Jiaxin Li, and Zhiyuan Liu. Whiteningbert: An easy unsupervised sentence embedding approach. In *Proceedings of The Web Conference 2020 (WWW '20)*, 2020.

Appendix A

Figures

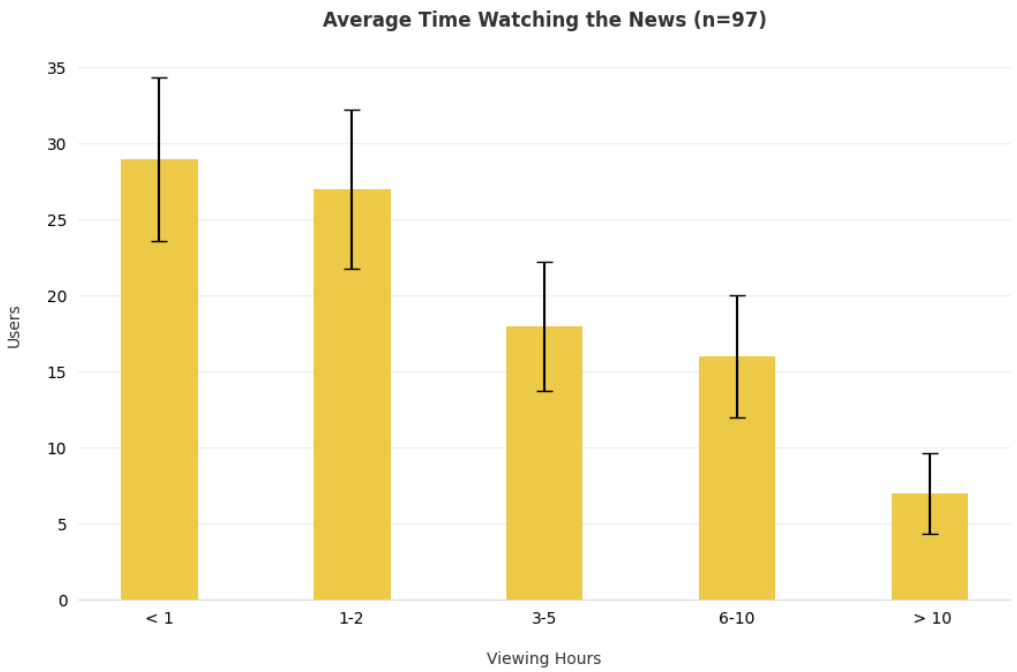

Figure A-1: Distribution of how many hours of news on average users surveyed in 5.2 watched per week.