Leipzig University
Institute of Computer Science
Degree Programme Computer Science, M.Sc.

Unsupervised Frame Identification in Argumentative Discussions

Master's Thesis

Dominik Schwabe

1. Referee: Junior-Prof. Dr. Martin Potthast
2. Referee: Assistant Prof. Dr. Khalid Al-Khatib

Supervisor: M.Sc. Shahbaz Syed

Submission date: October 24, 2023

Declaration

Ich versichere, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann. Ich versichere, dass das elektronische Exemplar mit den gedruckten Exemplaren übereinstimmt.

Leipzig, October 24, 2023

.....
Dominik Schwabe

Abstract

In contemporary society, individuals actively engage in online discussions to express their perspectives on topics. They do so by selecting and highlighting aspects aligned with their interests, knowledge, and beliefs. This practice, termed "framing", plays a crucial role in shaping public opinions and discovering frequently highlighted perspectives, called "frames", has been extensively studied particularly in the news domain. Providing a summary of the diverse frames within discussions not only enhances reader comprehension but may also increases willingness to consider different viewpoints. Existing methods for discovering frames are either limited to a small inventory of generic frames or are constrained by their reliance on models tailored for frame identification within a limited dataset, limiting their applicability to specific domains. In our work, we adopt a modular unsupervised approach that is adaptable to state-of-the-art (SOTA) techniques, focusing on sentence clustering based on language model representations to uncover frames. We leverage the capabilities of general language models, known for their impressive performance across tasks for which they were not explicitly trained. This characteristic proves invaluable for the demanding task of frame generation and assignment, where acquiring high-quality training data is often challenging. Specifically, we explore the application of 19 SOTA Large Language Models (LLMs) for generating issue-specific frames and assigning generic frames to identified frame clusters. Notably, our findings demonstrate that LLMs, when instruction-tuned, excel in frame generation and assignment. Looking ahead, we propose that future research should explore the potential of LLM-generated diverse label options, recognizing their pivotal role in enhancing information comprehension and aiding decision-making processes.

Contents

List of Figures

List of Tables

List of Prompts

List of Instructions

Chapter 1

Introduction

At 23:00 GMT on January 31, 2020, the United Kingdom (UK) formally exited the European Union (EU). Having been a member of the EU since January 1, 1973, the UK voted to exit the EU on June 23, 2016, with a 51.9% majority in favor, marking the highest voter turnout (72%) since the 1992 General Election. The outcome was surprising because most experts had forecasted a vote against leaving.

The formation of opinions is complex and still poorly understood. However, there is a consensus in the scientific community that framing significantly influences the formation of opinions. Entman [32] conceptualizes the framing by giving an elaborated definition:

> **To frame is to select some aspects of a perceived reality and make them more salient in a communicating text, in such a way as to promote a particular problem definition, causal interpretation, moral evaluation, and/or treatment recommendation for the item described.**

In short, framing involves selection and salience. The term "salience" refers to the act of making information more noticeable, meaningful, and memorable to the audience. Increased salience enhances the likelihood that recipients will both notice and extract meaning from information.

To make information more salient within a text, strategic placements, repetitions, and associations can be employed. Information becomes especially salient when it aligns with existing beliefs and relates to the core values of the receiver.

It is worth noting, as highlighted by Entman, that omissions in framing can be just as influential as the inclusions, because a receiver's response is significantly affected when they process one interpretation with limited alternatives.

In the realm of public opinion, Chong and Druckman [25] noted that citizens often exhibit low-quality and frequently unstable opinions, susceptible to

change due to selective information. While framing typically lacks the power to sway the opinion of an individual, it wields significant influence when applied to a group of people. This becomes evident when considering the work of Kahneman and Tversky [44], who demonstrated that the presentation of an objective reality can significantly impact decision-making, even when logically equivalent (e.g., 'the glass is half-full' vs. 'the glass is half-empty'). When elites, such as politicians, media outlets, or interest groups, consistently mold public opinion through framing, it raises fundamental questions about the legitimacy of democratic voting. The potency of these frames often relies on exaggerations or even falsehoods, exploiting public fears and prejudices.

As an example, Khabaz [48] examined what frames elites advocating for leaving the EU frequently used to convince British citizens that voting for leaving the EU is the right decision.

Getting my/our country back While this frame was never adequately defined by politicians who used it, it implies that the UK was besieged by East European migrants, plagued by open borders, subjected to EU diktats and had subsequently lost its national sovereignty.

Undemocratic EU The EU is seen as an "undemocratic super state" without a proper democratic oversight that imposes irrelevant edicts on British people. This frame can be seen as misleading because EU laws must overcome significant hurdles and the British government wields substantial influence in shaping those laws.

Take control of our own destiny Leaving the EU would empower the British government which would be much better in making decisions for British citizens. Immigration takes access to school places, housing and healthcare from British people.

Brexit has had significant economic consequences for the UK, including reduced business investment, increased offshoring by British firms to the EU, and labor shortages due to European workers returning to their home countries[1]. Furthermore, Brexit has negatively impacted British citizens by ending free movement within the EU, resulting in long delivery times and higher prices. A survey indicated that 55% of British citizens would vote to rejoin the EU, while 31% would opt to maintain the current state if a referendum were held again[2].

[1] https://en.wikipedia.org/wiki/Brexit#Impact (accessed: 01.09.2023)

[2] https://docs.cdn.yougov.com/5y66bpmr12/Internal_Brexit_230714.pdf (accessed: 01.09.2023)

Chong and Druckman found factors like knowledge, source credibility, cultural values, individual values and access to alternative information influence how easy frames can change a receivers opinion. Notably, professionals like scientists, lawyers, and economists, who were surveyed before the Brexit referendum, overwhelmingly favored staying in the EU (with 70% to 90% in favor)[3].

In addition to framing, high coverage and repetition of frames over an extended period played a crucial role in anchoring beliefs in the minds of the British public.

While framing can be viewed as a tool for manipulation and deception, it is also essential for understanding and communicating insights from our complex world. As pointed out by Chong and Druckman [25], framing becomes a liability when individuals lack the ability to differentiate among different frames and remain constantly vulnerable to changing representations of issues. It is crucial to note that holding extremely strong attitudes toward an issue can be as detrimental as having no or only weak attitudes. Those with strong attitudes tend to focus solely on frames that reinforce their existing beliefs, while individuals with weaker attitudes can be easily influenced through repeated exposure to specific frames. An ideal citizen possess well-informed opinions, along with the capacity to tolerate alternative perspectives and a willingness to reevaluate their views.

Achieving the ideal of well-informed, open-minded citizenship is challenging due to the complexity of political topics, which demand extensive research for a profound understanding. Consequently, many people rely on media and their framing of political issues to form opinions. However, journalists often fail to present information objectively, primarily because conducting objective research is time-consuming. Instead, they tend to report the frames presented by elites.

Secondly, framing research can be considered unreliable due to researchers' subjective perceived salience of frames, which can't guarantee the influence on audience thinking. According to De Vreese [29], it is essential to differentiate between two types of frames: issue-specific frames (related to specific topics or events) and generic frames (found across different topics). While issue-specific frames offer in-depth insights, they also make results less comparable. To address this, Boydstun et al. [16] have identified 14 generic frames commonly found in news texts. This classification improves frame identification comparability, but it still faces challenges, such as low agreement between annotators, as noted by Card et al. [21].

Establishment of automated systems for frame identification and the de-

[3] https://en.wikipedia.org/wiki/Opinion_polling_for_the_United_Kingdom_European_Union_membership_referendum#Polling_within_professional_groups (accessed: 01.09.2023)

velopment of frame-resistant mechanisms could significantly contribute to addressing these problems. If journalists had access to such systems, they could provide more balanced reporting, presenting multiple interpretations of news events. This approach would directly benefit the audience by offering a more comprehensive view of objective reality, empowering individuals affected by political decisions, and reducing the influence of elites. Furthermore, implementing these systems could help manage the overwhelming volume of unfiltered information spread through social media. This, in turn, would alleviate the problem of echo chambers, where individuals tend to focus on a limited set of interpretations.

Existing methods for building such systems are limited in that they either employ predefined sets of generic frames labels, or are trained on small subsets of news articles about specific geopolitical topics such as "gun violence in the U.S" or "immigration policies". In this work, we aim to develop an unsupervised approach to find issue-specific frames in online discussions, guided by the following research questions:

1. How to computationally model frames?
2. How to discover/generate issue specific frames using Large Language Models (LLMs)?
3. How to qualitatively evaluate the generated frame labels?

Our approach involves clustering interpretations and assigning concise labels to summarize these groups. Through clustering, we aim to balance the salience of interpretations, reducing the dominance of frequently presented interpretations and elevating the significance of those receiving less attention.

By using short labels to present all interpretations, we enable audiences to quickly grasp complex topics, minimizing the need for extensive selection and omission. To achieve this, we primarily rely on Large Language Models (LLMs) due to their exceptional performance in various human tasks. With the introduction of GPT3.5 [67] and ChatGPT [43], this performance has been elevated to a remarkable level, displaying near-perfect understanding of instructions and making human expert knowledge readily accessible to a wide audience at low cost.

1.1 Approach

We define the task of "Unsupervised Frame Identification" as identifying frames in a collection of texts centered on the same topic, without relying on domain-specific extraction methods or supervision.

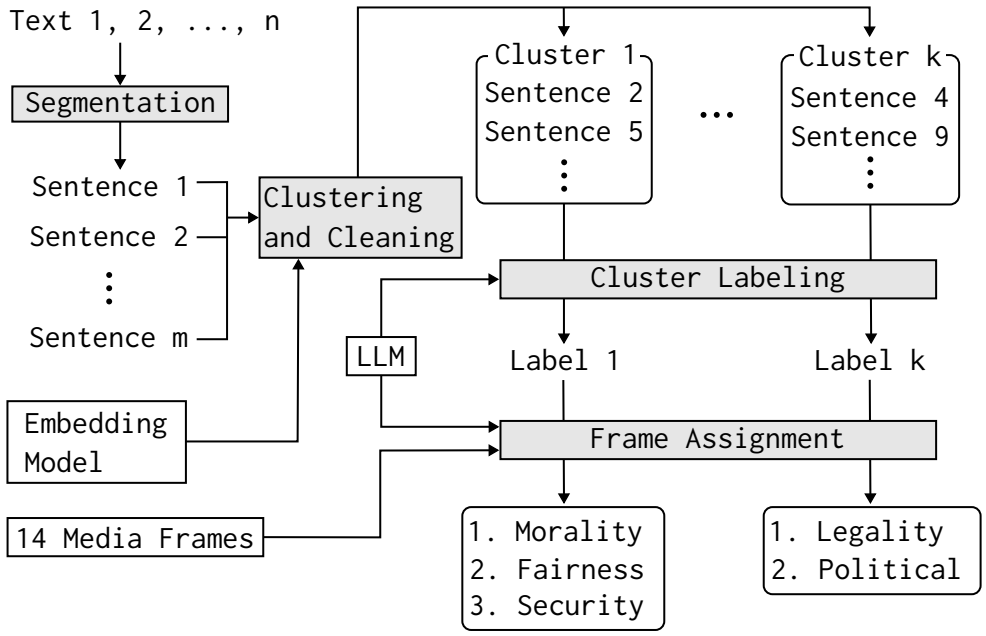


Figure 1.1: Our "Unsupervised Frame Identification" pipeline clusters sentences extracted from texts discussing the same topic, generates labels for the resulting clusters, and assigns up to three out of 14 generic media frames to each of these labels (refer to Appendix B for a list of all 14 media frames). This process leverages the capabilities of LLMs without any fine-tuning.

Figure 1.1 illustrates the pipeline we created to address this task, which takes a collection of texts centered around the same topic as its input. For our experiments, we employ argumentative discussions from ChangeMyView, which offer diverse perspectives on various topics. The rigorous moderation on this platform minimizes noise and keeps discussions focused on the topic.
We splits the texts into sentences, due to the availability of robust embedding models for converting sentences into numerical representations.
Subsequently, we apply clustering to the sentence embeddings to filter out sentences lacking frame information and cluster the remaining embeddings to obtain clusters that represent dominant viewpoints/frames (Chapter 7). To label each cluster, we present its content to a LLM that we instruct to generate a short descriptive label (Chapter 8). We employ the same LLM to assign up to three out of 14 generic media frames to each of the generated labels, ordered by importance (Chapter 9).
Finally, we assess our approach through both human and automatic evaluation, utilizing a dataset of 300 diverse clusters, annotated with one human label and up to two media frames per cluster (Chapter 10).

Chapter 2

Related Work

Frame Classification The framing concept is challenging to define and researches tend to come up with various methods for identifying frames in texts, making it hard to compare results. Boydstun et al. [16] address this issue by introducing 14 generic media frames frequently used in the news domain and release the Policy Frames Codebook which can be used to reliable identify media frames in texts.
Building on this work Card et al. [21] release the Media Frames Corpus, a dataset of several thousand news articles annotated with the 14 media frames. Naderi and Hirst [64] compare LSTM [39], Bi-LSTM [35], GRU [24] models and random forest classifiers using word embeddings, LDA [14] or TF-IDF features for assigning the right frame to sentences from the Media Frames Corpus. They find that LSTM, Bi-LSTM, and GRU models perform the best on this task while the performance differences between these three models is very small.
Liu et al. [57] create the Gun Violence Frame Corpus (GVFC), a collection of news-headlines annotated with 4 generic media frames and 5 issue-specific frames. They show that a fine-tuned BERT model [30] significantly outperforms LSTM, Bi-LSTM, Bi-LSTM with attention [8] and Bi-GRU with attention models on the GVFC and Media Frames Corpus for the task of frame assignment.
Ajjour et al. [1] develop the Webis-Argument-Framing-19 dataset by crawling arguments from debatepedia.org together with labels that describe the topical aspect (frame) of an argument. They find 80% of the labels to be issue-specific and 20% to be generic i.e. occurring in more than one topic. Additionally, they develop a novel frame identification approach by clustering the arguments into topics, removing topic specific words and clustering the resulting reduced arguments into clusters which they identify as frames.
Heinisch and Cimiano [38] examine if training an LSTM or GRU on both

the Webis-Argument-Framing-19 dataset and Media Frames Corpus improves the frame classification performance. They cluster frames from the Webis-Argument-Framing-19 dataset to combine frame labels that are very similar (e.g. sex and sexuality) and focus on text spans from Media Frames Corpus on which at least two annotators agreed. They show that the multi-task training yields substantial improvements over the single-task training on both datasets when using high quality data.
Syed et al. [85] suggest to generate extractive summaries for argumentative discussions by producing a ranking of arguments for each media frames and selecting the top ranked arguments for each frames as a summary. They find that a BM25 [78] model with default settings performs best for ranking arguments by frame-relevance and even outperforms a supervised classifier based on the approach from Heinisch and Cimiano [38].
While most related work focuses on assigning frames from a closed set of classes, we are focused on unbounded generation of issue-specific frames and mapping the generated frames to one of the 14 media frames.

Argument Clustering Boltuzic and Snajder [15] examine different vector space models (bag-of-words and skip-gram word vectors [62]) and a trained semantic textual similarity model for unsupervised prominent argument identification in online discussions via hierarchical agglomerative clustering. They find that averaging the skip-gram word vectors for a sentence's words after removing stopwords and comparing them with the cosine similarity gives the best results for unsupervised argument identification.
Misra, Ecker, and Walker [63] create the Argument Facet Similarity Corpus (AFSC), a dataset consisting of 6000 argument pairs over three topics annotated with a score that expresses the similarity in the facet/frame behind the arguments.
Reimers et al. [77] cluster arguments from their UKP ASPECT Corpus and the AFSC with hierarchical agglomerative clustering. They find TF-IDF to perform worst, InferSent [26] to perform best while there is no clear difference in performance between GloVe [69], ELMo [70] and BERT embeddings on the UKP ASPECT Corpus. Surprisingly TF-IDF vastly outperforms all other approaches and InferSent performs the worst on the AFSC. Fine-tuning a BERT model and testing it on unseen data hugely increases the performance making it similar to human performance.
Daxenberger et al. [28] create ArgumenText, a huge search index for retrieving pro and counter arguments for a given query. They also show three example clusters with human generated labels using the UKP ASPECT Corpus with the fine-tuned BERT model presented by Reimers et al. [77].
Dumani, Wiesenfeldt, and Schenkel [31] create De Argumentenfabriek, a dataset

of arguments annotated with 133 issue-specific frames. Due to the similarity of some frames, they group them into 22 groups. With a logistic regression on top of Sentence-BERT [76] representations they achieve a frame assignment accuracy of 0.68. When additionally fine-tuning the Sentence-BERT model, the accuracy increases to 0.96. They also experiment with clustering on their dataset but can not improve upon the state-of-the-art (SOTA) from Reimers et al. [77].
Färber and Steyer [33] compare the suitability of HDBSCAN [20], k-means, UMAP [58], TF-IDF vectors, and BERT embeddings for the task of clustering arguments by their topical aspect. Surprisingly TF-IDF vectors with HDBSCAN perform best which shows that the masked token pre-training task does not make BERT embeddings suitable for clustering. They also find applying UMAP to BERT embeddings before clustering with HDBSCAN to give huge performance improvements.
Grootendorst [36] present a new approach for creating topic models called BERTopic. They suggest to cluster contextualized word embeddings after applying dimensionality reduction. Similar clusters are then merged based on their TF-IDF vectors until the number of clusters matches the number of requested topics. They show that configuring this simple approach with a Sentence-BERT model, HDBSCAN, and UMAP strongly outperforms traditional topic models.

Argument Generation Schiller, Daxenberger, and Gurevych [81] fine-tune CTRL [47] on a custom dataset to generate arguments based on a topic, stance and aspect. This is the opposite of our task since we want to find aspects/frames for arguments.

Topic Labeling The topic labeling task was originally defined for topic models that define topics as distributions over a vocabulary (e.g. LDA). Therefore most related work for topic labeling focuses on finding labels based on the highest ranked words in word distributions.
Mei, Shen, and Zhai [60] give an overview of the challenges of topic labeling and present an approach based on syntactic features for addressing these challenges. A good label should be understandable to a user, capture the meaning of the topic and distinguish the topic from other topics. They also find phrases to be best suited as topic labels since they are not too general and not too specific.
Carmel, Roitman, and Zwerdling [22] query Wikipedia with the top ranked words from a topic and extract candidate labels from the titles and categories of the returned documents. They rank the candidates using different syntactic features and take the highest ranked candidate as topic label.

Lau et al. [52] generate candidate topic labels by querying Wikipedia with the top-10 topic terms and extracting noun-chunks from the highest ranked Wikipedia articles. The candidates are then re-ranked using various syntactic features and the highest ranked candidate is selected as the topic label.
Hulpus et al. [42] map the top ranked topic words to concepts in an ontology, expand to related concepts, combine the concept graphs into one and find the central concept in this graph which is than use to label the topic.
Aletras and Stevenson [2] create topic label candidates similar to Lau et al. [52]. Additionally, they use the top ranked words from the topic to query Bing[1] and use the title from the search results to build a graph by connecting neighboring words and applying PageRank [68]. The candidates are then ranked by summing the PageRank values for their words and the highest ranked candidate is chosen as a label for the topic.
Kou, Li, and Baldwin [50] map LDA topics and candidate labels to vector space using trigrams vectors, or the sum of word embeddings and compare the topic vectors with the candidate vector using cosine similarity. They find none of these vector space model to be superior but the trigram method performs consistently.
Wan and Wang [90] create summaries as topic labels by iteratively taking a sentence from the document collections that is relevant for the topic, covers as many topic words as possible, is different from sentences that have already been taken, and is different from other topics.
Bhatia, Lau, and Baldwin [10] create the topic labeling approach NETL, which treats Wikipedia document titles as candidate labels and ranks them according to their semantic similarity to the top topic words using word2vec [61], and doc2vec [53]. The top ranked labels are then re-ranked using a support vector regression trained on letter trigrams vectors, PageRank features obtained from Wikipedia hyperlinks and lexical features proposed by Lau et al. [52].
Popa and Rebedea [71] create BART-TL, a topic labeling approach obtained by training a BART model [54] to predict topic labels produced by NETL based on the concatenation of the top-20 topic words. The resulting BART-TL achieves similar results to NETL.
Kozbagarov, Mussabayev, and Mladenovic [51] build topic models by clustering sentences using contextualized word embeddings with k-means and select the sentences closest to the cluster center as the topic/cluster label.
We focus on labeling topic models obtained from BERTopic [36], which produces cluster of sentences instead of word distributions and therefore an entirely different approach for topic labeling is required than presented by previous works.

[1] https://www.bing.com/ (accessed: 01.09.2023)

Chapter 3

Large Language Models

Transformer-based architectures [89] show remarkable performances for natural language tasks. Kaplan et al. [45] illustrated that simply enlarging the model, dataset, and computational resources for training substantially enhances model performance. This has led to the widespread adoption of massively scaled Transformers for language modeling, commonly referred to as Large Language Models (LLMs). LLMs demonstrate remarkable performance in handling novel tasks, without extra fine-tuning. This quality makes them particularly valuable for the task of frame generation and assignment, where obtaining high-quality training data is typically challenging.

3.1 Language Modeling

Language modeling refers to the task of estimating the likelihood of a sequence of tokens within a text. Typically, with LLMs, this modeling is expressed by Equation 3.1, where x is a sequence of tokens, x_n is the n^{th} token in x, and $x_{<n}$ is the sequence of tokens preceding x_n.

$$p(x) = p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | x_{<i}) \tag{3.1}$$

This approach is called autoregressive language modeling, and it involves iteratively predicting the probability of the next token.

3.2 Transformers

The Transformer architecture [89] stands as the SOTA neural network architecture for language modeling tasks, with two key advantages that set it apart from earlier model designs:

- **Parallel Sequence Processing**: It allows to process sequences in parallel within training examples which makes it efficient regardless of the sequence length.
- **Attention Mechanism** The attention mechanism allows it to model dependencies between tokens independent from their distance in the input or output sequences.

This section covers the basics about the Transformer architecture, along with important derived architectures.

3.2.1 Tokenization

To enable a neural network to process text input, the text must be transformed into a numerical representation. This is achieved by breaking the text into discrete units called tokens and converting each token into a vector through dictionary lookup, resulting in a sequence of vectors.
A straightforward tokenization strategy involves splitting the text into individual words. However, this approach has a limitation: words that were not observed during training lack vector representations and must be substituted with an out-of-vocabulary token.
The standard tokenization method in modern text processing is Byte Pair Encoding (BPE), as introduced by Sennrich, Haddow, and Birch [83]. It begins by initializing the vocabulary with all valid characters. Next, the training text is segmented into characters, and the most frequent symbol pair (e.g., 'A' and 'B') is replaced with the merged symbol 'AB'. Subsequently, the vocabulary is updated to include this merged symbol. This process is iteratively repeated until the vocabulary reaches a predefined size.
Using this vocabulary, any valid text can be segmented into tokens without information loss. The resulting token sequence tends to have a comparable length to the sequence obtained through word-tokenization, making it efficient for neural network processing.

3.2.2 Encoder-Decoder Transformer

The original Transformer architecture was introduced by Vaswani et al. [89]. Figure 3.1 shows the Transformer architecture with some details omitted for clarity. The Transformer is a model for sequence-to-sequence tasks, consisting of an encoder that converts a sequence of tokens into a sequence of vectors, and a decoder that converts the vector sequence into a token sequence. Both the encoder and decoder consist of stacks of identical independent encoder layers and decoder layers, respectively.

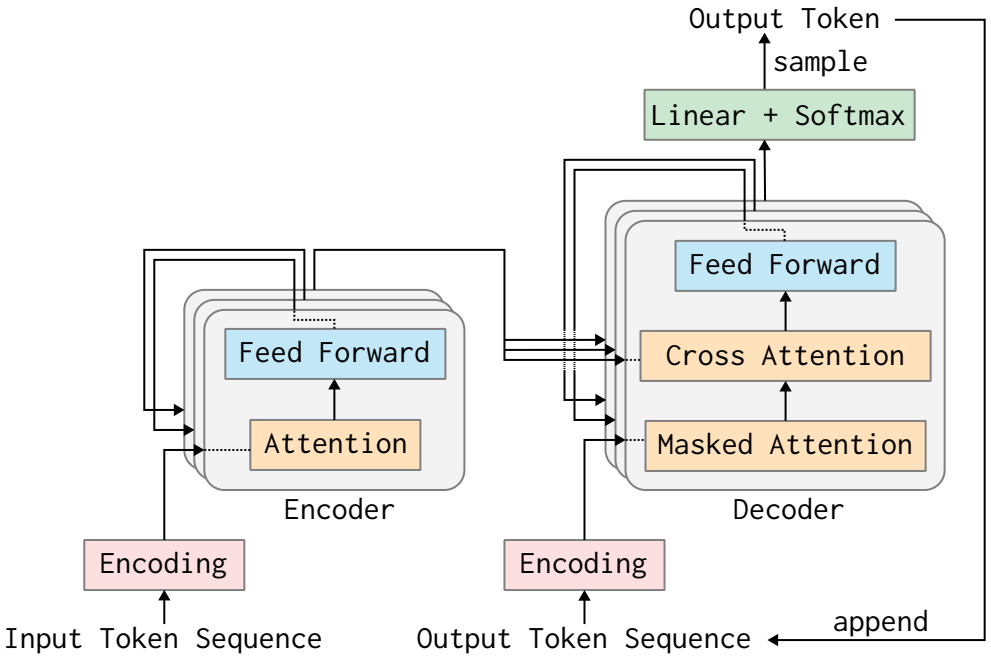


Figure 3.1: The Transformer architecture as introduced by Vaswani et al. [89, p. 3]. Skip connections and normalization layers are omitted for clarity.

To start the process, the "Encoding" component transfers the tokens into vector representations through dictionary lookup. Additionally, it adds positional information to each tokens since the Transformer lacks an inherent mechanism to distinguish token order.
The attention mechanism (Equation 3.2) allows for modeling direct dependencies between all tokens.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.2}$$

The matrices $Q, K \in \mathbb{R}^{d_{\text{model}} \times d_k}$ and $V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ are derived from the input tokens to the attention layer, with $d_{\text{model}}, d_k, d_v$ being hyperparameter for which d_k and d_v are usually derived from d_{model}. The normalization term $\sqrt{d_k}$ helps stabilize the gradient for the softmax during training.
Multi-head attention is a version of attention where h (a hyperparameter) independent attentions are computed by deriving h different (Q, K, V) triplets from the input vectors, applying the attention mechanism and combining the results afterwards. This enables the modeling of different meanings for tokens and finding the right meaning in different contexts. In Figure 3.1 all layers containing "Attention" utilize the multi-head attention mechanism by default.

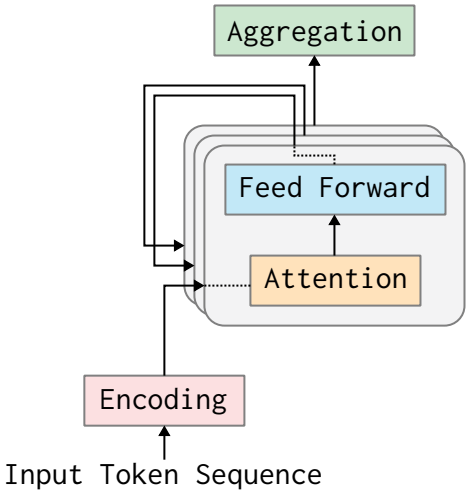


Figure 3.2: A typical encoder-only Transformer architecture as used with the BERT model [30]. Skip connections and normalization layers are omitted for clarity.

The decoder employs an auto-regressive decoding strategy where in each decoding step a new output token is generated based on the previously generated output tokens. While in the encoder, any input token can attend to any other input token, in the decoder, a token can only attend to previously generated tokens. This is restriction is implemented through masked-attention, where entries in QK^T to the right of the diagonal are set to $-\infty$, causing the corresponding positions in the softmax output in Equation 3.2 to be zero.
For cross-attention, the matrices K and V are derived from the encoder outputs, while Q is derived from the output of the previous decoder layer.
To predict the next-token, the embedding of the last input token to the decoder is used to calculate a probability distribution. This distribution is then used to sample the next-token. Typically, the generation process stops either when a special token is generated or after a specific number of output tokens have been generated.

3.2.3 Encoder-only Transformer

The encoder-only Transformer architecture was popularized with the introduction of the BERT (Bidirectional Encoder Representations from Transformers) model [30]. It is primarily employed for sequence-classification tasks. Figure 3.2 illustrates the encoder-only Transformer architecture, which is essentially the same as the encoder-decoder Transformer (see Figure 3.1) but with the decoder component removed.
To aggregate the model outputs into a single vector, common methods include

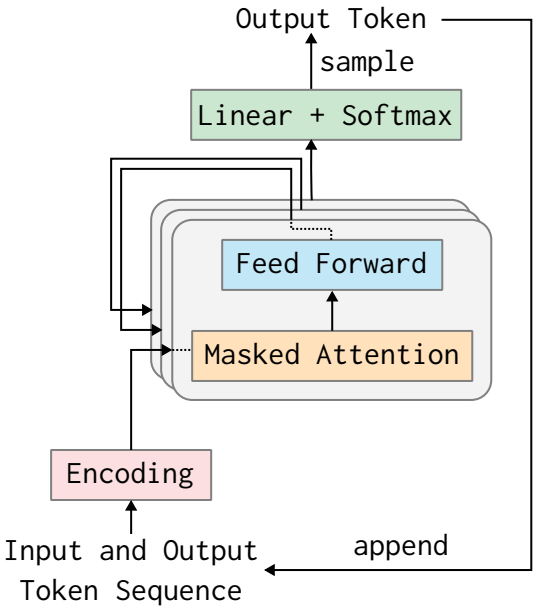


Figure 3.3: A typical decoder-only Transformer architecture as used with GPT models [73, 74, 18]. Skip connections and normalization layers are omitted for clarity.

computing the mean vector of the outputs or prepending a special "[CLS]" token to the sequence and using its output vector as the aggregate. These resulting feature vectors are then employed to train a simple classifier for downstream tasks.
BERT is pre-trained by taking a corpus of natural language text, randomly replacing tokens with the special "[MASK]" token in the input text, and tasking the model with predicting the original token based on the output for the masked token. Additionally, pairs of sentences are presented to BERT, and it determines whether the second sentence follows the first sentence in the original text from which they were extracted.

3.2.4 Decoder-only Transformer

The decoder-only Transformer architecture gained popularity through the GPT (Generative Pre-trained Transformer) model family [73, 74, 18]. It is commonly employed for autoregressive language modeling and utilized in nearly all widely adopted LLMs.
Figure 3.3 illustrates the decoder-only Transformer architecture, which is essentially the same as the encoder-decoder Transformer (Figure 3.1) but with the encoder component removed. Additionally, the cross-attention layer is omitted since there are no encoder tokens to attend to.

GPT models are pre-trained by taking a corpus of natural language text and training the model to maximizing the autoregressive language modeling objective (see Section 3.3).

3.2.5 Sentence-BERT

Sentence-BERT [76] is a training strategy typically applied to encoder-only Transformer models. It aims to produce vector representations for sentences and short texts, ensuring high cosine similarity between similar texts and a cosine similarity near zero for unrelated texts.
In the original paper, a pre-trained BERT model served as the base model for fine-tuned Sentence-BERT models. However, the currently best-performing official model, "`all-mpnet-base-v2`"[1] is based on "`microsoft/mpnet-base`"[2].
The paper outlines three different training strategies for Sentence-BERT models, depending on available annotations. The strategy employed for training "`all-mpnet-base-v2`" and other popular approaches [34] is the contrastive learning objective. For a dataset containing pairs of similar sentences, this objective randomly samples a sentence n from the dataset for each pair (a, p) and seeks to increase the cosine similarity between a and p while reducing the similarity between a and n. An alternative approach for the contrastive objective used with "`all-mpnet-base-v2`" computes the cosine similarity for every possible sentence pair in a batch. It aims to maximize similarity for positive pairs and set it to zero for negative pairs.

3.3 Pre-Training

Pre-training is the initial phase where an untrained language model, learns from a large and varied dataset of natural language texts to maximize its language modeling objective. It's worth noting that nearly all highly capable LLMs utilize a decoder-only Transformer architecture and employ the autoregressive language modeling objective for pre-training.
During the pre-training phase, the model learns to understand general aspects of language like grammar, syntax, facts, and overall language comprehension. Additionally, language models start acquiring the ability to perform tasks such as question answering, machine translation, reading comprehension, and summarization. For instance, Radford et al. [74] achieved leading results on 7 out of 8 language modeling datasets using only pre-training.
After pre-training, the model is a general-purpose language model that serves

[1] `https://www.sbert.net/docs/pretrained_models.html` (accessed: 01.09.2023)
[2] `https://huggingface.co/microsoft/mpnet-base` (accessed: 01.09.2023)

as a strong foundation for further fine-tuning. In the fine-tuning step, the pre-trained model is trained on a labeled dataset, adapting it to specific downstream tasks. This produces better results compared to training an untrained network on the same data.

3.4 Prompting

While fine-tuned models deliver strong performance, acquiring them can be challenging due to the need for high amount of labeled data and extra training.
Interestingly, it has been shown that simply rephrasing a task to resemble the text observed during pre-training without further information (zero-shot) can lead the model to perform well on the task without explicit training [74, 18]. For instance, the sentence "I saw a movie and didn't like it" can be reformulated for sentiment analysis as "I saw a movie and didn't like it. This makes me feel". The model can effectively express sentiment by predicting the next token, such as "bad" or "good".
Furthermore, providing a small number of task demonstrations (few-shot) further boost the model's performance, occasionally even surpassing the SOTA, making efficient use of little data [18].
This is promising because it allows a single pre-trained model to be applied to multiple tasks without requiring extra fine-tuning or task-specific architectures.

3.5 Instruction-Tuning

Instruction-tuning is a technique aimed at aligning the output of pre-trained LLMs with user intentions, ensuring they follow instructions more reliably. Prompting instruction-tuned models is simple compared to prompting a model that only underwent pre-training, because conveying a task to an instruction-tuned model effectively resembles explaining a task to a human.
At the time of conducting our experiments, the most effective instruction-tuning approach involved a two-phase process. Initially, fine-tuning of an LLM is carried out using a dataset of human demonstrations, known as Supervised Fine-Tuning (SFT). Subsequently, additional fine-tuning is performed using Reinforcement Learning from Human Feedback (RLHF). In this step, a reward model, trained on human rankings of the fine-tuned LLM's outputs, is employed to increase the likelihood of aligned outputs.

Chapter 4

HDBSCAN Clustering

Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) is a clustering algorithm developed by Campello, Moulavi, and Sander [20], that produces SOTA clustering. It builts a tree structure by applying a hierarchical density-based clustering algorithm to the data points and extracts a flat clustering from it.
In this chapter, we provide a detailed explanation of this clustering method because a solid understanding of it is crucial to comprehending our approach for uncovering frames in discussions.
We use the Python implementation from McInnes, Healy, and Astels [59] provided via the `hdbscan` package[1]. Parameter names and the description of the HDBSCAN algorithm were taken from this package and its documentation[2].

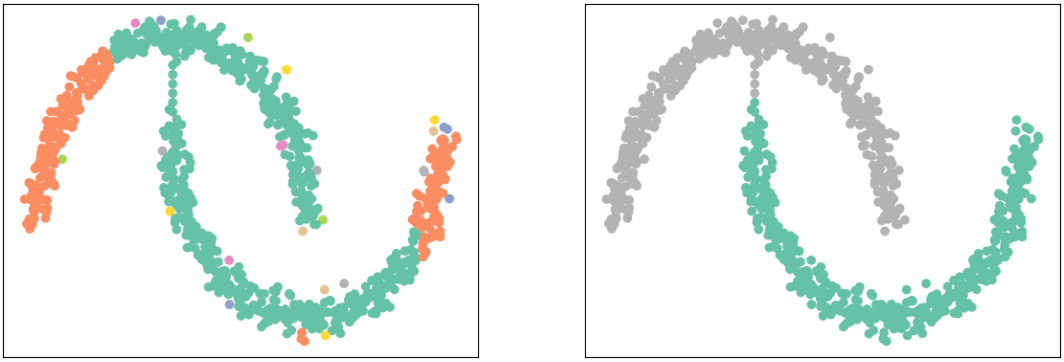
4.1 Robust Single-Linkage

The initial step of the HDBSCAN algorithm involves clustering data points using hierarchical agglomerative clustering with single-linkage, employing the mutual reachability distance.
Hierarchical agglomerative clustering begins by treating each data point as its own cluster, then repeatedly merges the two closest clusters until all data points belong to a single cluster. In single-linkage clustering, the distance between two clusters is determined by the closest pair of elements, one from each cluster.

$$d_{\mathrm{MRD}}(x, y \mid k, d) \coloneqq \max\Big(d\big(x, y\big), d\big(x, \mathrm{nn}_k(x)\big), d\big(y, \mathrm{nn}_k(y)\big)\Big) \tag{4.1}$$

[1] `https://pypi.org/project/hdbscan/` (accessed: 01.09.2023)

[2] `https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html` (accessed: 01.09.2023)



(a) euclidean distance

(b) mutual reachability distance with an euclidean base

Figure 4.1: Applying hierarchical agglomerative clustering with single-linkage to a collection of points that form two interleaving half circles, with a narrow bridge of points connecting the two circles.

The mutual reachability distance (Equation 4.1) is a special distance metric that extends any distance metric to incorporate sensitivity to the local density of the space, making it responsive to both density and noise in the data. Given a distance metric d, two data points x and y, and a parameter k, the mutual reachability distance d_{MRD} is calculated as the maximum among three distances: the distance from x to y, the distance from x to its k-th nearest neighbor (nn_k), and the distance from y to its k-th nearest neighbor. This effectively spreads points in low-density regions apart while preserving distances between points in dense regions.
Figure 4.1 illustrates a sample dataset where two distinct clusters are linked by a narrow bridge of data points. Hierarchical agglomerative clustering with single-linkage and the Euclidean metric produces three large clusters, which is not the desired outcome (Figure 4.1a). However, by employing the mutual reachability distance with the Euclidean distance as the base metric, the clustering process achieves the desired result, separating the two half circles into distinct clusters (Figure 4.1b).

4.2 Condensing the Cluster Tree

In hierarchical agglomerative clustering, the outcome is represented as a tree structure, where each leaf corresponds to a data point, and each internal node represents the merging of two clusters into a new cluster. To identify meaningful clusters, it can be advantageous to view the merging of a smaller cluster into a larger one as an expansion or growth of the larger cluster rather than

the creation of a new cluster.
The parameter `min_cluster_size` is responsible for determining whether a cluster is categorized as either big or small. A cluster is classified as small if its size is less than the specified value for `min_cluster_size`; otherwise, it is considered big. Consequently, there are three possible outcomes for a cluster merge based on the size classification of the clusters being merged:

- **small + small**: The merged cluster needs to be reclassified based on its size as small or big. If it is classified as big, it is considered a cluster creation (birth).
- **small + big**: The big cluster grows and no cluster is created or destroyed.
- **big + big**: A new cluster is created (born) and the two merged clusters are destroyed (death).

The condensed cluster tree is constructed by forming a graph where each big cluster serves as a node, and an edge exists between two nodes/clusters if one of the clusters takes part in a merge to create the other cluster.

4.3 Extract Clusters

HDBSCAN has a `cluster_selection_method` parameter that accepts the values "leaf" and "eom" (Excess of Mass).

leaf Return the leaf nodes within the condensed cluster tree, which are essentially clusters that are at least as big as the specified `min_cluster_size`, and all of their child clusters are smaller than `min_cluster_size`.

eom With this option, HDBSCAN extracts clusters with a high stability, meaning they exists for a long duration throughout the clustering process. Persistence is defined by Equation 4.2 where "distance" represents the distance between two clusters at the moment of their merge.

$$\lambda = 1/\text{distance} \tag{4.2}$$

The value $\lambda_{p,c}$ is the persistence value at which data point p gets added to cluster c (small + big) and $\lambda_{\texttt{death},c}$ is the persistence value when cluster c gets merged with another big cluster to form a new cluster (big + big). The $\lambda_{p,c}$ value for a point p that is already part of cluster c when c is created is equal to the persistence value of the creation of c and the $\lambda_{\texttt{death},c}$ value for the root

cluster (which gets not merged into any cluster) is 0.
Given a cluster c the stability of c is defined by Equation 4.3.

$$\sum_{p \in c} (\lambda_{p,c} - \lambda_{\text{death},c}) \tag{4.3}$$

To find the partitioning into clusters that maximizes stability, begin by visiting and marking all leaf clusters as selected. For each cluster c that hasn't been visited yet but has all its children visited, calculate the stability of c as the sum of its children's stabilities if it is greater than its own stability. If not, mark c as selected and unselect all of its descendants. Finally, when all clusters have been visited, return the selected clusters.

Chapter 5

Content Overlap Measures

We employ the following measures to automatically assess label quality, by comparing the generated labels to a reference label or summary.

5.1 ROUGE

ROUGE [56] measures the similarity between a reference summary R and a candidate summary S, primarily by evaluating the overlap of n-grams, which are contiguous sequences of n words or tokens, between the two texts.
Let $|R|$ be the number of words in R, $|S|$ be the number of words in S, R_n be the set of word n-grams in R, and S_n be the set of word n-grams in S.
ROUGE-N is defined by Equation 5.1, 5.2, and 5.3.

$$\text{Precision}_n = \frac{|R_n \cap S_n|}{|S_n|} \tag{5.1}$$

$$\text{Recall}_n = \frac{|R_n \cap S_n|}{|R_n|} \tag{5.2}$$

$$\text{F-measure}_n = 2 \cdot \frac{\text{Precision}_n \cdot \text{Recall}_n}{\text{Precision}_n + \text{Recall}_n} \tag{5.3}$$

Likewise, ROUGE-LCS is defined by Equation 5.4, 5.5, and 5.6. The LCS function calculates the length of the Longest Common Subsequence (LCS), which represents the longest sequence of words that appear in the same order in both R and S. This measures the ability of the candidate summary to capture the essential information and word order present in the reference summary.

$$\text{Precision}_{\text{lcs}} = \frac{\text{LCS}(R, S)}{|S|} \tag{5.4}$$

$$\text{Recall}_{\text{lcs}} = \frac{\text{LCS}(R, S)}{|R|} \tag{5.5}$$

$$\text{F-measure}_{\text{lcs}} = 2 \cdot \frac{\text{Precision}_{\text{lcs}} \cdot \text{Recall}_{\text{lcs}}}{\text{Precision}_{\text{lcs}} + \text{Recall}_{\text{lcs}}} \tag{5.6}$$

Details We employ the Python `rouge` package[1] to compute the mentioned metrics. It splits the reference and generated summaries into sentences before computing n-grams, which prevents considering n-grams over sentence boundaries.
The ROUGE paper does not specify whether n-grams should be counted multiple times when they repeat in the text (bag-of-words) or only once (set). The `rouge` package implements both versions, but it defaults to using the set version, which we also used for our evaluation.
Although the ROUGE paper only formulates Recall_n for ROUGE-N, the rouge package also implements Precision_n and F-measure_n. In our evaluations, we report the F-measure versions of these metrics.

5.2 BERTScore

BERTScore [94] measures the semantic similarity between a reference summary R and a generated/candidate summary S. This makes the metric robust in capturing paraphrases and significant changes in semantic ordering, as well as distant dependencies.
Let T_R and T_S be the tokenization of R and S respectively (refer to Section 3.2.1) and C_R and C_S be the sequence of contextualized word embeddings for T_R and T_S respectively.
BERTScore utilizes cosine similarity to match each embedding in C_S to an embedding in C_R to compute precision (Equation 5.7) and each embedding in C_R to an embedding in C_S to compute recall (Equation 5.8). Like ROUGE, the scores are combined to derive the F-measure (Equation 5.9).

$$\text{Precision}_{\text{BERT}} = \frac{1}{|T_S|} \sum_{\mathbf{s} \in C_S} \max_{\mathbf{r} \in C_R} \texttt{cosine-sim}(\mathbf{r}, \mathbf{s}) \tag{5.7}$$

$$\text{Recall}_{\text{BERT}} = \frac{1}{|T_R|} \sum_{\mathbf{r} \in C_R} \max_{\mathbf{s} \in C_S} \texttt{cosine-sim}(\mathbf{r}, \mathbf{s}) \tag{5.8}$$

$$\text{F-measure}_{\text{BERT}} = 2 \cdot \frac{\text{Precision}_{\text{BERT}} \cdot \text{Recall}_{\text{BERT}}}{\text{Precision}_{\text{BERT}} + \text{Recall}_{\text{BERT}}} \tag{5.9}$$

[1] `https://pypi.org/project/rouge/` (accessed: 01.09.2023)

The authors used BERT (refer to Section 3.2.3) to obtain contextualized word embeddings but we employ "`microsoft/deberta-xlarge-mnli`"[2] for our evaluations, which is currently the best model according to the official implementation[3].
BERTScore computes scores within the range of -1 and 1. In practice, these scores tend to be close together, which can make them less readable. To address this, we apply baseline rescaling, which involves using an empirically estimated lower bound constant b to spread the values apart. While Equation 5.10 demonstrates the rescaling formula for recall, it is similarly applied to precision and F-measure, each with their own empirically estimated constant b.

$$\text{RescaledRecall}_{\text{BERT}} = \frac{\text{Recall}_{\text{BERT}} - b}{1 - b} \tag{5.10}$$

[2] `https://huggingface.co/microsoft/deberta-xlarge-mnli` (accessed: 01.09.2023)
[3] `https://github.com/Tiiiger/bert_score` (accessed: 01.09.2023)

Chapter 6

Prompt Engineering

In this chapter we present a list of LLMs that were considered the most capable at the time of conducting our experiments, together with prompts that we engineered by gathering recommendations from the research community.

6.1 Examined Large Language Models

We made our first experiments in early 2023, when most SOTA models struggled to effectively follow instructions. Subsequently, with the widespread availability of ChatGPT [43] and its API [17], numerous capable LLMs emerged, prompting us to conduct additional experiments using these models. Therefore, we logically divided this section into two segments: "Pre-ChatGPT Models" and "Post-ChatGPT Models".

6.1.1 Pre-ChatGPT Models

The LLMs we feature in this section were chosen from those evaluated by Liang et al. [55]. Their evaluation encompassed a wide range of LLMs, including those with open, limited, and closed access. In our experiments, we focused on models with open access and popular limited-access models. We selected models based on the criterion of not being surpassed by newer versions, and we opted for simpler models when resource constraints required it. For instance, we opted for T0 over T5, GPT-NeoX over GPT-J, and OPT-66B instead of OPT-175B.

GPT-3

GPT-3 [18] is a LLM with 175 billion parameters. The study conducted by Kaplan et al. [45] observed consistent improvements in language modeling

performance with increased compute, dataset size, and model size. GPT-3's performance aligns with these observations, showcasing robust zero-shot and few-shot capabilities across various NLP datasets without the need for additional fine-tuning.

T0

T0 [79] is an encoder-decoder model that was pre-trained and subsequently fine-tuned on a large set of supervised datasets transformed into a natural language format [7]. They hypothesize that pre-trained LLMs like GPT-3 generalize to new tasks because these tasks are implicitly and explicitly embedded in the training data. Notably, T0 generalizes better to held-out tasks than GPT-3 while being 16 times smaller.

InstructGPT

InstructGPT [67] was developed to align the output of pre-trained LLMs with user intentions. The primary goal is to ensure that the model provides helpful, honest, and safe responses in accordance with user instructions. This alignment is achieved through a two-step process. First, GPT-3 undergoes fine-tuning using a dataset of human demonstrations, referred to as Supervised Fine-Tuning (SFT). After this initial phase, additional fine-tuning is conducted using Reinforcement Learning from Human Feedback (RLHF). In this step, a reward model is employed, which has been trained based on human rankings of GPT-3 outputs.
The final model outputs are significantly more preferable over GPT-3's outputs. Interestingly, even outputs from a smaller InstructGPT model with 1.3 billion parameters are preferred over GPT-3 outputs.

OPT

OPT [93] represents a set of eight LLMs with parameter sizes spanning from 125 million to 175 billion. What sets OPT apart is that its architecture, training procedure, and parameters are entirely open and accessible to the broader research community. This accessibility contrasts with many other capable LLMs, which are typically restricted behind commercial APIs, limiting their study and impact analysis by researchers.

GPT-NeoX-20B

Like OPT, GPT-NeoX-20B [13] is an openly accessible LLM aimed at facilitating the examination of LLMs impacts. Its architecture mostly follows the

architecture of GPT-3 while incorporating minor improvements and using a much lower parameter count, comprising 20 billion parameters.

BLOOM

BLOOM [80] is a LLM developed by BigScience, a collaborative research effort involving approximately 1200 participants representing diverse fields from 38 countries. With 176 billion parameters, BLOOM has a similar size to GPT-3 while being fully accessible to the research community. Furthermore, it was trained on a multilingual corpus including many different programming languages.

6.1.2 Post-ChatGPT Models

The models featured in this section were either developed after ChatGPT became available through its API, with many of them incorporating ChatGPT output in their fine-tuning process, or they were established prior to this API release but got significant attention and recognition due to being popular choices for fine-tuning on ChatGPT outputs. The majority of these LLMs excel at adhering to instructions by implementing various forms of instruction-tuning.
Our selection of models was guided by their reported performance on the HuggingFace Open LLM Leaderboard [9].

ChatGPT

ChatGPT [43], which succeeds InstructGPT, underwent training in a similar fashion but with a focus on engaging in dialogues instead of solely responding to single instructions. The data collection process also received some modifications. The training process for ChatGPT starts with SFT, using dialogues in which a human participant takes on both the user and assistant roles. During these interactions, the model generates suggestions that are used by the participants to compose their responses. The process of training the reward model with RLHF remains largely unchanged. Both the SFT and RLHF steps are iterated multiple times to refine and arrive at the final model.

GPT-4

GPT-4 [66, 19] represents the next iteration after ChatGPT. This model demonstrates human-level performance and represents a significant advancement over its predecessors. GPT-4 exhibits a wide range of capabilities, including proficiency in natural language understanding, text generation, text

manipulation, and translation across various tones, styles, and domains. It also accepts image data as input, expanding its versatility.
Despite its impressive capabilities, GPT-4 is not without limitations. These include tendencies for hallucinations, a lack of knowledge about events occurring after September 2021, overconfidence in incorrect predictions, and the potential for generating biased outputs.
Important technical details of GPT-4, such as its architecture, hardware and compute resources used for training, dataset construction, and training methods, are not publicly disclosed due to competitive considerations and safety concerns.

LLaMA

Hoffmann et al. [40] show that when increasing model parameters, the number of training tokens should be proportionally scaled to achieve optimal performance. LLaMA (Large Language Model Meta AI) [88] comprises a range of models with parameter sizes ranging from 7 billion to 65 billion. Interestingly, these models were trained on a much larger amount of tokens than recommended by Hoffmann et al. [40]. Remarkably, LLaMA-13B already outperforms GPT-3 on most benchmarks despite being ten times smaller in terms of parameter count. This reduction in size enables these models to fit onto a single GPU and results in shorter inference times, making highly capable LLMs accessible to researchers with limited hardware resources.

Pythia

Pythia [12] is a series of 16 LLMs with parameter sizes spanning from 70 million to 12 billion. A notable aspect of Pythia is the release of multiple checkpoints for each model, captured at various stages of the training process. This release enables researchers to examine how LLMs behave during their training across different scales, providing valuable insights into their development and capabilities.

OASST

OpenAssistant [49] represents an initiative aimed at replicating ChatGPT. This effort involves the creation of OpenAssistant Conversations, a multilingual corpus of human-generated and human-annotated assistant-style conversations. Leveraging their dataset, the OpenAssistant project releases versions of LLaMA and Pythia models that have undergone instruction-tuning. This process enhances the models' ability to follow user instructions effectively, mak-

ing them more proficient in generating helpful and contextually relevant responses in conversational interactions.

Alpaca

Wang et al. [91] present a straightforward method for replicating InstructGPT without requiring data collection from human annotators. They begin by gathering 175 human-written task instructions and then task the model with generating additional instructions by presenting a subset of the existing instructions. Instructions are only accepted if they sufficiently differ from already generated instructions.
Subsequently, they employ an existing instruction-following model to generate responses for the collected instructions. This dataset is then utilized for fine-tuning a LLM. Taori et al. [87] used this approach to collect a dataset from ChatGPT and fine-tune LLaMA-7B and LLaMA-13B, resulting in Alpaca-7B and Alpaca-13B, respectively. It is noteworthy that only the model weights for Alpaca-7B have been publicly released.

Vicuna

Vicuna [23] is a LLM that is acquired through fine-tuning LLaMA using conversations sourced from ShareGPT[1], a platform designed for sharing conversations with ChatGPT.
The Vicuna project provides two variants: Vicuna-7B and Vicuna-13B, both obtained from fine-tuning LLaMA-7B and LLaMA-13B, respectively. Notably, Vicuna-13B achieves 90% of the quality exhibited by ChatGPT and significantly outperforms Alpaca-13B.

Baize

Xu et al. [92] construct a dataset of conversations by tricking ChatGPT into deviating from its initial instruction and instead generating fictional dialogues between a user and an assistant. They use questions from Quora[2] as starting point for a conversation and fine-tune LLaMA with the generated dataset. The fine-tuned models are then employed to generate multiple answers for the Quora questions, with ChatGPT selecting the best answer. These answers are subsequently used for further fine-tuning of the model. The Baize project provides two model variants: Baize-7B and Baize-13B, which were obtained by applying this approach to LLaMA-7B and LLaMA-13B, respectively. Baize-13B achieves performance matching that of Vicuna-13B.

[1] https://sharegpt.com/ (accessed: 01.09.2023)
[2] https://www.quora.com/ (accessed: 01.09.2023)

Falcon-40B

Falcon-40B [4] is a fully open model that underwent training in a manner similar to GPT-3. It is engineered for both performance and efficiency, accessible to everyone, and can be utilized for commercial purposes.
Furthermore, there is a variant known as Falcon-40B-Instruct, which represents a fine-tuned version of Falcon-40B. This fine-tuning process employs the approach described in Section 6.1.2. It is noteworthy that at the time of our experiments, Falcon-40B-Instruct was the best-performing open LLM according to the HuggingFace Open LLM Leaderboard [9].

LLaMA-30B-SuperCOT

LLaMA-30B-SuperCOT [6] is a version of LLaMA-30B that was mainly fine-tuned on Alpaca-CoT [72], to enhance the model's instruction-following capabilities.

6.2 Prompt Optimization

For each included model, we craft a prompt based on diverse recommendations with the goal of optimizing the model's performance.

6.2.1 Prompts

We develop different prompts that can be used generically for different tasks. Every prompt contains an "instruction" placeholder and an "input" placeholder. By substituting the "instruction" placeholder with a specific task description, and the "input" placeholder with a relevant example, we form distinct task instances. These tasks are solved by presenting the prompt to the model, and the model's output serves as the solution for the respective task instance.
We design prompts for each model that we find to be optimal according to various online sources.

Default We use Prompt 6.1 for all models for which we could not find a recommended prompt. It is inspired from the recommended prompt for Vicuna[3] and we chose to use it for other models due to its simplicity.

```
{instruction}

USER: {input}

ASSISTANT:
```

Prompt 6.1: Default prompt for all models for which we could not find a recommended prompt.

GPT3.5 We designed Prompt 6.2 especially for GPT3.5 using the best practice guide for prompt engineering from OpenAI [84].

```
{instruction}

Input: """{input}"""

Answer:
```

Prompt 6.2: Specifically designed prompt for GPT3.5 using official recommendations.

[3] https://github.com/lm-sys/FastChat/blob/4e2c942b8d785eb5e2aef1d0df2150e756f381ab/fastchat/conversation.py#L326 (accessed: 21.07.2023)

Alpaca We took Prompt 6.3 from the official GitHub repository for the Alpaca project [87][4]. LLaMA-30B-SuperCoT uses the same prompt according to the official HuggingFace repository[5].

```
Below is an instruction that describes a task, paired with an input that
↪ provides further context. Write a response that appropriately
↪ completes the request.

### Instruction:
{instruction}

### Input:
{input}

### Response:
```

Prompt 6.3: The recommended prompt for the Alpaca and LLaMA-30B-SuperCoT models

T0 With Prompt 6.4 we designed a simple prompt that goes well with the instructions that we designed for specific tasks in later chapters, because we use the word "input" with all our instructions to refer to the input.

```
{instruction}

Input: """{input}"""
```

Prompt 6.4: Prompt tailored for T0 to present the important information in a simple way.

OpenAssistant There are multiple prompt formats defined in the official guide[6]. With Prompt 6.5 we used the prompt that worked best for us.

[4] https://github.com/tatsu-lab/stanford_alpaca#data-release (accessed: 21.07.2023)

[5] https://huggingface.co/ausboss/llama-30b-supercot#prompting (accessed: 21.07.2023)

[6] https://github.com/LAION-AI/Open-Assistant/blob/c2f444d0d85f81db2f9d35507831865b45d40d6c/model/MESSAGE_AND_TOKEN_FORMAT.md#message-format (accessed: 21.07.2023)

```
<|system|>{instruction}<|endoftext|><|prompter|>{input}<|endoftext|>
<|assistant|>
```

Prompt 6.5: The recommended prompt for the OpenAssistant models that worked best for us.

OpenAI-Chat The models from OpenAI are only accessible via an API. We can not pass a prompt directly to ChatGPT and GPT-4 but instead we need to pass a list of JSON objects that contains the instruction to the model together with a user input[7]. Prompt 6.6 shows the list of JSON objects that we pass to the model.

```
[
  { "role": "system", "content": "{instruction}" },
  { "role": "user", "content": "{input}" }
]
```

Prompt 6.6: The prompt developed for OpenAI chat models.

Baize We developed Prompt 6.7 based on the official GitHub repository for Baize[8]. During the time of our experiments, there were no clear guidelines available on how to construct prompts effectively for Baize models. However, the authors later included a note on their official HuggingFace Repository for Baize[9] specifying that the model should utilize the following instruction to achieve optimal results.

```
The following is a conversation between a human and an AI assistant
↪ named Baize (named after a mythical creature in Chinese folklore).
↪ Baize is an open-source AI assistant developed by UCSD and Sun
↪ Yat-Sen University. The human and the AI assistant take turns
↪ chatting. Human statements start with [|Human|] and AI assistant
↪ statements start with [|AI|]. The AI assistant always provides
↪ responses in as much detail as possible, and in Markdown format. The
↪ AI assistant always declines to engage with topics, questions and
↪ instructions related to unethical, controversial, or sensitive
↪ issues. Complete the transcript in exactly that format.
[|Human|]Hello!
[|AI|]Hi!
```

[7] https://platform.openai.com/docs/guides/gpt/chat-completions-api (accessed: 21.07.2023)

[8] https://github.com/project-baize/baize-chatbot (accessed: 21.07.2023)

[9] https://huggingface.co/project-baize/baize-v2-13b (accessed: 21.07.2023)

Consequently, our findings with these models may be slightly less favorable compared to using the optimal instruction.

```
{instruction}
[|Human|]{input}
[|AI|]
```

Prompt 6.7: Initially we found this prompt to be the recommended way of prompting Baize. After we completed all of our experiments we found a different prompt that gives optimal results according to the authors of Baize.

6.2.2 Instruction Types

Models trained using instruction-tuning are capable of comprehending instructions and executing tasks accordingly, whereas models that underwent solely pre-training lack direct understanding of instructions and can only contextually complete scenarios. Consequently, we classify instructions into two categories: "direct" for models with instruction-following capabilities and "dialogue" for models that exclusively underwent pre-training. A "direct" instruction explicitly conveys the task for the model to perform based on the input. Conversely, a "dialogue" instruction characterizes the ongoing scenario, typically involving a user and an assistant, and outlines the expected behavior of the assistant.
We configure instruction types with our prompts in the following manner:

dialogue:

- Default
- Baize

direct:

- Alpaca
- GPT3.5
- OpenAI-Chat
- OpenAssistant
- T0

6.2.3 Configuration

Table 6.1 presents a comprehensive list of all models that utilized in our experiments, along with the corresponding prompts we configured for each model as outlined in Section 6.2.1. The "key" column denotes the name we use in this thesis to reference each respective model.
Figure 6.1 provides a visual representation of the process involved in configuring the models for their respective tasks, as detailed in this chapter.

Table 6.1: Overview over all models that we used in our experiments. The "key" column contains the model identifiers used throughout this thesis. The "source" column displays either the HuggingFace model identifier (accessible at `https://huggingface.co/<model source>`) or the OpenAI identifier[10] (designated by the "OpenAI": prefix). The "prompt" column contains the prompt names, as detailed in Section 6.2.1, paired with their respective models.

Key	Model Source	Prompt
Alpaca-7B	**tatsu-lab/alpaca-7b-wdiff**	Alpaca
BLOOM	**bigscience/bloom-petals**	Default
Baize-13B	**project-baize/baize-v2-13b**	Baize
Baize-7B	**project-baize/baize-v2-7b**	Baize
Falcon-40B	**tiiuae/falcon-40b**	Default
Falcon-40B-I	**tiiuae/falcon-40b-instruct**	Default
ChatGPT	OpenAI: **gpt-3.5-turbo**	OpenAI-Chat
GPT-4	OpenAI: **gpt-4**	OpenAI-Chat
GPT-NeoX	**EleutherAI/gpt-neox-20b**	Default
LLaMA-30B	**huggyllama/llama-30b**	Default
LLaMA-CoT	**ausboss/llama-30b-supercot**	Alpaca
LLaMA-65B	**huggyllama/llama-65b**	Default
OASST	**Yhyu13/oasst-rlhf-2-llama-30b-7k-steps-hf**	OpenAssistant
OPT-66B	**facebook/opt-66b**	Default
Pythia	**OpenAssistant/pythia-12b-sft-v8-7k-steps**	OpenAssistant
T0++	**bigscience/T0pp**	T0
GPT3.5	OpenAI: **text-davinic-003**	GPT3.5
Vicuna-13B	**lmsys/vicuna-13b-delta-v1.1**	Default
Vicuna-7B	**lmsys/vicuna-7b-delta-v1.1**	Default

[10] `https://platform.openai.com/docs/models` (accessed: 21.07.2023)

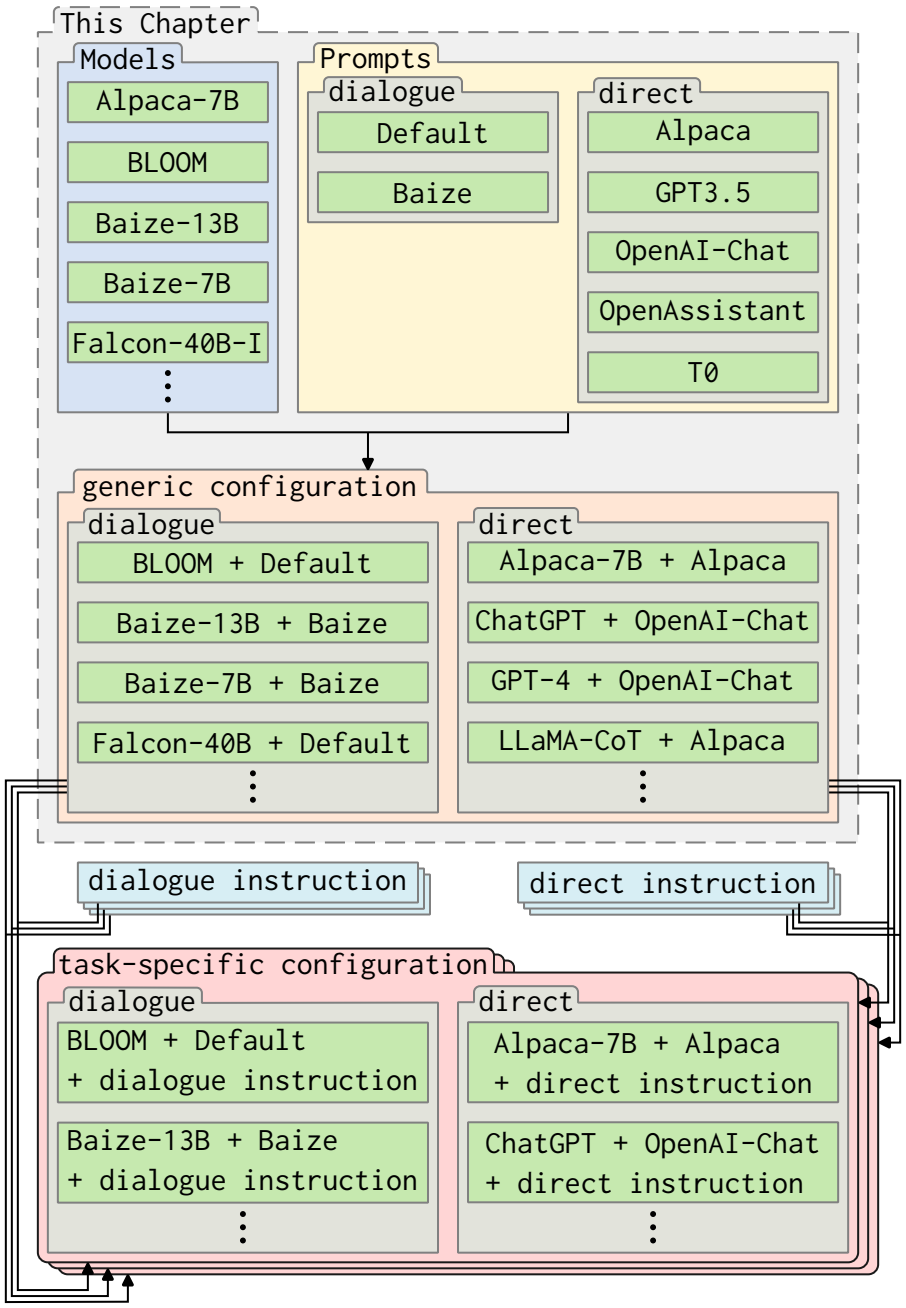


Figure 6.1: This schema illustrates the process of combining the models selected for our experiments with prompts to create a generic configuration. The diagram within the box labeled "This Chapter" depicts the concepts explained in this chapter. Subsequently, we integrate the generic configuration with instructions developed based on the instruction type in later sections to obtain the task-specific configuration.

Chapter 7

Data and Preprocessing

In this chapter, we outline the preprocessing steps applied to our data source of argumentative discussions.
We introduce an approach to eliminate sentences that lack meaningful content related to the discussion topic, referred to as "meta sentence removal". Afterward, within this cleaned discussion, we identify frame clusters by applying our sentence clustering procedure, treating the resulting clusters as issue-specific frames. Due to the meta sentence removal procedure depending on the sentence clustering procedure, we will first explain the sentence clustering procedure and afterwards the meta sentence removal procedure.

7.1 Dataset

For our experiments, we use argumentative discussions because they involve participants with diverse perspectives, giving a wide range of viewpoints on various topics. This diversity makes them particularly suitable for frame identification.
More precisely, we use the ChangeMyView dataset from Tan et al. [86] as our source for argumentative discussions. The ChangeMyView subreddit is known for its robust moderation, which helps maintain a relatively noise-free environment and ensures that arguments are centered on the topic.
The dataset contains 25,043 discussions held between 2013 and 2016.

Preprocessing Comments are in an HTML format. We remove `blockquote` tags as they redundantly cite text from previous comments. We divide the text into sections by utilizing the `p`, `ul`, `ol`, and `li` tags. Subsequently, we break down these sections into sentences using spaCy [41]. Some comments serve as moderations, providing explanations for actions such as comment removal. We identify such comments by various criteria, including whether they are

"stickied" (pinned to the top of the discussion), authored by "DeltaBot" or if they contain one of the following text patterns.

- hello, users of cmv! this is a footnote from your moderators
- comment has been remove
- comment has been automatically removed
- if you would like to appeal, please message the moderators by clicking this link
- this comment has been overwritten by an open source script to protect
- then simply click on your username on reddit, go to the comments tab, scroll down as far as possible (hint:use res), and hit the new overwrite button at the top
- reply to their comment with the delta symbol

If any of these criteria are met, we remove both the identified comments and their associated replies.
Deleted comments, while still part of the discussion, have their text made invisible. We identify these comments by searching for specific text patterns, and once identified, we empty their content:

- `[deleted]`
- `[removed]`
- `[Wiki][Code][/r/DeltaBot]`
- `[History]`

We further convert all text to ASCII and remove control characters.

7.2 Sentence Clustering

Our sentence clustering method draws inspiration from BERTopic [36]. We also considered alternatives like Top2Vec [5] and CTM [11]. However, Top2Vec relies on basic word2vec and doc2vec models without utilizing contextualized word embeddings, and CTM requires training a small neural network for each clustering task while achieving a similar performance to BERTopic. BERTopic overcomes these limitations and stands out as a simple solution, with all its components being interchangeable with the latest SOTA techniques, ensuring future scalability.
See Figure 7.2 for an example of sentence clustering .

7.2.1 Sentence Embedding with Sentence-BERT

To enable clustering, the text must be converted into a numerical representation. We employ Sentence-BERT (refer to Section 3.2.5) due to its SOTA performance with the "`all-mpnet-base-v2`" model, which is the best performing official model according to the documentation[1].

7.2.2 Dimensionality Reduction with UMAP

In high dimensions, the distance to the nearest data point approaches that of the farthest data point. Consequently, in high-dimensional space, the concept of spatial locality becomes unclear, and distance measurements barely change, a phenomenon known as the curse of dimensionality. Dimensionality reduction, as demonstrated by Färber and Steyer [33] and Allaoui, Kherfi, and Cheriet [3], helps mitigate the curse of dimensionality.
We opted to employ Uniform Manifold Approximation and Projection (UMAP) [58] dimensionality reduction, inspired by BERTopic, due to its exceptional ability to retain both local and global features of high-dimensional data in lower dimensions, surpassing other commonly used dimensionality reduction methods.
In the upcoming section, we will clarify the crucial parameters of UMAP and detail the process by which we arrived at our parameter choices.

Parameters

`metric` This parameter determines how to measure the distance between input samples. We set this to "cosine" since it aligns with the inherent metric for Sentence-BERT embeddings.

`output_metric` The output metric determines how distances are measured in the target space. We opt for the "euclidean" because it simplifies the visualization of the embedding space.

`n_neighbors` According to official recommendations[2] it is better to use larger values when using UMAP for clustering. Smaller values makes the approach focus on local structures, which is very sensitive to noise. Conversely, larger values preserve more of the global structure, thus retaining essential data patterns. Thus, we adhere to the official recommendations and raise this value from its default of 15 to 30.

[1] `https://www.sbert.net/docs/pretrained_models.html` (accessed: 01.09.2023)

[2] `https://umap-learn.readthedocs.io/en/latest/clustering.html` (accessed: 01.09.2023)

n_components This parameter controls the number of dimensions in the embedding space. It is crucial to strike a balance, setting it low enough to avoid the curse of dimensionality but high enough to capture important details. We empirically found values higher than 3 are sufficient, with less importance on the upper bound. Therefore, we set this value to 10.

min_dist This parameter defines how tightly points can be packed together. To enhance sensitivity to dense regions, we opt for the minimum value of 0, allowing for very tight packing.

7.2.3 Clustering with HDBSCAN

We utilize HDBSCAN due to its SOTA density-based clustering performance. In the next section, we will elaborate on the essential parameters of HDBSCAN and clarify the process employed to make our parameter choices (for detailed information on HDBSCAN, refer to Chapter 4).

Parameters

metric This parameter corresponds to the distance metric parameter d in Equation 4.1. We select "euclidean" because the input vectors originated from UMAP dimensionality reduction, which embedded them into Euclidean space (refer to Section 7.2.2 `output_metric`).

min_cluster_size This parameter determines the minimum number of samples required for a cluster to be recognized as a meaningful cluster (refer to Section 4.2).
Choosing the appropriate value for this parameter is crucial. If it is too low, distinct clusters may fragment into smaller ones, and if it's too high, smaller clusters may go undetected. There isn't a universal value since the size of a subtopics of a discussion increases when a discussion grows. Consequently, we determine `min_cluster_size` using a regression model that correlates it with the number of sentences in a discussion.
We collected a sample of 50 discussions randomly and another 50 discussions stratified by discussion length from the pool of discussions containing at least 20 comments. Stratified sampling involved evenly dividing the interval $[20, 900]$ into 10 subintervals and randomly selecting 5 discussions from each subinterval whose number of comments lie in this interval. For these 100 discussions, we conducted clustering experiments using various values for `min_cluster_size`, while keeping all other parameters at their optimal settings. Through manual analysis, we determine a lower and upper bound for `min_cluster_size` where

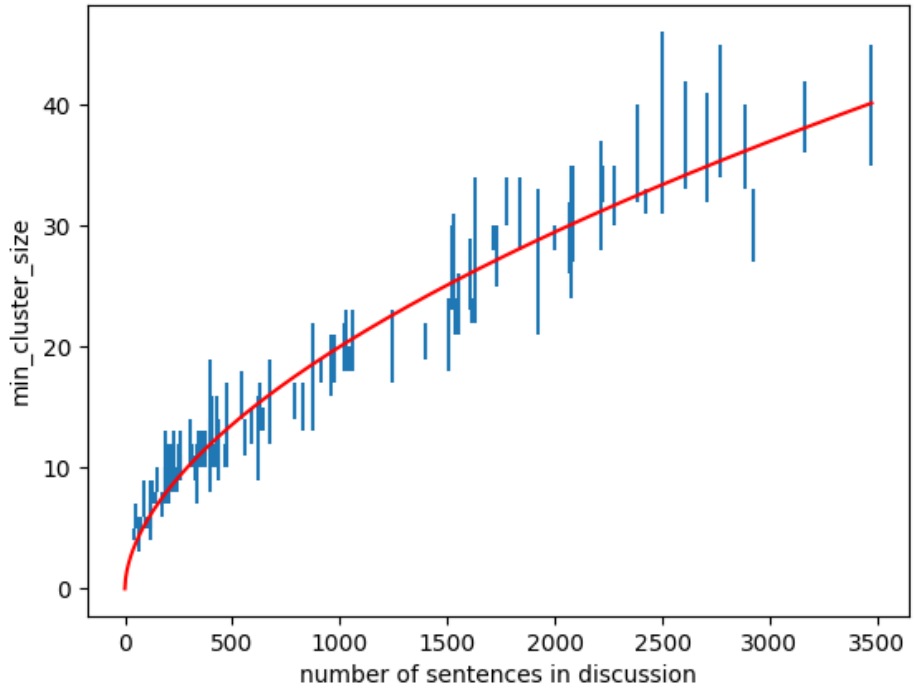


Figure 7.1: Vertical blue bars represent the range for the `min_cluster_size` parameter that results in effective clustering for the respective discussions. The red curve represents the optimal regression fit.

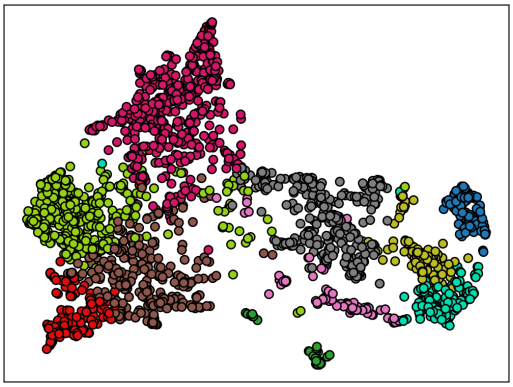
the clustering appears natural, essential clusters are identified, and clusters should not be merged further.
We established a regression model based on the function family $f(x|a,b) = a \cdot x^b$, where x represents the number of sentences in the discussion, and the output variable is the average of the upper and lower bound for `min_cluster_size`. This model was chosen due to its desirable properties: it has a fixed point at $f(0) = 0$ (reflecting that an empty discussion should have no clusters and a short discussion should only have a few clusters compared to a longer one), and its simplicity aids interpretation while guarding against overfitting.
The resulting function for computing `min_cluster_size` is given by Equation 7.1.

$$f(x) = 0.421 \cdot x^{0.559} \tag{7.1}$$

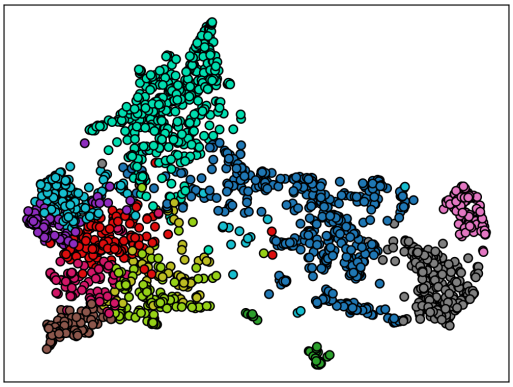
Figure 7.1 visualizes upper and lower bounds as well as the found model.

`min_samples` This parameter corresponds to the k parameter in Equation 4.1. Increasing this value reduces the number of points within a cluster, emphasizing the points in the core of a cluster. The smallest valid value is 1, and it cannot exceed the value chosen for `min_cluster_size`. To precisely capture the central meaning of a cluster, we set this value to its maximum, prioritizing the core elements of a cluster.

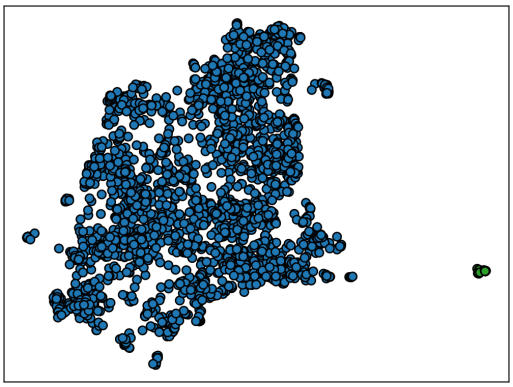
`cluster_selection_method` This parameter accepts the options "eom" and "leaf". "eom" determines clusters by selecting those with high stability from the hierarchy, while "leaf" treats each leaf as its own cluster (refer to Section 4.3). We exclusively employ the "leaf" method in our experiments because "eom" tends to create unreasonably large clusters, as illustrated in Figure 7.2c. There are cases where it produces only two or three clusters even for extensive discussions. The "leaf" method doesn't encounter this issue but is more reliant on the `min_cluster_size` parameter.



(a) An example clustering using a `min_cluster_size` value depending on the discussion size and setting `cluster_selection_method` to "leaf"



(b) Same example as (a), using `cluster_selection_method` "eom" with a `min_cluster_size` of 15.



(c) This clustering with "eom" results in just two clusters, one significantly larger than the other.

Figure 7.2: An example of our sentence clustering approach using "leaf" compared to "eom". While "eom" typically produces effective clusterings, it can experience significant failures, as shown in (c). Utilizing "leaf" with the described parameters yields comparable results to "eom" but without the drawbacks of "eom".

7.3 Meta Sentence Removal

Every discussion consists of meta sentences that do not add valuable information to its topic but instead show interactions between the participants:

- Well said.
- I'll give you a !delta for this.
- The problem I have with your argument is that I feel like the logic is flawed.
- Do you have the link for that?
- You are putting up a straw man.
- Deltabot is having some issues at the moment.

Including these sentences in clustering can lead to noisy clusters that are challenging to interpret or clusters consisting entirely of meta sentences.
We developed a method for "meta sentence removal" on a domain basis (in our case, argumentative discussions from ChangeMyView). Using this approach, we estimated that approximately 23% of all sentences in a ChangeMyView discussion are meta sentences. While this method can be applied to other domains, additional adjustments may be required based on the specific domain characteristics.

General Outline of the Approach

1. **Reference Set Creation** Build a reference set of meta sentences by applying the sentence clustering procedure to various discussions, identifying clusters of meta sentences manually, and including them in the reference set. This step needs to be done only once to obtain a reference set that can be reused to filter meta sentences from any discussion by applying the following two steps.

2. **Meta Sentence Clustering** Randomly draw sentences from the reference set and cluster them together with sentences from the discussion for which meta sentences need to be removed.

3. **Meta Sentence Classification** Classify clusters that contain a lot of sentences from the reference set as meta clusters and remove all sentences that are part of or close to a meta cluster from the discussion.

Figure 7.3 illustrates an example of the approach.

7.3.1 Reference Set Creation

1. Initialize the reference set $A \coloneqq \emptyset$.
2. Draw a random discussion from the database that has not been drawn yet and apply the sentence clustering procedure (refer to Section 7.2).
3. Manually inspect the created clusters and add all sentences from clusters that mainly consist of meta sentences to the reference set A.
4. Return to step 2 if either the reference set size limit is not reached or the meta sentence removal is unsatisfactory when implementing the complete removal algorithm.

7.3.2 Meta Sentence Clustering

Let D be the discussion for which meta sentences have to be removed, $|D|$ the number of sentences in D and $|A|$ the size of the reference set A.
Draw random sentences from A and collect them into a new set N with $|N| = \min(\max(|D|, 300), |A|)$. We choose an equal amount of meta sentences because taking too much samples from the reference set may distort the embedding space and therefore yield an higher amount of false positives. Taking too little samples on the other hand gives an high variance estimate of the meta regions in the embedding space and therefore unreliable results. Consequently, we take at least 300 samples from our reference set and at most $|A|$ since we cannot take more samples.
We apply our sentence clustering procedure (refer to Section 7.2) to the combined sentences $N \cup D$ and keep all parameters for the clustering unchanged (refer to Section 7.2.2 and 7.2.3) except for the following parameters:

- `min_samples`: We set this parameter to 1 because it creates larger clusters which offers a more accurate approximation of the density.
- `min_cluster_size`: Choosing an appropriate value for this parameter is crucial. A high value can result in the merging of distinct clusters, while a very low value can lead to high variance estimates. We have found that using a value of 15 produces satisfactory results.

7.3.3 Meta Sentence Classification

Let C_n be the set of sentences that are part of some cluster n. Calculate the probability of a sentence originating from N when selecting a sentence from

C_n:

$$P(N|C_n) = \frac{|N \cap C_n|}{|C_n|} \tag{7.2}$$

The probability of a sentence coming from N when drawing a sentence from $N \cup D$ is:

$$P(N) = \frac{|N|}{|D| + |N|} \tag{7.3}$$

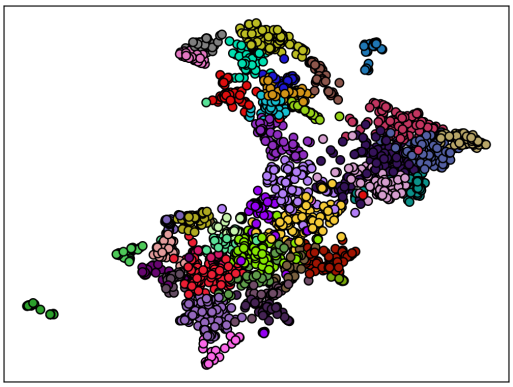
As we anticipate, meta sentences should cluster alongside other meta sentences, and non-meta sentences should cluster alongside other non-meta sentences. Consequently, clusters with a high number of sentences from N are also likely to contain many meta sentences from D, whereas clusters with only a few sentences from N should have very few meta sentences from D. Under the assumption of independence we expect: $P(N|C_n) = P(N)$.
Thus, if we observe that $P(N|C_n) \ll P(N)$, it suggests that cluster C_n contains significantly fewer meta sentences and should not be classified as meta. Conversely, if we observe that $P(N|C_n) \gg P(N)$, it indicates that cluster C_n contains a considerable number of meta sentences and should be categorized as meta. Therefore, it is reasonable to classify a cluster as meta when:

$$P(N|C_n) > P(N) \tag{7.4}$$

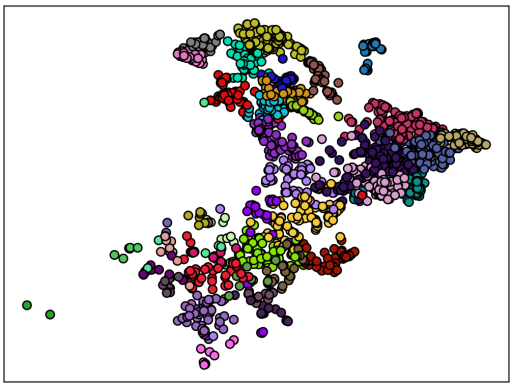
We noticed that certain types of meta sentences are less common in the reference set A, likely due to variations during its creation or when randomly selecting sentences from it to form N. Additionally, we observed that clusters containing non-meta sentences almost never mix with sentences from N. As a result, we have decided to reduce the threshold by applying an empirically determined coefficient of $\frac{2}{3}$ to minimize the number of false negatives without significantly impacting the number of false positives:

$$P(N|C_n) > \frac{2}{3} \cdot P(N) \tag{7.5}$$

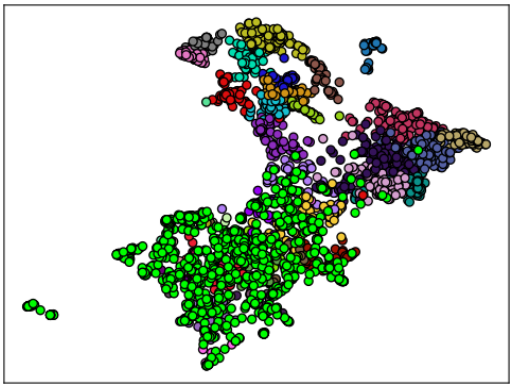
We classify sentences as meta sentences if they are part of a meta cluster or their closest cluster is a meta cluster and remove them from the discussion to obtain the cleaned discussion.



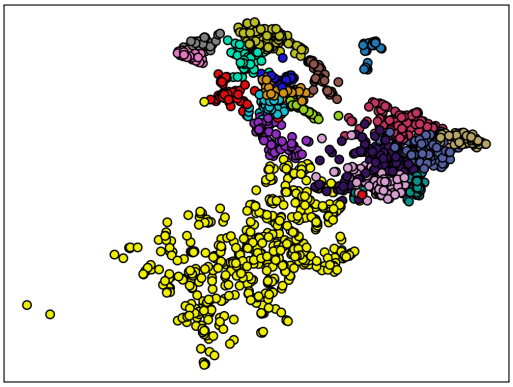
(a) Clustering showing points representing sentences from $N \cup D$.



(b) Same clustering as (a), but only showing points representing sentences from D.



(c) Same as (a), but points representing sentences from N are colored in green.



(d) Same as (b) but points representing sentences classified as meta due to the "Meta Sentence Classification" step are colored in yellow.

Figure 7.3: Example clustering for the sentence removal approach on a discussion D with 2166 sentences and a meta sentences set N containing 955 sentences. If not specified otherwise, the colors of points indicate the cluster membership of corresponding sentences as a result of applying the "Meta Sentence Clustering" step. (c) shows the estimation of the meta density while (d) shows sentence from D that lie in this meta density and are therefore flagged for removal.

Chapter 8

Cluster Labeling

In order to generate an appropriate label for a cluster that effectively describes its content, we combine the sentences within the cluster to create a document. This document is then utilized as context within a template, which guides the LLM to generate a label for the cluster.

Context Trimming We must remove sentences from larger clusters due to token limitations in the employed models. This is achieved by arranging sentences based on their persistence value (explained in Section 4.3, denoted as $\lambda_{p,c}$) in a descending order. Sentences are then progressively discarded from the tail end of the sorted list until the tokenized input fits within the model's capacity.
This approach prioritizes the removal of sentences that are further from the cluster center (i.e., sentences added later during clustering), while retaining those at the cluster's core.
For decoder-only models, it is necessary to discard additional sentences to accommodate each output token's inclusion in the input during generation, requiring a slight buffer at the end of the sequence.

Pre-ChatGPT and Post-ChatGPT Models As described in Chapter 6, our initial experiments in early 2023 involved only five out of the total 19 LLMs. Subsequently, when the remaining 14 more capable models became available, experiments were conducted with these models.
Notably, the Post-ChatGPT models demonstrate strong adherence to instructions, while the majority of Pre-ChatGPT models struggle with comprehension. To address this, various prompts were tested extensively with the Pre-ChatGPT models in an attempt to identify prompts that effectively guided each model's behavior.
We included the Pre-ChatGPT models in new experiments, employing a generic

prompt as outlined in Section 6.2.1 to simplify experiments and reduce variability. However, we designed these prompts and instructions based on the insights from the old experiments with the Pre-ChatGPT models.
The old experiments, which involved only the five Pre-ChatGPT models and includes a manual evaluation, are detailed in Section 8.2. Subsequent experiments involving all 19 LLMs are discussed in Section 8.3, with a focus on automatic evaluation.

Mistakes In our old approach, while searching for an optimal prompt for our models, we inadvertently maximized BERTScore precision instead of the intended BERTScore F-measure. This error was isolated to this specific instance, and throughout the thesis, all other references to BERTScore included reporting precision, recall, and F-measure as intended.
It is worth noting that this mistake may not have had a significant impact, as precision emphasizes accurate but concise labels, whereas recall prioritizes longer labels that may contain more erroneous information.

8.1 Evaluation Dataset

We randomly sampled 300 discussions from the ChangeMyView dataset and applied our frame identification approach (see Chapter 7) to each discussion. This process resulted in a total of 2,666 frames/clusters, from which we selected 300 randomly for our experiments. For each of these selected clusters, we manually assigned a concise label that captures the cluster's core meaning.

Limitations Clusters consist of sentences that are highly similar but do not logically connect, making it difficult to combine them to a coherent and easily readable text. As a result, assigning a single label that captures the cluster's meaning is a demanding and time-consuming task for human labelers.
In many instances, there is no clear, definitive choice for a label, and different labelers may propose varying labels based on their interpretations of the cluster. Furthermore, the clusters exhibit significant diversity, and some topics can be challenging to understand for labelers who lack familiarity with them. This inherent bias introduces difficulty in distinguishing subtle differences between highly capable models that approach human performance. Therefore, our focus primarily centers on comparing models based on general patterns in automatic evaluations.

8.2 Old Approach

We identify a specific prompt for each of the five Pre-ChatGPT model that effectively guides its behavior in generating cluster labels.
Furthermore, we carried out experiments involving keyphrase extraction, but there were no benefits observed for label generation, as detailed in Appendix A. For a manual evaluation of this approach, refer to Section 10.1.1.

8.2.1 T0++

T0++ [79] is an encoder-decoder model that is capable of comprehending and executing instructions for various tasks without requiring additional fine-tuning. It was derived from fine-tuning a T5 model [75] using the P3 dataset, which was constructed using PromptSource [7]
PromptSource consists of templates designed for 180 supervised datasets, transforming their respective tasks into natural language instructions.
Here are two templates showcasing the task conversion for the Extreme Summarization (XSum) [65] dataset, which is intended for summarization tasks:

```
{document}
How would you rephrase that in a few words? ||| {summary}
```

```
Article: {document}
Summary: ||| {summary}
```

To train **T0++**, templates are converted into prompts by replacing template placeholders with specific values derived from samples within the dataset. The text preceding the "`|||`" characters is input to the encoder, while the decoder's objective is to generate the text following the "`|||`" characters.
In Table 8.1, we present datasets that we identified as relevant to label generation tasks by reviewing PromptSource. After a thorough examination of these datasets, we obtained the following insights:

- The context if typically described by a single word, such as "Article", or by a question that refers to the context.
- Instructions are commonly separated from the context using either newlines or colons. When context is concise, it is often directly incorporated into the instruction within quotation marks.
- Commonly occurring terms that could be beneficial for the task of label generation are: "concepts", "content", "essential ideas", "headline", "sentence", "subject", "summary", "title", "tl;dr", and "topic".

Table 8.1: Datasets retrieved from PromptSource relevant to the label generation task.

Dataset	Description
aeslc	Email messages from employees in the Enron Corporation.
billsum	Summaries of US Congressional and California state bills.
cc_news	News articles from various news sites.
common_gen	Generative commonsense reasoning.
dbpedia_14	Texts associated with a title and one of 14 non-overlapping classes.
gigaword	Headline-generation on article pairs.
multi_news	News articles and human-written summaries from newser.com.
narrativeqa	Stories and questions to test reading comprehension.
samsum	16k messenger-like conversations with summaries.
sent_comp	Instances on which deletion-based algorithms can be trained.
wiki_qa	Question and sentence pairs for open-domain question answering.
xsum	One sentence summaries for news articles.

Prompts

We generate various prompts inspired by patterns identified in PromptSource to assess the performance of **T0++**.
We use the term `input_type` to describe the context and `output_type` to describe the text that the model has to generate. These serve as placeholders in our templates, enabling us to experiment with diverse phrasings to optimize model performance.

prefix Prompt 8.1 presents the instruction before the context. This tests whether the models have a preference for instructions at beginning of the prompt.

```
What {output_type} would you choose for the {input_type} below?

{text}
```

Prompt 8.1: prefix

postfix In contrast to the prefix prompt, prompt 8.2 presents the instruction after the context, aiming to evaluate whether the models favor instructions positioned at the end.

```
{text}

What {output_type} would you choose for the {input_type} above?
```

Prompt 8.2: postfix

prefix-postfix Prompt 8.3 presents the instruction both before and after the context, testing whether this redundancy encourages the model to adhere to the instruction with greater reliability.

```
What {output_type} would you choose for the {input_type} below?

{text}

What {output_type} would you choose for the {input_type} above?
```

Prompt 8.3: prefix-postfix

short Prompt 8.4 exclusively features the words describing the `input_type` and `output_type`, assessing whether shorter instructions increase the model's adherence to instructions.

```
{input_type}:
{text}
{output_type}:
```

Prompt 8.4: short

explicit Prompt 8.5 resembles the short prompt but employs uppercase letters for the instruction and explicitly marks the context's beginning and end. It also capitalizes the words representing `input_type` and `output_type`.

```
{input_type} START
{text}
{input_type} END
{output_type} OF THE {input_type}:
```

Prompt 8.5: explicit

question answering The **T0++** model underwent fine-tuning using numerous question answering datasets. Prompt 8.6 formulates the cluster labeling task as a question answering task.

```
Read the following context and answer the question.
Context:
{text}
Question: What is the {output_type} of the {input_type}?
Answer:
```

Prompt 8.6: question answering

Table 8.2: Variables for optimizing label generation prompts, with the default values determined by the most commonly used values in PromptSource.

Factor	Possible Values	default
template	explicit, postfix, prefix, prefix-postfix, question answering, short	–
`input_type`	argumentation, article, conversation, debate, dialogue, discussion, email, speech, text	text
`output_type`	concept, content, core information, description, essence, question, subject, subject matter, summary, theme, thesis, title, topic	title

Finding a Good Prompt

Our objective is to identify a single prompt that consistently produces favorable results for the label generation task. Table 8.2 provides a summary of all the factors that can be adjusted in our search for an optimal prompt. To determine the ideal value for each factor, we optimize for BERTScore precision according to the following sequence:

$$\text{template} \rightarrow \texttt{input\_type} \rightarrow \texttt{output\_type}$$

We configure all remaining factors based on their best-known value, if available. Otherwise, we set them to their default values, as outlined in Table 8.2.

Errors As **T0++** is a multi-tasking model capable of handling various tasks, there are instances where it misinterprets instructions, leading to the generation of non-viable answers. Table 8.3 provides insights into the frequency of specific erroneous outputs generated by **T0++**.

Table 8.3: List of errors made by the **T0++** model and how often they were generated out of 12600 generations. Errors were identified by skimming over all generated labels with a focus on frequently generated outputs.

Output	Number of Occurrences
not enough information	390
no	207
a debate	19
answer not in context	16
yes	3
option	2
not logical	1

Table 8.4: Average BERTScore precision values for different prompts.

Prompt	Mean	95% CI	#Errors
question answering	0.152	[0.122, 0.182]	14
prefix-postfix	0.112	[0.085, 0.140]	10
prefix	0.098	[0.068, 0.128]	14
postfix	0.069	[0.037, 0.102]	31
explicit	0.033	[-0.001, 0.066]	38
short	-0.072	[-0.105, -0.040]	60

Best Prompt Table 8.4 displays the labeling performance of the developed templates, while Table 8.5 and Table 8.6 present the performance based on different `input_type` and `output_type` configurations, respectively. Prompt 8.7 presents the most effective prompt for the **T0++** model.

Table 8.5: Average BERTScore precision values for different `input_type` values.

Input Type	Mean	95% CI	#Errors
discussion	0.185	[0.156, 0.214]	5
dialogue	0.181	[0.151, 0.211]	3
conversation	0.172	[0.143, 0.201]	7
article	0.157	[0.128, 0.187]	7
debate	0.157	[0.128, 0.186]	7
text	0.152	[0.122, 0.182]	14
argumentation	0.148	[0.120, 0.177]	2
email	0.114	[0.083, 0.145]	24
speech	0.052	[0.022, 0.082]	46

Table 8.6: Average BERTScore precision values for different `output_type` values.

Output Type	Mean	95% CI	#Errors
title	0.152	[0.122, 0.182]	14
question	0.149	[0.128, 0.169]	0
topic	0.139	[0.112, 0.167]	2
subject matter	0.128	[0.097, 0.159]	25
concept	0.127	[0.099, 0.155]	21
subject	0.124	[0.093, 0.155]	25
thesis	0.120	[0.096, 0.143]	7
summary	0.119	[0.096, 0.142]	2
essence	0.116	[0.093, 0.139]	4
theme	0.114	[0.087, 0.141]	28
core information	0.106	[0.083, 0.129]	10
content	0.077	[0.050, 0.105]	21
description	-0.016	[-0.044, 0.012]	119

```
Read the following context and answer the question.
Context:
{text}
Question: What is the title of the discussion?
Answer:
```

Prompt 8.7: Best prompt for the **T0++** model.

8.2.2 BLOOM, OPT-66B, GPT-NeoX

BLOOM, **OPT-66B**, and **GPT-NeoX** are decoder-only models that underwent only pre-training. Consequently, they are unable to reliably adhere to instructions like **T0++** or **GPT3.5**. To address this limitation, we had to present tasks to these models indirectly.

To explore effective prompts and phrasings, we utilized `alpa.ai`[1], an online platform for experimenting with **OPT-175B**. We developed six prompt templates, drawing inspiration from ideas employed for **T0++** prompts.

Unlike our extensive experimentation with **T0++**, we conducted fewer trials for these models due to the absence of references like PromptSource.

Just like **T0++**, we use `input_type` to denote words describing the context and `output_type` to describe the text the model needs to generate. Acceptable values for `input_type` are "discussion" and "dialogue", while `output_type` accepts "title" and "topic". We opted for "title" and "topic" over "label" due to

[1] `https://alpa.ai/opt` (accessed: 19.01.2023)

their unambiguous semantics, as "label" can have various interpretations.
We incorporated a quotation mark at the end of prompts and stopped generation when the model produced a second quotation mark. This method facilitates the detection of when the generation needs to stop, which is particularly crucial for label generation tasks where quoted text tends to be concise.

Prompts

explicit Prompt 8.8 is inspired by the **T0++** prompt 8.5 and is designed to test how the models handle concise instructions.

```
{input_type} START
{text}
{input_type} END
{output_type} OF THE {input_type}: "
```

Prompt 8.8: explicit

question answering Prompt 8.9 draws inspiration from **T0++** prompt 8.6 and aims to test how the models perform when formulating the labeling task as a question-answering task.

```
{input_type} START
{text}
{input_type} END
Q: What is the {output_type} of the {input_type}?
A: The {output_type} of the {input_type} is "
```

Prompt 8.9: question answering

fake debate Prompt 8.10 presents the context as a debate between two persons. Prior to incorporating the text into the template, we preprocess it by appending "Person x:" to the start of each sentence, with x alternating between "1" and "2". For example:

```
Person 1: {sentence 1}
Person 2: {sentence 2}
Person 1: {sentence 3}
Person 2: {sentence 4}
```

The resulting debate may not necessarily resemble a typical debate, as, for instance, person 2 can present an argument that supports person 1's position.

Additionally, arguments typically span beyond a single sentence. Our hypothesis is that this approach encourages the model to adhere to the instruction more consistently and reliably, as the contextual presentation becomes more plausible.

```
The {input_type} between Person 1 and Person 2 begins.
{text}
Thank you very much, this is the end of the {input_type}. The
↪  {output_type} of the {input_type} was "
```

Prompt 8.10: fake debate

assistant solo Prompt 8.11 employs an AI assistant to integrate the instruction into the prompt. Upon completing the prompt, the model is tasked with acting as the assistant and generating a label that aligns with the given instruction. We present the context in the same way as we did with the fake debate prompt 8.10.

```
AI assistant: I am an expert AI assistant and I am very good in
↪  identifying {output_type} of {input_type}.
{input_type} START
{text}
{input_type} END
AI assistant: The {output_type} of the {input_type} between the two
↪  participants is "
```

Prompt 8.11: assistant solo

assistant solo about Prompt 8.12 is the identical to prompt 8.11 but employs the word "about" inplace of one of the valid values for `input_type`. In this manner, we evaluate how the model performs when using a term with less clear semantics than "title" or "topic".

```
AI assistant: I am an expert AI assistant and I am very good at
↪  recognizing what {input_type} are about.
{input_type} START
{text}
{input_type} END
AI assistant: The {input_type} between the two participants is about "
```

Prompt 8.12: assistant solo about

assistant interaction Prompt 8.13 presents the instruction as a dialogue between an AI assistant and a human actor. We hypothesize that this format enhances the model's ability to assist more reliably compared to using only the assistant.

```
AI assistant: I am an expert AI assistant. How can I help you?
Human: Can you tell me what the {output_type} of the following
↪  {input_type} is?
{input_type} START
{text}
{input_type} END
AI assistant: The {output_type} of the debate is "
```

Prompt 8.13: assistant interaction

Best Prompts

To address the substantial computational demands, we assess all prompts with each model by using only the first 10 examples from our evaluation dataset. We conduct manual inspections of the generated results and retain only the prompts that yield the most satisfactory outcomes for each prompt template. The chosen prompts undergo additional evaluation using BERTScore, while those that produce a substantial number of incorrect outputs are eliminated from consideration.
Common errors are:

- Ignoring the instruction
- Generating very long labels
- Generating a new discussion
- Acting as a participant of the discussion

Table 8.7, Table 8.8, and Table 8.9 display the labeling performance. The values in brackets following the prompt name represent the parameters used for `input_type` and `output_type`. Prompts 8.14, 8.15, and 8.16 are the most effective prompts for BLOOM, OPT, and GPT-NeoX, respectively.

Table 8.7: Mean BERTScore values for the different prompts used with the **BLOOM** model.

Prompt	Mean	95% CI
assistant solo (debate, title)	0.150	[0.126, 0.173]
assistant interaction (debate, title)	0.130	[0.108, 0.152]
explicit (debate, title)	0.127	[0.104, 0.149]

```
AI assistant: I am an expert AI assistant and I am very good in
↪  identifying titles of debates.
DEBATE START
{text}
DEBATE END
AI assistant: The title of the debate between the two participants is "
```

Prompt 8.14: Best prompt for **BLOOM**

Table 8.8: Mean BERTScore values for the different prompts used with the OPT model.

Prompt	Mean	95% CI
assistant solo about (debate)	0.154	[0.124, 0.184]
question answering (debate, topic)	0.149	[0.126, 0.172]
fake debate (debate, topic)	0.139	[0.116, 0.162]
explicit (debate, topic)	0.131	[0.107, 0.154]
assistant interaction (debate, title)	0.100	[0.078, 0.121]

```
AI assistant: I am an expert AI assistant and I am very good at
↪  recognizing what debates are about.
DEBATE START
{text}
DEBATE END
AI assistant: The debate between the two participants is about "
```

Prompt 8.15: Best prompt for **OPT-66B**

Table 8.9: Mean BERTScore values for the different prompts used with the **GPT-NeoX** model.

Prompt	Mean	95% CI
question answering (discussion, topic)	0.181	[0.156, 0.206]
fake debate (debate, topic)	0.161	[0.138, 0.184]
assistant solo (debate, title)	0.140	[0.115, 0.164]
assistant interaction (debate, title)	0.128	[0.103, 0.154]

```
DISCUSSION START
{text}
DISCUSSION END
Q: What is the topic of the discussion?
A: The topic of the discussion is "
```

Prompt 8.16: Best prompt for **GPT-NeoX**

8.2.3 GPT3.5

GPT3.5 excels in following and understanding instructions, demonstrating robustness against variations in prompt wording. Consequently, our primary aim is to craft a single, clear prompt that effectively directs the model to generate the desired outputs.
Prompt 8.17 incorporates the following instructions:

- **single descriptive phrase**: We seek a concise text, shorter than a sentence, that captures the essence of the cluster.
- **simple language**: This instruction is intended to yield easily understandable labels, even for complex debates that require a deep understanding of the topic.
- **without talking about the debate or the author**: The purpose of this instruction is to reduce the frequency of labels beginning with uninformative phrases like "The debate is about" or "The author talks about".

Table 8.10 highlights the prompt's performance, notably achieving the highest score when compared to other models.

```
Generate a single descriptive phrase that describes the following debate
↪  in very simple language, without talking about the debate or the
↪  author.

Debate: """{text}"""
```

Prompt 8.17: `GPT3.5`

Table 8.10: Average BERTScore precision values for the prompt used with the **`GPT3.5`** model.

Prompt	Mean	95% CI
instruct gpt phrase	0.220	[0.203, 0.238]

8.3 New Approach

In this section, we present instructions with which we configured the 19 LLMs for cluster labeling, following the setup described in Chapter 6. For our evaluation, that focuses solely on automated rather than manual assessment, refer to Section 10.1.2.

8.3.1 Instructions

We employ instruction 8.1 for direct models and instruction 8.2 for dialogue models, as detailed in Section 6.2.2. These instructions drew inspiration from the prompt used with **`GPT3.5`** (prompt 8.17). For a deeper understanding of the design rationale behind these instructions, refer to Section 8.2.3.

```
Every input is the content of a debate. For every input, you generate a
↪  single descriptive phrase that describes that input in very simple
↪  language, without talking about the debate or the author.
```

Instruction 8.1: Direct instruction for cluster labeling.

```
A chat between a curious user and an artificial intelligence assistant.
The user presents a debate and the assistant generates a single
↪  descriptive phrase that describes the debate in very simple
↪  language, without talking about the debate or the author.
```

Instruction 8.2: Dialogue instruction for cluster labeling.

Chapter 9

Frame Assignment

We investigate the capacity of LLMs to attribute generic media frames to cluster labels, as introduced Boydstun et al. [16]. For detailed descriptions of all 14 generic media frames, both long and short, refer to Appendix B.
We provide the models with varying amount of information about the available frames, which we call frame context, and instruct them to assign three media frames in order of importance. For the automatic evaluation of frame assignment, refer to Section 10.2.

9.1 Evaluation Dataset

We construct our evaluation dataset by repurposing the dataset from our manual evaluation (refer to Section 8.1). For each of the 300 labels generated by the top-performing model in our manual evaluation, **`GPT3.5`**, we assign a maximum of two media frames. We exclude 15 labels for which we couldn't assign any media frames. Additionally, we randomly select three labels for each of the 14 media frames as few-shot examples from the remaining 285 labels, resulting in 243 labels available for our experiments. Among these, 126 have one media frame assigned, while 117 have two media frames assigned.

9.2 Frame Context

We aim to investigate how varying the amount of information about available frames impacts frame assignment performance.
Even with highly capable models, we found that providing no information about the available frames resulted in unreliable generation of valid media frames. Consequently, we provide at least the names of the available frames as context.

To present the frames along with different types of information effectively, we use a JSON format. All tested models, trained on code snippets, should have the capability to comprehend this format.
We've developed the following frame contexts, ordered based on the level of information they offer:

Zero-Shot (extreme) This frame context, as displayed in Appendix C.1, provides a JSON list containing the names of the 14 valid frames. This setup enables us to evaluate whether the models possess implicit knowledge of the meanings linked to these frames.

Zero-Shot (short) This frame context, as displayed in Appendix C.2, provides the names of the 14 valid media frames as a JSON object together with brief descriptions extracted from Card et al. [21, p. 2], which consist of concise phrases highlighting the key aspects of each media frame.

Zero-Shot (long) This frame context, as displayed in Appendix C.3, provides the names of the 14 valid frames as a JSON object together with lengthy descriptions extracted from Boydstun et al. [16, pp. 6–7], which provide detailed, multi-sentence explanations of for each media frames.

Few-Shot This frame context, as displayed in Appendix C.4, extends the "Zero-shot (long)" frame context by additionally providing three examples for each media frame.

9.3 Instructions

For direct models, we utilize instruction 9.1, and for dialogue models, we employ instruction 9.2 (refer to Section 6.2.2). We substitute the `frames` placeholder with the frame context and the "`object`" placeholder with either "json" or "list" depending on the type of the JSON object used to represent the frame context.

```
The following {object} contains all available media frames as defined in
↪ the work from Boydstun, Amber E. et al. "Tracking the Development of
↪ Media Frames within and across Policy Issues." (2014): {frames}
For every input, you answer with three of these media frames
↪ corresponding to that input, in order of importance.
```

Instruction 9.1: Direct instruction for frame assignment with reference to the scientific work where the taxonomy of media frames was first introduced.

```
A chat between a curious user and an artificial intelligence assistant.
The assistant knows all media frames as defined by Boydstun, Amber E. et
↪  al. "Tracking the Development of Media Frames within and across
↪  Policy Issues." (2014): {frames}
The assistant answers with three of these media frames corresponding to
↪  the user's text, in order of importance.
```

Instruction 9.2: Dialogue instruction for frame assignment with reference to the scientific work where the taxonomy of media frames was first introduced.

9.3.1 Citation Effect

We propose that referencing the paper that originally introduced the media frame categorization assists the models in recalling relevant ideas, as they may have encountered the media frame concept during their training.
We utilize the following citation to refer to the introductory paper:

```
Boydstun, Amber E. et al. "Tracking the Development of Media Frames
↪  within and across Policy Issues." (2014)
```

We test this hypothesis for **Falcon-40B**, **ChatGPT**, and **LLaMA-65B** by comparing instructions that do not include the citation (instructions 9.3 and 9.4) with instructions that explicitly mention the citation (instructions 9.1 and 9.2).

```
The following {object} contains all the available media frames: {frames}
For every input, you answer with three of these media frames
↪  corresponding to that input, in order of importance.
```

Instruction 9.3: Direct instruction for frame assignment without a reference to the introductory work.

```
A chat between a curious user and an artificial intelligence assistant.
The assistant knows all the following media frames: {frames}
The assistant answers with three of these media frames corresponding to
↪  the user's text, in order of importance.
```

Instruction 9.4: Dialogue instruction for frame assignment without a reference to the introductory work.

Table 9.1 demonstrates a significant advantage in citing the paper when offering limited frame context for **Falcon-40B** and **LLaMA-65B**, with no discernible impact on **ChatGPT**. Consequently, we retain the citation for our experiments.

Table 9.1: Examining the influence of citing the paper that introduced the 14 generic media frames from Boydstun et al. [16] as additional information in the instructions for the frame assignment task. Providing citation information (**Cite.**) shows up to 12% improvement for **Falcon-40B** and 9% for **LLaMA-65B** in the zero-shot scenario, where only frame labels are provided in the prompt. Few-Shot values for **LLaMA-65B** are missing due to insufficient graphic memory resources.

Prompt	**Falcon-40B**		**ChatGPT**		**LLaMA-65B**	
	Cite.	–	**Cite.**	–	**Cite.**	–
Zero-Shot (extreme)	46.5	34.2	60.9	60.1	53.1	44.4
Zero-Shot (short)	46.5	42.8	58.0	57.2	50.6	42.4
Zero-Shot (long)	46.1	46.5	58.8	60.9	39.5	39.1
Few-Shot	38.3	39.1	63.4	64.6	–	–

Chapter 10

Evaluation

In this chapter, we present the evaluation for both the developed cluster labeling and frame assignment approach.

10.1 Cluster Labeling

For the evaluation of cluster labeling we employed the evaluation dataset presented in Section 8.1.

10.1.1 Old Approach

In this section we compare the labeling performance for the Pre-ChatGPT models with their found best prompts, as described in Section 8.2, by conducting an manual evaluation to assess their performance.
For each cluster, we present the generated labels to four human annotators and ask them to rank labels to determine the best-performing model.

Label Preprocessing

We preprocessed the generated labels to increase the difficulty for human annotators to identify the label model, based on factors such as label length, label structure, and word choices.

Long Labels Some models have a tendency to generate particularly long labels, as demonstrated in the following example:

- The universe is very large, and very old. Why would that universe be any more likely than the one we think exists?

We reduce label length by splitting the label into sentences and selecting the first sentence if the total number of tokens in the string exceeds 15. This transforms the previous example to:

- The universe is very large, and very old.

Choice of words The **GPT3.5** model often employs language such as "Exploring", "Arguments for", and "A discussion of". This usage can be observed in the following examples:

- Exploring the complexities of food choices and their effects on health.
- Arguments for Intelligent Design
- A discussion of the pros and cons of voluntary registration.

We employ the following regular expression to delete superficial language from the beginning of labels:

```
^a?\s*\w*\s*(argument|debate|discussion|exploring)s?
↪  (of|about|on|against)?\s*(the)
```

Out of the 300 labels generated by **GPT3.5**, 117 of them match this expression. It transforms the previous examples to:

- complexities of food choices and their effects on health.
- Intelligent Design
- pros and cons of voluntary registration.

General Transformations We replace sequences of whitespace characters with a single space and capitalize the first letter of the label. If a label doesn't end with a '.', '?', or '!', we append a '.' to the end.

Redundant Labels When multiple models generate the same label, we present the label only once to the user and assign the same rank to the models that generated that label.

Evaluation Interface

We present the discussion title and the reference label to the user (Figure C.1) and ask the user to rank the generated labels (Figure C.2) based on how well they correspond to the reference (refer to Section C.5.1 for the precise instructions). It is worth noting that the reference can be biased and may not always accurately describe the cluster. To provide labelers with a more

comprehensive understanding of the context and potential bias sources, we also present 5 random sentences from the cluster and 5 sentences from the cluster that are most similar to the reference (Figure C.3).

Combining User Rankings

Rank Fusion To arrive at a final ranking that makes user preferences comparable across models, we employ Reciprocal Rank Fusion (RRF) [27].
Given a set of documents D and a set R of permutations of these documents, a score is computed for each document according to Equation 10.1.

$$\mathrm{RRF}_{\mathrm{score}}(d \in D) = \sum_{r \in R} \frac{1}{k + r(d)} \tag{10.1}$$

The documents are sorted based on their scores to obtain the fused ranking. Following the recommendations of Cormack, Clarke, and Büttcher, we set the constant k to a value of 60 to mitigate the influence of exceptionally high rankings from outlier systems.

Transforming the Rankings of Labels into a single Ranking of Models
The annotators produced rankings of the generated labels for each cluster. To obtain a unified ranking of models, we sorted the models based on the $\mathrm{RRF}_{\mathrm{score}}$ of their generated labels. The rank assigned to a model corresponds to the position of the first model in the sorted list with the same $\mathrm{RRF}_{\mathrm{score}}$. This procedure is essential because different labels can receive the same score, and certain labels are generated by multiple models, as explained in Section 10.1.1.

Results

Table 10.1 displays the average ranks assigned to the models, with **GPT3.5** evidently producing more favorable labels than the other models. We calculated Kendall's W for rank correlation [46] to gauge the agreement among annotators. The resulting value of 0.66 indicates a high level of agreement.

Table 10.1: Results of qualitative evaluation of generative cluster labeling across the fused rankings. Presented are the average model ranks, the number of times a model was ranked first, and statistics regarding label lengths. Notably, **GPT3.5** demonstrated superior performance compared to other models and tended to generate longer labels, on average.

Model	Mean Rank	#First	Length		
			Min	Max	Mean
GPT3.5	**1.38**	**225**	3	27	9.44
BLOOM	2.95	33	1	37	8.13
GPT-NeoX	3.20	20	1	34	7.42
OPT-66B	3.36	12	1	30	8.27
T0++	3.72	28	1	18	3.10

10.1.2 New Approach

In this section, we compare all 19 LLMs based on the setup described in Section 8.3.
Table 10.2 displays model performance measured in ROUGE, while Table 10.3 illustrates performance based on BERTScore.

General Findings **ChatGPT** demonstrates the best performance, followed by the open-source **LLaMA-CoT** model.
In general, models that undergo instruction-tuning outperform those that do not receive such training. This difference is particularly evident when comparing **Falcon-40B-I**, achieving an BERTScore F-measure of 0.20, to the base model (**Falcon-40B**) with an F-measure of 0.10, and **LLaMA-CoT**, which achieves an F-measure of 0.22, compared to the base model (**LLaMA-30B**) with an F-measure of 0.08. This could be attributed to the context being significantly longer than the instruction. Models trained with instruction-tuning consistently prioritize the instruction, irrespective of the context's length. Conversely, models without instruction tuning might place greater emphasis on recent tokens, potentially neglecting the instruction.
LLaMA-30B achieves superior performance compared to **LLaMA-65B** despite its smaller size. We observe that in 56 instances, **LLaMA-65B** generates the beginning of its instruction (8.2), while **LLaMA-30B** does so in 20 cases. However, we lack an explanation for the discrepancy in complying with the instruction.
GPT-4 exhibits surprisingly lower performance than **ChatGPT**, despite its greater capability. We hypothesize that our dataset may lack the necessary expressive

power to effectively compare highly capable models. This limitation may arise from biases introduced by human labelers and the inherent complexity of the task. Consequently, it is possible that **GPT-4** performs better at cluster labeling than **ChatGPT**, but this difference might not be discernible using our dataset. Refer to Table 10.4 for non-cherry-picked example labels for both **ChatGPT** and **GPT-4**.
To address this issue in future experiments or when designing complex datasets, we suggest an approach where highly capable models are initially used to generate a dataset, which can then be refined by human labelers. This methodology can result in a less constrained and more comprehensive dataset, especially for tasks where obtaining high-quality training data is challenging.

Comparison to Old Approach The entries marked with "(old)" in the tables represent results obtained using prompts developed in our old approach (referred to as old prompts) as detailed in Section 8.2. Entries without this prefix are from the setup discussed in Chapter 6, referred to as new prompts. Old prompts exhibit higher precision compared to new prompts due to precision-focused optimization. With the exception of **T0++** and **GPT3.5**, old and new prompts yield similar F-measure results. The significant difference in recall for **GPT3.5** may stem from shorter labels generated in the new approach compared to the old one, despite the prompts' similarity. The reason for this drastic length difference remains unclear, but we speculate that updates to OpenAI's model and instruction-following behavior between the February 2023 early experiments and the May 2023 late experiments may have played a role.
The **T0++** model exhibits poor performance for both old and new prompts, marked by a dramatic difference in recall and precision, although the F-measure remains similar. We prefer the old prompt due to its notably higher precision. The lower recall is likely due to the concise nature of the labels generated by **T0++**, which effectively describes clusters. Refer to Table 10.4 for non-cherry-picked example labels for **T0++** with both the new and old prompts.

Table 10.2: Complete results of automatic evaluation via ROUGE in terms of ROUGE-1 (**R-1**), ROUGE-2 (**R-2**), ROUGE-LCS (**R-LCS**) for the cluster labeling task across all 19 LLMs. We compared them against the manually annotated reference (**Reference**; refer to Section 8.1) and the best performing model from our manual evaluation (**GPT3.5**; refer to Section 10.1.1). The top three models for each metric are highlighted, with notable strong performance from **ChatGPT** and **LLaMA-CoT**.

Model	Reference			GPT3.5		
	R-1	R-2	R-LCS	R-1	R-2	R-LCS
Alpaca-7B	13.89	3.10	12.65	19.98	6.08	18.05
Baize-13B	14.44	2.28	13.02	24.59	8.23	22.53
Baize-7B	17.40	2.88	14.95	26.35	9.43	23.89
BLOOM	11.71	2.13	10.80	11.84	2.76	10.96
BLOOM (old)	12.52	2.52	11.34	13.10	3.74	12.20
Falcon-40B	14.30	3.06	13.26	13.08	3.49	11.92
Falcon-40B-I	17.59	3.97[3]	15.48[3]	21.72	7.66	19.57
ChatGPT	**20.15**[1]	**4.88**[1]	**17.42**[1]	**29.52**[1]	10.99[2]	25.75[2]
GPT-4	16.43	2.84	14.42	27.76[3]	9.57[3]	24.80[3]
GPT-NeoX	13.13	3.17	12.22	16.38	5.01	15.29
GPT-NeoX (old)	12.93	2.37	11.72	11.67	2.41	10.72
LLaMA-30B	12.30	2.60	11.19	12.07	2.70	11.14
LLaMA-CoT	18.91[2]	4.50[2]	16.83[2]	28.94[2]	**11.69**[1]	**26.38**[1]
LLaMA-65B	10.25	1.93	9.40	10.81	2.49	9.95
OASST	18.28[3]	3.58	16.15	27.13	9.09	23.88
OPT-66B	13.59	2.90	11.96	13.45	2.76	12.26
OPT-66B (old)	11.67	2.68	10.87	10.56	2.17	9.55
Pythia	14.78	2.99	13.23	21.64	6.44	19.67
T0++	11.74	1.94	10.46	13.00	2.69	11.49
T0++ (old)	9.80	2.01	9.61	7.64	1.70	7.52
GPT3.5	8.61	1.25	8.19	22.46	10.24	21.68
GPT3.5 (old)	16.82	2.96	14.61	–	–	–
Vicuna-13B	16.90	3.02	14.81	25.32	8.66	22.66
Vicuna-7B	17.04	2.62	14.81	23.88	7.42	20.87

Table 10.3: Complete results of automatic evaluation via BERTScore precision (**P**), recall (**R**), and F-measure (**F1**) for the cluster labeling task across all 19 LLMs. We compared them against the manually annotated reference (**Reference**; refer to Section 8.1) and the best performing model from our manual evaluation (**GPT3.5**; refer to Section 10.1.1). The top three models are indicated for each metric. Similar to the ROUGE evaluation, we see a strong performance by **ChatGPT** and **LLaMA-CoT**. Also shown are the statistics of the length of the generated cluster labels (in number of tokens).

Model	Reference			GPT3.5			Length		
	P	**R**	**F1**	**P**	**R**	**F1**	**Min**	**Max**	**Mean**
Alpaca-7B	0.20	0.15	0.17	0.31	0.28	0.29	3	21	7.92
Baize-13B	0.17	0.15	0.16	0.33	0.32	0.32	1	39	8.47
Baize-7B	0.22[3]	0.19	0.20	0.38[3]	0.38	0.38[3]	2	46	10.73
BLOOM	0.11	0.07	0.08	0.20	0.19	0.19	1	51	8.81
BLOOM (old)	0.15	0.09	0.11	0.22	0.19	0.20	1	54	8.13
Falcon-40B	0.12	0.09	0.10	0.17	0.17	0.17	1	57	9.57
Falcon-40B-I	0.22[3]	0.18	0.20	0.34	0.32	0.33	2	33	9.34
ChatGPT	0.23[2]	**0.24**[1]	**0.23**[1]	0.39[2]	**0.43**[1]	**0.41**[1]	3	34	11.10
GPT-4	0.21	0.19	0.20	0.37	0.36	0.37	4	18	7.50
GPT-NeoX	0.15	0.09	0.12	0.27	0.23	0.24	1	34	9.41
GPT-NeoX (old)	0.19	0.07	0.12	0.24	0.17	0.20	1	34	7.42
LLaMA-30B	0.12	0.06	0.08	0.19	0.17	0.17	1	46	9.58
LLaMA-CoT	**0.24**[1]	0.21[2]	0.22[2]	**0.41**[1]	0.39[3]	0.40[2]	3	29	8.45
LLaMA-65B	0.08	0.02	0.05	0.14	0.14	0.14	1	46	10.27
OASST	0.22[3]	0.21[2]	0.21[3]	0.39[2]	0.40[2]	0.40[2]	3	31	10.15
OPT-66B	0.12	0.10	0.11	0.19	0.21	0.20	1	50	9.88
OPT-66B (old)	0.16	0.09	0.12	0.22	0.19	0.20	1	30	8.27
Pythia	0.19	0.13	0.16	0.31	0.27	0.29	2	34	7.69
T0++	0.03	0.08	0.05	0.11	0.21	0.16	1	57	13.31
T0++ (old)	0.15	0.00	0.06	0.15	0.03	0.09	1	18	3.10
GPT3.5	0.23[2]	0.06	0.14	0.49	0.31	0.40	1	13	4.18
GPT3.5 (old)	0.23[2]	0.20[3]	0.21[3]	–	–	–	3	27	9.44
Vicuna-13B	0.21	0.21[2]	0.21[3]	0.36	0.39[3]	0.37	3	39	11.87
Vicuna-7B	0.20	0.19	0.19	0.34	0.37	0.35	2	42	11.47

Table 10.4: Four examples for labels generated by **T0++ old**, **T0++ new**, **ChatGPT**, and **GPT-4**, together with the human generated **Reference**.

Reference	obese people in public transportation especially in an emergency situation
T0++ old	Obese people can block emergency exits
T0++ new	Obese people can impede the evacuation of a plane or bus in an emergency.
ChatGPT	Obese individuals may impede escape routes in emergency situations on public transport and airplanes.
GPT-4	Obesity and evacuation concerns in public transport
Reference	justice and punishment
T0++ old	Punishment
T0++ new	Perhaps not, but there has to be some sort of punishment in response to crimes. Punishment itself is not the only way to do that. It is also enforced through the threat of further punishment (often jail time). Those are hardly good punishments. We have evolved to this point in our
ChatGPT	Debate on the effectiveness and appropriateness of punishment for crimes.
GPT-4	Discussing punishment and its role in society
Reference	patriarchy and the dominance of male culture
T0++ old	Patriarchy
T0++ new	What you describe is not a patriarchy at all. I'm sorry that your feelings are hurt by referring to the generally male-oriented, historically male and catering-towards-males power system in the United States and many other countries as the "patriarchy", but I
ChatGPT	Debate on the definition and existence of patriarchy.
GPT-4	Society dominated by male-created institutions
Reference	golden rice is beneficial because it helps combating hunger and Vitamin A deficiency
T0++ old	Golden Rice
T0++ new	Anti-GMO scaremongering is not the only thing holding back Golden Rice. Greenpeace is not solely responsible for Golden Rice's lack of proliferation. Perhaps you can print a reasonable reason to block Golden Rice. If you can't, intellectual honesty should compel you to say
ChatGPT	Debate on the introduction of Golden Rice to combat Vitamin A deficiencies in poor/rural countries.
GPT-4	Discussing pros and cons of Golden Rice and its impact on vitamin A deficiency.

10.2 Frame Assignment

In this section we compare all 19 LLMs for frame assignment based on the setup detailed in Chapter 9 and the evaluation dataset described in Section 9.1. Table 10.5 presents the performance results. In our interpretation of these results we focus solely on the frequency with which the first generated frame (**T1**) corresponds to one of the manually assigned frames.

Pre-ChatGPT Models The older models, such as **BLOOM**, **GPT-NeoX**, and **OPT-66B**, which were only pre-trained, exhibit notably inferior performance compared to other models. Their subpar performance largely stems from their limited understanding of or disregard for the instructions. Notably, **OPT-66B** frequently ignores the instructions and generates media frames in the given order.

OpenAI OpenAI models, including **GPT3.5**, **ChatGPT**, and **GPT-4**, consistently achieve the highest rankings across various frame contexts, except in the few-shot scenario, where **LLaMA-CoT** surpasses **GPT3.5** to claim third place. Additionally, **GPT-4** secures the top ranking in all scenarios, except for the "Zero-Shot (short)" scenario, where it is outperformed by **GPT3.5** by a margin of 0.4 percentage points.

Longer Training **LLaMA-30B**, **LLaMA-65B**, and **Falcon-40B** consistently exhibit comparable or superior performance compared to models that underwent instruction-tuning. This observation indicates that extended pre-training significantly enhances the models' capacity to comprehend and adhere to instructions.

Few-Shot Many models exhibit a decrease in performance as more frame context is introduced, with few-shot scenarios having a particularly detrimental effect on most models. We propose that this decline can be attributed to the models' reliance on instruction-tuning, where they learn from short demonstrations provided by more capable models like **ChatGPT**. These demonstrations are typically brief, making it challenging for the models to comprehend longer few-shot contexts. Notably, **ChatGPT**, **GPT-4**, **LLaMA-CoT**, and **GPT-NeoX** stand out as exceptions, performing well with few-shot contexts. We posit that, in the case of **ChatGPT** and **GPT-4**, their ability to understand instructions more broadly, acquired through RLHF training, enables them to benefit from the longer examples.

RLHF **ChatGPT**, **GPT-4**, **GPT3.5**, **OASST**, and **Pythia** have undergone training using RLHF. These models consistently maintain stable performance across various frame contexts, and notably, their performance in the "Few-Shot" scenario is on par with or even surpasses that in the "Zero-Shot (extreme)" scenario. This underscores the effectiveness of RLHF in rendering the model performance robust to variations in wording choices and instruction length.

T0 **T0++** achieves the fourth rank in both the "Zero-Shot (short)" and "Zero-Shot (long)" scenarios, as well as the fifth rank in the "Few-Shot" scenario. Notably, it outperforms some models trained with newer more promising approaches like RLHF. This achievement is particularly remarkable considering that **T0++** is one of the oldest models in the evaluation, and its training process significantly differs from other approaches.

Table 10.5: Complete results of automatic evaluation for the frame assignment task. The results depict the percentage of instances in which a model's first (**T1**), second (**T2**), and third (**T3**) predicted frames match one of the reference frames. Absent values indicate model inferences that exceeded our computational resources.

Model	Zero-Shot									Few-Shot		
	extreme			short			long					
	T1	**T2**	**T3**	**T1**	**T2**	**T3**	**T1**	**T2**	**T3**	**T1**	**T2**	**T3**
`Alpaca-7B`	39.1	53.9	64.2	39.5	51.0	64.6	28.4	37.4	57.2	20.6	26.7	49.4
`BLOOM`	26.7	46.5	53.5	31.7	52.7	57.6	25.5	51.9	60.1	–	–	–
`Baize-13B`	42.4	53.5	58.4	48.1	59.3	63.4	42.0	53.5	60.5	39.5	46.5	49.4
`Baize-7B`	34.2	44.4	52.7	34.6	46.9	56.8	39.1	46.5	53.9	30.9	38.3	45.7
`Falcon-40B`	46.5	68.3	72.0	46.5	67.5	75.7	46.1	56.8	64.2	38.3	53.5	68.3
`Falcon-40B-I`	51.4[5]	64.6	72.8	44.4	56.4	68.3	32.9	44.9	57.6	28.4	49.4	63.8
`ChatGPT`	60.9[2]	76.1	86.4	58.0[3]	78.6	88.5	58.8[2]	76.1	84.8	63.4[2]	80.2	90.1
`GPT-4`	**63.4**[1]	82.3	91.8	60.5[2]	84.4	90.1	**65.4**[1]	83.1	90.5	**67.1**[1]	84.8	88.5
`GPT-NeoX`	19.3	28.4	50.6	25.1	31.3	51.9	31.3	36.6	50.2	31.3	39.5	49.0
`LLaMA-30B`	45.7	63.0	70.8	41.2	57.2	65.4	39.1	58.0	66.3	40.7	70.0	77.8
`LLaMA-CoT`	46.9	73.3	84.0	54.3[4]	75.7	85.6	49.8	71.2	82.3	57.2[3]	70.0	77.0
`LLaMA-65B`	53.1[4]	65.4	81.9	50.6	70.8	82.3	39.5	64.6	78.6	–	–	–
`OASST`	48.6	72.8	82.3	48.1	66.3	76.5	53.5[5]	73.7	82.7	47.7	65.0	79.8
`OPT-66B`	16.0	18.9	43.2	13.2	16.5	45.3	14.8	18.1	45.7	–	–	–
`Pythia`	31.7	44.0	52.3	33.3	43.6	49.4	30.5	39.1	44.9	29.6	34.2	38.7
`T0++`	48.6	58.4	64.2	54.3[4]	60.1	65.4	55.6[4]	59.7	63.8	49.8[5]	52.3	53.5
`GPT3.5`	53.5[3]	74.1	81.9	**60.9**[1]	65.4	66.7	58.0[3]	58.8	59.7	53.9[4]	57.6	58.0
`Vicuna-13B`	44.0	52.7	62.1	40.7	55.1	67.1	42.0	53.1	64.6	38.3	50.2	60.1
`Vicuna-7B`	28.4	34.6	50.2	36.2	48.1	61.3	35.4	42.8	55.1	20.2	24.3	46.1

Chapter 11

Conclusion and Future Work

In this thesis, we investigated the application of LLMs in uncovering frames within argumentative discussions. Our approach consists of interchangeable components that can adapt to SOTA techniques, ensuring scalability with evolving insights.

We discover frames by clustering sentences, aligning with Entman's [32] "selection and salience" definition, as speakers select and encapsulate aspects into sentences. Aspects recurring across sentences and their connection within the same sentences boost salience, resulting in larger, more significant clusters. Exploring different units for clustering (e.g., phrases, paragraphs, or entire comments) remains an open question for frame discovery.

LLMs that were only pre-trained exhibited inadequate performance, while those subjected to instruction-tuning demonstrated notable proficiency in both frame generation and identification. Further enhancement in labeling performance may be achievable through alternative cluster content presentations for frame generation.

Frame generation and identification pose inherent challenges due to the subjective nature of salience, varying among individuals. This challenge became evident during the development of our evaluation datasets. We anticipate that advanced LLMs can help mitigate these limitations by generating a diverse set of candidate labels, allowing human labelers to choose the most suitable ones.

We also recognize the potential of LLMs in aiding audiences in comprehending complex topics and offering comprehensive perspectives of objective reality. Presenting multiple interpretations of events could lead to more informed decision-making and empower individuals. We look forward to future efforts in establishing systems that strike a balance between selection and omission while delivering easily understandable abstractions of reality.

Appendix A

Keyphrase Extraction

We experimented with extracting keyphrases from the text, which are short phrases that convey crucial information, to improve label generation. We utilized two methods for keyphrase extraction: the SOTA approach KeyBERT [37], and a custom clustering-based method.

A.1 Keyphrase Extraction Methods

A.1.1 KeyBERT

KeyBERT ranks keyphrases extracted from a document according to the cosine similarity between the embeddings of the keyphrases and the embedding of the document.
To extract keyphrases, we utilize KeyphraseVectorizers [82], a keyphrase extractor that identifies grammatically accurate keyphrases by considering their part-of-speech tags. Additionally, we configured KeyBERT with Sentence-BERT to create embeddings.

A.1.2 Clustering-based Keyphrase Extraction

We developed a keyphrase extraction approach, which we call "Cluster", tailored for encoder models that utilize mean pooling, such as Sentence-BERT, that it is specifically designed to function in conjunction with sentence clustering, as demonstrated in Chapter 7.

Candidate Extraction We observe that, during the mean pooling process, contextualized word embeddings with higher magnitudes exert a greater influence on the resulting sentence embedding. Consequently, our keyphrase

extraction method initiates by selecting tokens associated with contextualized word embeddings of high magnitudes. Subsequently, we expand these initial keyphrases by incorporating neighboring tokens that also possess high-magnitude embeddings.

Candidate Ranking Given a clustering of sentences denoted as S, we apply the sentence clustering approach with a `minimum_cluster_size` of 30 and the `cluster_selection_method` "eom" to create a clustering W of contextualized word embeddings for all tokens found in S.
We calculate the likelihood of encountering members of a token embedding cluster W_m within a sentence cluster S_n compared to random chance using Pointwise Mutual Information (PMI) as described in Equation A.1.

$$\mathrm{PMI}(x, y) = \log_2 \frac{p(x, y)}{p(x) \cdot p(y)} \tag{A.1}$$

The PMI score $\mathrm{PMI}(S_n, W_m)$ quantifies the significance of the word meaning W_m within the context of cluster S_n in comparison to other sentence clusters. Each token in a sentence is assigned the PMI score corresponding to its cluster in W and the cluster of its sentence in S, and we rank keyphrases based on the mean PMI score of their constituent tokens.

A.1.3 Comparison

Table A.1 displays the results of our comparison between KeyBERT and Cluster, where we extracted 10 keyphrases for 50 clusters and manually determined which ranking is preferable. In our initial evaluation, Cluster was the clear winner, however, there was a complaint that KeyBERT generated keyphrases that were too similar.
To address this issue, we employed Maximal Marginal Relevance (MMR), as defined in Equation A.2, where λ denotes the diversity and T represents the set of already selected keyphrases.

$$\mathrm{MMR}_\lambda(k) \coloneqq (1 - \lambda) \cdot \mathrm{score}(k) + \lambda \cdot (1 - \max_{t \in T} (\mathrm{sim}(k, t))) \tag{A.2}$$

MMR prioritizes keyphrases that differ from those already selected, leading to an increase in diversity. In the subsequent evaluation, KeyBERT emerged as the superior method. As a result, we retained both extraction methods for further experimentation.

Table A.1: Comparison of the preference frequency for keyphrase rankings generated by Cluster and KeyBERT in our evaluation.

Approach	Without MMR	With $MMR_{0.5}$
Cluster	34	19
KeyBERT	14	29

Table A.2: Average BERTScore precision values for different prompts including prompts employing keyphrases.

Prompt	Mean	95% CI	#Errors
question answering	0.152	[0.122, 0.182]	14
prefix-postfix	0.112	[0.085, 0.140]	10
concepts+text (keybert)	0.103	[0.076, 0.130]	4
prefix	0.098	[0.068, 0.128]	14
concepts (keybert)	0.087	[0.064, 0.110]	0
postfix	0.069	[0.037, 0.102]	31
concepts+text (cluster)	0.067	[0.038, 0.096]	4
explicit	0.033	[-0.001, 0.066]	38
concepts (cluster)	0.024	[-0.002, 0.049]	0
short	-0.072	[-0.105, -0.040]	60

A.2 T0++

We conducted additional experiments with **T0++** using keyphrase extraction methods.

A.2.1 Prompts

We created two new prompts incorporating keyphrases.

concepts Prompt A.1 presents the top extracted keyphrases to the model without presenting the text.

```
The following concepts were extracted from a {input_type}: "{concepts}"
What is the {output_type} of this {input_type}?
```

Prompt A.1: concepts

concepts+text Prompt A.2 provides the top keyphrases along with the corresponding text from which they were extracted to the model. We hypothesize

that this approach may assist the model in concentrating on the essential aspects of the text.

```
Given the following concepts: "{concepts}", what {output_type} would
↪  you choose for the {input_type} below?
{text}
```

Prompt A.2: concepts+text

Table A.2 displays the experiments conducted, which include those from Table 8.4 as well as the two newly developed prompts for both keyphrase extraction approaches. The results suggest that providing keyphrases does not yield a discernible benefit.

A.2.2 Sentence Selection

We investigated whether using keyphrases as a basis for sentence selection might improve label generation. We consider two methods: one where all sentences were ordered by their centrality in the cluster known as "persistence", and another where only sentences containing one of the top 10 highest-ranked keyphrases, extracted using KeyBERT, were included, referred to as "keyphrase".
Table A.3 demonstrates that the "keyphrase" method did not outperform the "persistence" method. Since the keyphrase approach added complexity without showing superior performance, we opted not to further explore its benefits in our research.

Table A.3: Average BERTScore precision values for different sentence selection methods.

Sentence Selection Method	Mean	95% CI	#Errors
persistence	0.185	[0.156, 0.214]	5
keyphrase	0.132	[0.100, 0.163]	1

Appendix B

Media Frames

Here, we present the 14 generic media frames introduced by Boydstun et al. [16, pp. 6–7]. Centered below the definitions from Boydstun et al. we also include the concise defintions from Card et al. [21, p. 2].

Economic The costs, benefits, or monetary/financial implications of the issue (to an individual, family, community or to the economy as a whole).

costs, benefits, or other financial implications

Capacity and Resources The lack of or availability of physical, geographical, spatial, human, and financial resources, or the capacity of existing systems and resources to implement or carry out policy goals.

availability of physical, human or financial resources, and
capacity of current systems

Morality Any perspective or policy objective or action (including proposed action) that is compelled by religious doctrine or interpretation, duty, honor, righteousness or any other sense of ethics or social responsibility.

religious or ethical implications

Fairness and Equality Equality or inequality with which laws, punishment, rewards, and resources are applied or distributed among individuals or groups. Also the balance between the rights or interests of one individual or group compared to another individual or group.

balance or distribution of rights, responsibilities, and
resources

Legality, Constitutionality and Jurisprudence The constraints imposed on or freedoms granted to individuals, government, and corporations via the Constitution, Bill of Rights and other amendments, or judicial interpretation. This deals specifically with the authority of government to regulate, and the authority of individuals/corporations to act independently of government.

rights, freedoms, and authority of individuals, corporations, and government

Policy Prescription and Evaluation Particular policies proposed for addressing an identified problem, and figuring out if certain policies will work, or if existing policies are effective.

discussion of specific policies aimed at addressing problems

Crime and Punishment Specific policies in practice and their enforcement, incentives, and implications. Includes stories about enforcement and interpretation of laws by individuals and law enforcement, breaking laws, loopholes, fines, sentencing and punishment. Increases or reductions in crime.

effectiveness and implications of laws and their enforcement

Security and Defense Security, threats to security, and protection of one's person, family, in-group, nation, etc. Generally an action or a call to action that can be taken to protect the welfare of a person, group, nation sometimes from a not yet manifested threat.

threats to welfare of the individual, community, or nation

Health and Safety Healthcare access and effectiveness, illness, disease, sanitation, obesity, mental health effects, prevention of or perpetuation of gun violence, infrastructure and building safety.

health care, sanitation, public safety

Quality of Life The effects of a policy, an individual's actions or decisions, on individuals' wealth, mobility, access to resources, happiness, social structures, ease of day-to-day routines, quality of community life, etc.

threats and opportunities for the individual's wealth, happiness, and well-being

Cultural Identity The social norms, trends, values and customs constituting culture(s), as they relate to a specific policy issue.

traditions, customs, or values of a social group in relation to a policy issue

Public Opinion References to general social attitudes, polling and demographic information, as well as implied or actual consequences of diverging from or getting ahead of public opinion or polls.

attitudes and opinions of the general public, including polling and demographics

Political Any political considerations surrounding an issue. Issue actions or efforts or stances that are political, such as partisan filibusters, lobbyist involvement, bipartisan efforts, deal-making and vote trading, appealing to one's base, mentions of political maneuvering. Explicit statements that a policy issue is good or bad for a particular political party.

considerations related to politics and politicians, including lobbying, elections, and attempts to sway voters

External Regulation and Reputation A country's external relations with another nation; the external relations of one state with another; or relations between groups. This includes trade agreements and outcomes, comparisons of policy outcomes or desired policy outcomes.

international reputation or foreign policy of the U.S.

Other Any frames that do not fit into the above categories.

Appendix C

Frame Prompts

C.1 Zero-Shot (extreme)

```
[
  "economic",
  "capacity and resources",
  "morality",
  "fairness and equality",
  "legality, constitutionality and jurisprudence",
  "policy prescription and evaluation",
  "crime and punishment",
  "security and defense",
  "health and safety",
  "quality of life",
  "cultural identity",
  "public opinion",
  "political",
  "external regulation and reputation"
]
```

C.2 Zero-Shot (short)

```
{
  "economic": {
    "description": "costs, benefits, or other financial implications"
  },
  "capacity and resources": {
    "description": "availability of physical, human or financial
    ↪  resources, and capacity of current systems"
  },
```

```
"morality": { "description": "religious or ethical implications" },
"fairness and equality": {
  "description": "balance or distribution of rights, responsibilities,
  ↪  and resources"
},
"legality, constitutionality and jurisprudence": {
  "description": "rights, freedoms, and authority of individuals,
  ↪  corporations, and government"
},
"policy prescription and evaluation": {
  "description": "discussion of specific policies aimed at addressing
  ↪  problems"
},
"crime and punishment": {
  "description": "effectiveness and implications of laws and their
  ↪  enforcement"
},
"security and defense": {
  "description": "threats to welfare of the individual, community, or
  ↪  nation"
},
"health and safety": {
  "description": "health care, sanitation, public safety"
},
"quality of life": {
  "description": "threats and opportunities for the individual's
  ↪  wealth, happiness, and well-being"
},
"cultural identity": {
  "description": "traditions, customs, or values of a social group in
  ↪  relation to a policy issue"
},
"public opinion": {
  "description": "attitudes and opinions of the general public,
  ↪  including polling and demographics"
},
"political": {
  "description": "considerations related to politics and politicians,
  ↪  including lobbying, elections, and attempts to sway voters"
},
"external regulation and reputation": {
  "description": "international reputation or foreign policy of the
  ↪  U.S."
}
```

```
}
```

C.3 Zero-Shot (long)

```
{
  "economic": {
    "description": "The costs, benefits, or monetary/financial
    ↪  implications of the issue (to an individual, family, community,
    ↪  or to the economy as a whole)."
  },
  "capacity and resources": {
    "description": "The lack of or availability of physical,
    ↪  geographical, spatial, human, and financial resources, or the
    ↪  capacity of existing systems and resources to implement or carry
    ↪  out policy goals."
  },
  "morality": {
    "description": "Any perspective or policy objective or action
    ↪  (including proposed action) that is compelled by religious
    ↪  doctrine or interpretation, duty, honor, righteousness or any
    ↪  other sense of ethics or social responsibility."
  },
  "fairness and equality": {
    "description": "Equality or inequality with which laws, punishment,
    ↪  rewards, and resources are applied or distributed among
    ↪  individuals or groups. Also the balance between the rights or
    ↪  interests of one individual or group compared to another
    ↪  individual or group."
  },
  "legality, constitutionality and jurisprudence": {
    "description": "The constraints imposed on or freedoms granted to
    ↪  individuals, government, and corporations via the Constitution,
    ↪  Bill of Rights and other amendments, or judicial interpretation.
    ↪  This deals specifically with the authority of government to
    ↪  regulate, and the authority of individuals/corporations to act
    ↪  independently of government."
  },
  "policy prescription and evaluation": {
    "description": "Particular policies proposed for addressing an
    ↪  identified problem, and figuring out if certain policies will
    ↪  work, or if existing policies are effective."
  },
  "crime and punishment": {
```

```
    "description": "Specific policies in practice and their enforcement,
    ↪  incentives, and implications. Includes stories about enforcement
    ↪  and interpretation of laws by individuals and law enforcement,
    ↪  breaking laws, loopholes, fines, sentencing and punishment.
    ↪  Increases or reductions in crime."
  },
  "security and defense": {
    "description": "Security, threats to security, and protection of
    ↪  one's person, family, in-group, nation, etc. Generally an action
    ↪  or a call to action that can be taken to protect the welfare of a
    ↪  person, group, nation sometimes from a not yet manifested
    ↪  threat."
  },
  "health and safety": {
    "description": "Healthcare access and effectiveness, illness,
    ↪  disease, sanitation, obesity, mental health effects, prevention
    ↪  of or perpetuation of gun violence, infrastructure and building
    ↪  safety."
  },
  "quality of life": {
    "description": "The effects of a policy on individuals' wealth,
    ↪  mobility, access to resources, happiness, social structures, ease
    ↪  of day-to-day routines, quality of community life, etc."
  },
  "cultural identity": {
    "description": "The social norms, trends, values and customs
    ↪  constituting culture(s), as they relate to a specific policy
    ↪  issue."
  },
  "public opinion": {
    "description": "References to general social attitudes, polling and
    ↪  demographic information, as well as implied or actual
    ↪  consequences of diverging from or \"getting ahead of\" public
    ↪  opinion or polls."
  },
  "political": {
    "description": "Any political considerations surrounding an issue.
    ↪  Issue actions or efforts or stances that are political, such as
    ↪  partisan filibusters, lobbyist involvement, bipartisan efforts,
    ↪  deal-making and vote trading, appealing to one's base, mentions
    ↪  of political maneuvering. Explicit statements that a policy issue
    ↪  is good or bad for a particular political party."
  },
  "external regulation and reputation": {
```

```
    "description": "The United States' external relations with another
    ↪  nation; the external relations of one state with another; or
    ↪  relations between groups. This includes trade agreements and
    ↪  outcomes, comparisons of policy outcomes or desired policy
    ↪  outcomes."
  }
}
```

C.4 Few-Shot

```
{
  "economic": {
    "description": "The costs, benefits, or monetary/financial
    ↪  implications of the issue (to an individual, family, community,
    ↪  or to the economy as a whole).",
    "examples": [
      "Necessity of minimum wage laws and their effects on the labor
      ↪  market.",
      "Consequences of unregulated capitalism and the potential of a
      ↪  libertarian society.",
      "Risk-based insurance premiums determined by complex modeling of
      ↪  probability and cost factors."
    ]
  },
  "capacity and resources": {
    "description": "The lack of or availability of physical,
    ↪  geographical, spatial, human, and financial resources, or the
    ↪  capacity of existing systems and resources to implement or carry
    ↪  out policy goals.",
    "examples": [
      "Potential of biofuels as an alternative to fossil fuels.",
      "Physical fitness tests measure upper body strength and running
      ↪  ability for military service.",
      "Physical strength and endurance needed for modern combat."
    ]
  },
  "morality": {
    "description": "Any perspective or policy objective or action
    ↪  (including proposed action) that is compelled by religious
    ↪  doctrine or interpretation, duty, honor, righteousness or any
    ↪  other sense of ethics or social responsibility.",
    "examples": [
      "Fighting for the weak and vulnerable despite the odds.",
```

```
    "Victim-blaming debate on police brutality.",
    "Potential corruption of some native canadian bands and the need
    ↪  for transparency."
  ]
},
"fairness and equality": {
  "description": "Equality or inequality with which laws, punishment,
  ↪  rewards, and resources are applied or distributed among
  ↪  individuals or groups. Also the balance between the rights or
  ↪  interests of one individual or group compared to another
  ↪  individual or group.",
  "examples": [
    "Differences between humanism and feminism and their respective
    ↪  goals.",
    "Disparities in scholarship opportunities for minority students.",
    "Violent suppression of native american populations for centuries
    ↪  leading to a lack of advocacy and rights."
  ]
},
"legality, constitutionality and jurisprudence": {
  "description": "The constraints imposed on or freedoms granted to
  ↪  individuals, government, and corporations via the Constitution,
  ↪  Bill of Rights and other amendments, or judicial interpretation.
  ↪  This deals specifically with the authority of government to
  ↪  regulate, and the authority of individuals/corporations to act
  ↪  independently of government.",
  "examples": [
    "Guns acquired through legal and illegal channels for criminal
    ↪  use.",
    "Importance of the 2nd amendment and the implications of gun
    ↪  ownership in a democracy.",
    "Relevance of sexual history in rape cases."
  ]
},
"policy prescription and evaluation": {
  "description": "Particular policies proposed for addressing an
  ↪  identified problem, and figuring out if certain policies will
  ↪  work, or if existing policies are effective.",
  "examples": [
    "Religious scientists making major contributions to the world
    ↪  despite majority of scientists being agnostic atheists.",
    "Pros and cons of voluntary registration.",
    "Collective ownership of production for the betterment of society,
    ↪  with workers profiting from the sale of their labor."
```

```
    ]
  },
  "crime and punishment": {
    "description": "Specific policies in practice and their enforcement,
    ↪  incentives, and implications. Includes stories about enforcement
    ↪  and interpretation of laws by individuals and law enforcement,
    ↪  breaking laws, loopholes, fines, sentencing and punishment.
    ↪  Increases or reductions in crime.",
    "examples": [
      "Complexities of police shootings and race.",
      "Men are more likely to commit violent crimes than women.",
      "Punishment as a response to crime debated, with consideration of
      ↪  morality, severity, and aims."
    ]
  },
  "security and defense": {
    "description": "Security, threats to security, and protection of
    ↪  one's person, family, in-group, nation, etc. Generally an action
    ↪  or a call to action that can be taken to protect the welfare of a
    ↪  person, group, nation sometimes from a not yet manifested
    ↪  threat.",
    "examples": [
      "Protective physical self-defense in a fight.",
      "Powerful military technology making infantry obsolete in war.",
     "Protection of infants and mentally disabled through social policy."
    ]
  },
  "health and safety": {
    "description": "Healthcare access and effectiveness, illness,
    ↪  disease, sanitation, obesity, mental health effects, prevention
    ↪  of or perpetuation of gun violence, infrastructure and building
    ↪  safety.",
    "examples": [
      "Complexities of food choices and their effects on health.",
      "Potentially fatal consequences of taking too much acetaminophen.",
      "Encouraging healthy habits without shaming or pressuring people to
      ↪  lose weight."
    ]
  },
  "quality of life": {
    "description": "The effects of a policy on individuals' wealth,
    ↪  mobility, access to resources, happiness, social structures, ease
    ↪  of day-to-day routines, quality of community life, etc.",
    "examples": [
```

```
    "Differences between adults and children in terms of understanding
    ↪  and perception.",
    "Importance of extracurriculars and academics for college
    ↪  admissions.",
    "Appropriate times to yell at customer service workers."
  ]
},
"cultural identity": {
  "description": "The social norms, trends, values and customs
  ↪  constituting culture(s), as they relate to a specific policy
  ↪  issue.",
  "examples": [
    "Rapid shift in acceptance of homosexuality in the u.s.",
    "Collective action necessary for social progress and change.",
    "Complexities of gender identity and expression."
  ]
},
"public opinion": {
  "description": "References to general social attitudes, polling and
  ↪  demographic information, as well as implied or actual
  ↪  consequences of diverging from or \"getting ahead of\" public
  ↪  opinion or polls.",
  "examples": [
    "Gender roles and expectations are socially constructed and
    ↪  changing.",
    "Pros and cons of the 40-hour work week.",
    "Potential appeal of a political candidate."
  ]
},
"political": {
  "description": "Any political considerations surrounding an issue.
  ↪  Issue actions or efforts or stances that are political, such as
  ↪  partisan filibusters, lobbyist involvement, bipartisan efforts,
  ↪  deal-making and vote trading, appealing to one's base, mentions
  ↪  of political maneuvering. Explicit statements that a policy issue
  ↪  is good or bad for a particular political party.",
  "examples": [
    "Differences between right-wing and left-wing politics.",
    "Complexities of anarchy.",
    "Power struggle between branches of government."
  ]
},
"external regulation and reputation": {
```

```
    "description": "The United States' external relations with another
    ↪  nation; the external relations of one state with another; or
    ↪  relations between groups. This includes trade agreements and
    ↪  outcomes, comparisons of policy outcomes or desired policy
    ↪  outcomes.",
    "examples": [
      "Implications of us involvement in nato and its allies.",
      "Potential consequences of us intervention in ukraine.",
      "Conflicting opinions on us involvement in foreign affairs."
    ]
  }
}
```

C.5 Human Evaluation

C.5.1 Instruction

```
How similar are the small phrases to the reference phrase? Drag and drop
↪  the boxes with the phrases on the left and bring them in your
↪  preferred order on the right. The most preferred phrase is on the top
↪  and the less you prefer a phrase, the lower it should be in the
↪  ranking.

Similarity is less in a sense of exact meaning but much rather in a
↪  meaning of is there some relation between the reference and
↪  hypotheses.

To get a better understanding of the meaning of the reference, the title
↪  of the original discussion and some central sentences from the
↪  cluster are provided (click the "show cluster" button next to the
↪  reference). The central sentences are selected based on how central
↪  they are in the original cluster and their mean similarity to the
↪  reference and hypotheses. So these are not perfectly representative
↪  to the cluster, but they can help you to get a better understanding
↪  of some hard to understand meanings.

Recommended Strategy for judging:
  The relation between the reference and hypotheses is understandable:
    only read the reference and the hypotheses
  The reference is a bit weird:
    read the title to get a better idea in what context the reference is
    ↪  used
  The hypotheses are hard to understand:
```

```
    read the central sentences from the cluster for more context
  The relation between the reference and hypotheses are not clear:
    read the central and random sentences from the cluster

We are looking for a label that describes the content of a cluster of
↪  sentences very well. It is important to understand that the reference
↪  is not the perfect label but much rather something that is strongly
↪  related to the perfect label.

When a lot of hypotheses are talking about something but the reference is
↪  not mentioning this specific thing, it can be a sign, that the
↪  reference might not be complete. In that case it might be sensible to
↪  update the reference with this specific thing (in your head).

Example:
  Reference: responsibilities between employee and employer
  A lot of hypotheses mentioning: the service industry
  New reference: responsibilities between employee and employer in the
  ↪  service industry

In the end we are looking for the central meaning of the cluster and it
↪  is very likely that at least one model got the central meaning right
↪  and the task is to guess what model got the central meaning best
↪  based on what the reference suggests the best central meaning is.
```

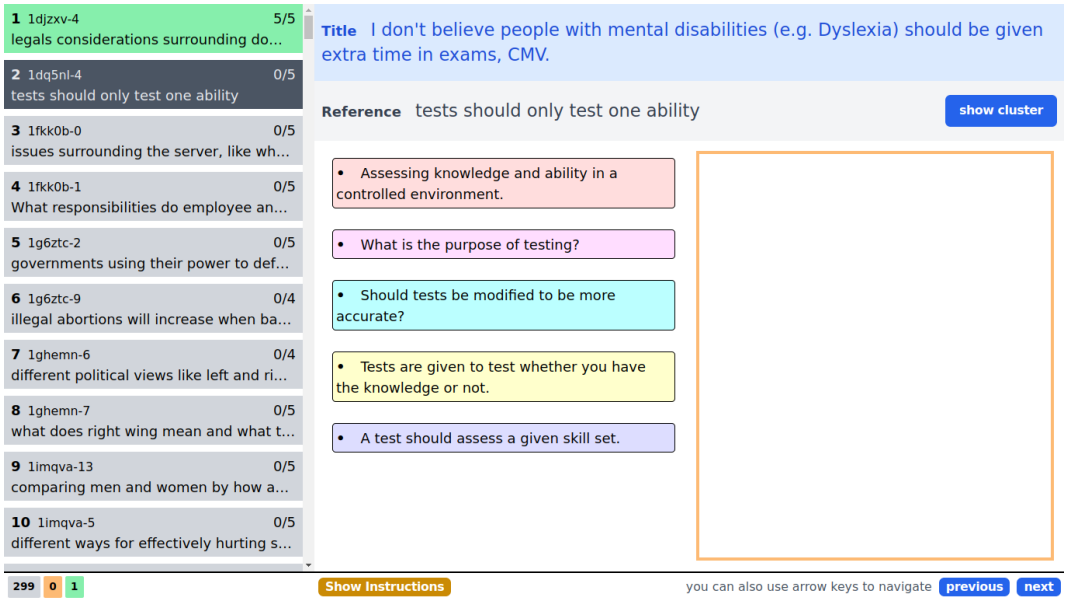


Figure C.1: The interface before ranking. The order of the sentences and the their colors are assigned randomly.

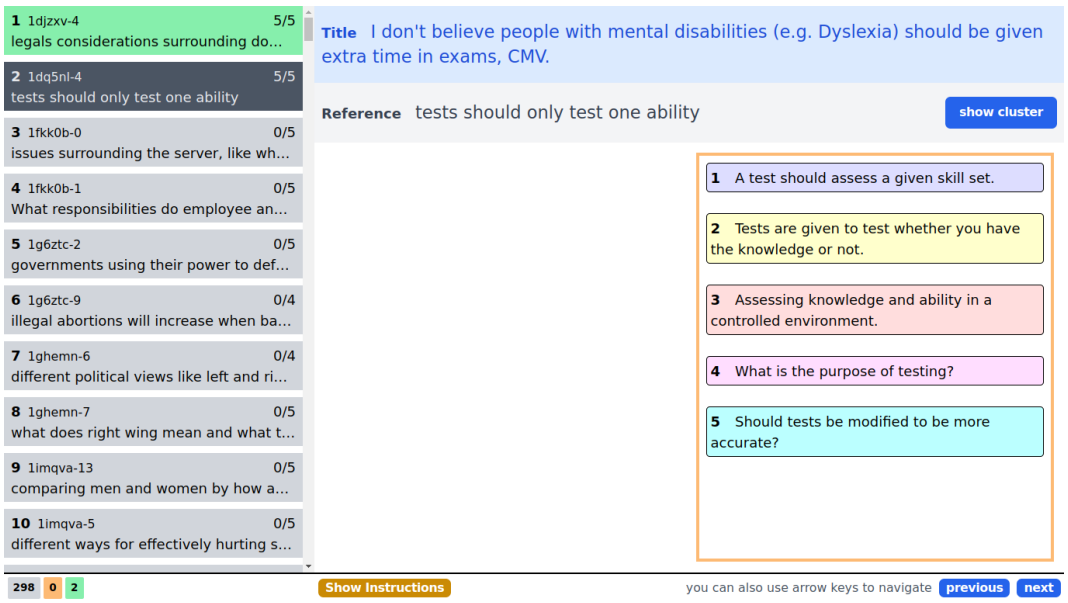


Figure C.2: The interface after ranking.

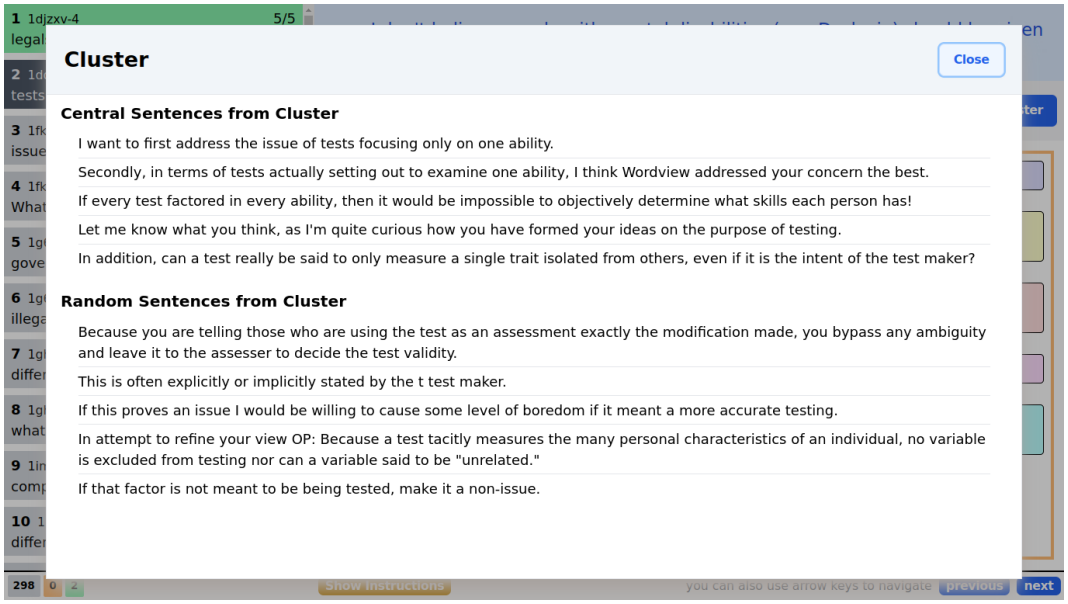


Figure C.3: Example sentences from the cluster. Shown are 5 sentences that are most similar to the reference and 5 random sentences from the cluster.

Bibliography

[1] Yamen Ajjour et al. "Modeling Frames in Argumentation". 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 2019, Hong Kong, China, November 3-7, 2019*. Ed. by Kentaro Inui et al. Association for Computational Linguistics, 2019, pp. 2922–2932. DOI: 10.18653/v1/D19-1290. URL: https://doi.org/10.18653/v1/D19-1290 (cit. on p. 6).

[2] Nikolaos Aletras and Mark Stevenson. "Labelling Topics using Unsupervised Graph-based Methods". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 2: Short Papers*. The Association for Computer Linguistics, 2014, pp. 631–636. DOI: 10.3115/v1/p14-2103. URL: https://doi.org/10.3115/v1/p14-2103 (cit. on p. 9).

[3] Mebarka Allaoui, Mohammed Lamine Kherfi, and Abdelhakim Cheriet. "Considerably Improving Clustering Algorithms Using UMAP Dimensionality Reduction Technique: A Comparative Study". In: *Image and Signal Processing - 9th International Conference, ICISP 2020, Marrakesh, Morocco, June 4-6, 2020, Proceedings*. Ed. by Abderrahim El Moataz et al. Vol. 12119. Lecture Notes in Computer Science. Springer, 2020, pp. 317–325. DOI: 10.1007/978-3-030-51935-3_34. URL: https://doi.org/10.1007/978-3-030-51935-3%5C_34 (cit. on p. 38).

[4] Ebtesam Almazrouei et al. "Falcon-40B: an open large language model with state-of-the-art performance". In: (2023) (cit. on p. 29).

[5] Dimo Angelov. "Top2Vec: Distributed Representations of Topics". In: *CoRR* abs/2008.09470 (2020). arXiv: 2008.09470. URL: https://arxiv.org/abs/2008.09470 (cit. on p. 37).

[6] *ausboss/llama-30b-supercot*. https://huggingface.co/ausboss/llama-30b-supercot. 2023 (cit. on p. 29).

[7] Stephen H. Bach et al. *PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts*. 2022. arXiv: 2202.01279 [cs.LG] (cit. on pp. 25, 49).

[8] 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 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1409.0473 (cit. on p. 6).

[9] Edward Beeching et al. *Open LLM Leaderboard*. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard. 2023 (cit. on pp. 26, 29).

[10] Shraey Bhatia, Jey Han Lau, and Timothy Baldwin. "Automatic Labelling of Topics with Neural Embeddings". In: *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*. Ed. by Nicoletta Calzolari, Yuji Matsumoto, and Rashmi Prasad. ACL, 2016, pp. 953–963. URL: https://aclanthology.org/C16-1091/ (cit. on p. 9).

[11] Federico Bianchi, Silvia Terragni, and Dirk Hovy. "Pre-training is a Hot Topic: Contextualized Document Embeddings Improve Topic Coherence". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*. Ed. by Chengqing Zong et al. Association for Computational Linguistics, 2021, pp. 759–766. DOI: 10.18653/v1/2021.acl-short.96. URL: https://doi.org/10.18653/v1/2021.acl-short.96 (cit. on p. 37).

[12] Stella Biderman et al. "Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling". In: *CoRR* abs/2304.01373 (2023). DOI: 10.48550/arXiv.2304.01373. arXiv: 2304.01373. URL: https://doi.org/10.48550/arXiv.2304.01373 (cit. on p. 27).

[13] Sid Black et al. "GPT-NeoX-20B: An Open-Source Autoregressive Language Model". In: *CoRR* abs/2204.06745 (2022). DOI: 10.48550/arXiv.2204.06745. arXiv: 2204.06745. URL: https://doi.org/10.48550/arXiv.2204.06745 (cit. on p. 25).

[14] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation". In: *J. Mach. Learn. Res.* 3 (2003), pp. 993–1022. URL: http://jmlr.org/papers/v3/blei03a.html (cit. on p. 6).

[15] Filip Boltuzic and Jan Snajder. "Identifying Prominent Arguments in Online Debates Using Semantic Textual Similarity". In: *Proceedings of the 2nd Workshop on Argumentation Mining, ArgMining@HLT-NAACL 2015, June 4, 2015, Denver, Colorado, USA*. The Association for Computational Linguistics, 2015, pp. 110–115. DOI: 10.3115/v1/w15-0514. URL: https://doi.org/10.3115/v1/w15-0514 (cit. on p. 7).

[16] Amber E. Boydstun et al. "Tracking the Development of Media Frames within and across Policy Issues". In: (2014). DOI: 10.1184/R1/6473780.v1. URL: https://kilthub.cmu.edu/articles/journal_contribution/Tracking_the_Development_of_Media_Frames_within_and_across_Policy_Issues/6473780 (cit. on pp. 3, 6, 61, 62, 64, 81).

[17] Greg Brockman et al. *Introducing ChatGPT and Whisper APIs*. https://openai.com/blog/introducing-chatgpt-and-whisper-apis. 2023 (cit. on p. 24).

[18] Tom B. Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle et al. 2020. URL: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html (cit. on pp. 14, 16, 24).

[19] Sébastien Bubeck et al. "Sparks of Artificial General Intelligence: Early experiments with GPT-4". In: *CoRR* abs/2303.12712 (2023). DOI: 10.48550/arXiv.2303.12712. arXiv: 2303.12712. URL: https://doi.org/10.48550/arXiv.2303.12712 (cit. on p. 26).

[20] Ricardo J. G. B. Campello, Davoud Moulavi, and Jörg Sander. "Density-Based Clustering Based on Hierarchical Density Estimates". In: *Advances in Knowledge Discovery and Data Mining, 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part II*. Ed. by Jian Pei et al. Vol. 7819. Lecture Notes in Computer Science. Springer, 2013, pp. 160–172. DOI: 10.1007/978-3-642-37456-2_14. URL: https://doi.org/10.1007/978-3-642-37456-2%5C_14 (cit. on pp. 8, 17).

[21] Dallas Card et al. "The Media Frames Corpus: Annotations of Frames Across Issues". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*. The Association for Computer Linguistics, 2015,

pp. 438–444. DOI: 10.3115/v1/p15-2072. URL: https://doi.org/10.3115/v1/p15-2072 (cit. on pp. 3, 6, 62, 81).

[22] David Carmel, Haggai Roitman, and Naama Zwerdling. "Enhancing cluster labeling using wikipedia". In: *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*. Ed. by James Allan et al. ACM, 2009, pp. 139–146. DOI: 10.1145/1571941.1571967. URL: https://doi.org/10.1145/1571941.1571967 (cit. on p. 8).

[23] Wei-Lin Chiang et al. *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*. Mar. 2023. URL: https://lmsys.org/blog/2023-03-30-vicuna/ (cit. on p. 28).

[24] Kyunghyun Cho et al. "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches". In: *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014*. Ed. by Dekai Wu et al. Association for Computational Linguistics, 2014, pp. 103–111. DOI: 10.3115/v1/W14-4012. URL: https://aclanthology.org/W14-4012/ (cit. on p. 6).

[25] Dennis Chong and James N Druckman. "Framing theory". In: *Annu. Rev. Polit. Sci.* 10 (2007), pp. 103–126 (cit. on pp. 1, 3).

[26] Alexis Conneau et al. "Supervised Learning of Universal Sentence Representations from Natural Language Inference Data". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. Ed. by Martha Palmer, Rebecca Hwa, and Sebastian Riedel. Association for Computational Linguistics, 2017, pp. 670–680. DOI: 10.18653/v1/d17-1070. URL: https://doi.org/10.18653/v1/d17-1070 (cit. on p. 7).

[27] Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. "Reciprocal rank fusion outperforms condorcet and individual rank learning methods". In: *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*. Ed. by James Allan et al. ACM, 2009, pp. 758–759. DOI: 10.1145/1571941.1572114. URL: https://doi.org/10.1145/1571941.1572114 (cit. on p. 67).

[28] Johannes Daxenberger et al. "ArgumenText: Argument Classification and Clustering in a Generalized Search Scenario". In: *Datenbank-Spektrum* 20.2 (2020), pp. 115–121. DOI: 10.1007/s13222-020-00347-7. URL: https://doi.org/10.1007/s13222-020-00347-7 (cit. on p. 7).

[29] Claes H De Vreese. "News framing: Theory and typology". In: *Information design journal+ document design* 13.1 (2005), pp. 51–62 (cit. on p. 3).

[30] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/v1/n19-1423. URL: https://doi.org/10.18653/v1/n19-1423 (cit. on pp. 6, 13).

[31] Lorik Dumani, Tobias Wiesenfeldt, and Ralf Schenkel. "Fine and Coarse Granular Argument Classification before Clustering". In: *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*. Ed. by Gianluca Demartini et al. ACM, 2021, pp. 422–432. DOI: 10.1145/3459637.3482431. URL: https://doi.org/10.1145/3459637.3482431 (cit. on p. 7).

[32] Robert M Entman. "Framing: Toward clarification of a fractured paradigm". In: *Journal of communication* 43.4 (1993), pp. 51–58 (cit. on pp. 1, 76).

[33] Michael Färber and Anna Steyer. "Towards Full-Fledged Argument Search: A Framework for Extracting and Clustering Arguments from Unstructured Text". In: *CoRR* abs/2112.00160 (2021). arXiv: 2112.00160. URL: https://arxiv.org/abs/2112.00160 (cit. on pp. 8, 38).

[34] Tianyu Gao, Xingcheng Yao, and Danqi Chen. "SimCSE: Simple Contrastive Learning of Sentence Embeddings". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*. Ed. by Marie-Francine Moens et al. Association for Computational Linguistics, 2021, pp. 6894–6910. DOI: 10.18653/v1/2021.emnlp-main.552. URL: https://doi.org/10.18653/v1/2021.emnlp-main.552 (cit. on p. 15).

[35] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. "Hybrid speech recognition with Deep Bidirectional LSTM". In: *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, December 8-12, 2013*. IEEE, 2013, pp. 273–278. DOI:

10.1109/ASRU.2013.6707742. URL: https://doi.org/10.1109/ASRU.2013.6707742 (cit. on p. 6).

[36] Maarten Grootendorst. "BERTopic: Neural topic modeling with a class-based TF-IDF procedure". In: *CoRR* abs/2203.05794 (2022). DOI: 10.48550/arXiv.2203.05794. arXiv: 2203.05794. URL: https://doi.org/10.48550/arXiv.2203.05794 (cit. on pp. 8, 9, 37).

[37] Maarten Grootendorst. *KeyBERT: Minimal keyword extraction with BERT.* Version v0.3.0. 2020. DOI: 10.5281/zenodo.4461265. URL: https://doi.org/10.5281/zenodo.4461265 (cit. on p. 77).

[38] Philipp Heinisch and Philipp Cimiano. "A multi-task approach to argument frame classification at variable granularity levels". In: *it Inf. Technol.* 63.1 (2021), pp. 59–72. DOI: 10.1515/itit-2020-0054. URL: https://doi.org/10.1515/itit-2020-0054 (cit. on pp. 6, 7).

[39] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735 (cit. on p. 6).

[40] Jordan Hoffmann et al. "Training Compute-Optimal Large Language Models". In: *CoRR* abs/2203.15556 (2022). DOI: 10.48550/arXiv.2203.15556. arXiv: 2203.15556. URL: https://doi.org/10.48550/arXiv.2203.15556 (cit. on p. 27).

[41] Matthew Honnibal et al. "spaCy: Industrial-strength Natural Language Processing in Python". In: (2020). DOI: 10.5281/zenodo.1212303 (cit. on p. 36).

[42] Ioana Hulpus et al. "Unsupervised graph-based topic labelling using dbpedia". In: *Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, Rome, Italy, February 4-8, 2013*. Ed. by Stefano Leonardi et al. ACM, 2013, pp. 465–474. DOI: 10.1145/2433396.2433454. URL: https://doi.org/10.1145/2433396.2433454 (cit. on p. 9).

[43] *Introducing ChatGPT*. https://openai.com/blog/chatgpt. 2022 (cit. on pp. 4, 24, 26).

[44] Daniel Kahneman and Amos Tversky. "Choices, values, and frames." In: *American psychologist* 39.4 (1984), p. 341 (cit. on p. 2).

[45] Jared Kaplan et al. "Scaling Laws for Neural Language Models". In: *CoRR* abs/2001.08361 (2020). arXiv: 2001.08361. URL: https://arxiv.org/abs/2001.08361 (cit. on pp. 10, 24).

[46] Maurice George Kendall. "Rank correlation methods." In: (1948) (cit. on p. 67).

[47] Nitish Shirish Keskar et al. "CTRL: A Conditional Transformer Language Model for Controllable Generation". In: *CoRR* abs/1909.05858 (2019). arXiv: 1909.05858. URL: http://arxiv.org/abs/1909.05858 (cit. on p. 8).

[48] David Khabaz. "Framing Brexit: The role, and the impact, of the national newspapers on the EU Referendum". In: *Newspaper Research Journal* 39.4 (2018), pp. 496–508 (cit. on p. 2).

[49] Andreas Köpf et al. "OpenAssistant Conversations - Democratizing Large Language Model Alignment". In: *CoRR* abs/2304.07327 (2023). DOI: 10.48550/arXiv.2304.07327. arXiv: 2304.07327. URL: https://doi.org/10.48550/arXiv.2304.07327 (cit. on p. 27).

[50] Wanqiu Kou, Fang Li, and Timothy Baldwin. "Automatic Labelling of Topic Models Using Word Vectors and Letter Trigram Vectors". In: *Information Retrieval Technology - 11th Asia Information Retrieval Societies Conference, AIRS 2015, Brisbane, QLD, Australia, December 2-4, 2015. Proceedings*. Ed. by Guido Zuccon et al. Vol. 9460. Lecture Notes in Computer Science. Springer, 2015, pp. 253–264. DOI: 10.1007/978-3-319-28940-3_20. URL: https://doi.org/10.1007/978-3-319-28940-3%5C_20 (cit. on p. 9).

[51] Olzhas Kozbagarov, Rustam Mussabayev, and Nenad Mladenovic. "A New Sentence-Based Interpretative Topic Modeling and Automatic Topic Labeling". In: *Symmetry* 13.5 (2021), p. 837. DOI: 10.3390/sym13050837. URL: https://doi.org/10.3390/sym13050837 (cit. on p. 9).

[52] Jey Han Lau et al. "Automatic Labelling of Topic Models". In: *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*. Ed. by Dekang Lin, Yuji Matsumoto, and Rada Mihalcea. The Association for Computer Linguistics, 2011, pp. 1536–1545. URL: https://aclanthology.org/P11-1154/ (cit. on p. 9).

[53] Quoc V. Le and Tomás Mikolov. "Distributed Representations of Sentences and Documents". In: *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*. Vol. 32. JMLR Workshop and Conference Proceedings. JMLR.org, 2014, pp. 1188–1196. URL: http://proceedings.mlr.press/v32/le14.html (cit. on p. 9).

[54] Mike Lewis et al. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*. Ed. by Dan Jurafsky et al. Association for Computational Linguistics, 2020, pp. 7871–7880. DOI: 10.18653/v1/2020.acl-main.703. URL: https://doi.org/10.18653/v1/2020.acl-main.703 (cit. on p. 9).

[55] Percy Liang et al. "Holistic Evaluation of Language Models". In: *CoRR* abs/2211.09110 (2022). DOI: 10.48550/arXiv.2211.09110. arXiv: 2211.09110. URL: https://doi.org/10.48550/arXiv.2211.09110 (cit. on p. 24).

[56] Chin-Yew Lin. "ROUGE: A Package for Automatic Evaluation of Summaries". In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013 (cit. on p. 21).

[57] Siyi Liu et al. "Detecting Frames in News Headlines and Its Application to Analyzing News Framing Trends Surrounding U.S. Gun Violence". In: *Proceedings of the 23rd Conference on Computational Natural Language Learning, CoNLL 2019, Hong Kong, China, November 3-4, 2019*. Ed. by Mohit Bansal and Aline Villavicencio. Association for Computational Linguistics, 2019, pp. 504–514. DOI: 10.18653/v1/K19-1047. URL: https://doi.org/10.18653/v1/K19-1047 (cit. on p. 6).

[58] Leland McInnes and John Healy. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction". In: *CoRR* abs/1802.03426 (2018). arXiv: 1802.03426. URL: http://arxiv.org/abs/1802.03426 (cit. on pp. 8, 38).

[59] Leland McInnes, John Healy, and Steve Astels. "hdbscan: Hierarchical density based clustering". In: *The Journal of Open Source Software* 2.11 (2017), p. 205 (cit. on p. 17).

[60] Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. "Automatic labeling of multinomial topic models". In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007*. Ed. by Pavel Berkhin, Rich Caruana, and Xindong Wu. ACM, 2007, pp. 490–499. DOI: 10.1145/1281192.1281246. URL: https://doi.org/10.1145/1281192.1281246 (cit. on p. 8).

[61] Tomás Mikolov et al. "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. Ed. by Christopher J. C. Burges et al. 2013, pp. 3111–3119. URL: https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html (cit. on p. 9).

[62] Tomás Mikolov et al. "Efficient Estimation of Word Representations in Vector Space". In: *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2013. URL: http://arxiv.org/abs/1301.3781 (cit. on p. 7).

[63] Amita Misra, Brian Ecker, and Marilyn A. Walker. "Measuring the Similarity of Sentential Arguments in Dialogue". In: *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA*. The Association for Computer Linguistics, 2016, pp. 276–287. DOI: 10.18653/v1/w16-3636. URL: https://doi.org/10.18653/v1/w16-3636 (cit. on p. 7).

[64] Nona Naderi and Graeme Hirst. "Classifying Frames at the Sentence Level in News Articles". In: *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, Varna, Bulgaria, September 2 - 8, 2017*. Ed. by Ruslan Mitkov and Galia Angelova. INCOMA Ltd., 2017, pp. 536–542. DOI: 10.26615/978-954-452-049-6_070. URL: https://doi.org/10.26615/978-954-452-049-6%5C_070 (cit. on p. 6).

[65] Shashi Narayan, Shay B. Cohen, and Mirella Lapata. "Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Ed. by Ellen Riloff et al. Association for Computational Linguistics, 2018, pp. 1797–1807. DOI: 10.18653/v1/d18-1206. URL: https://doi.org/10.18653/v1/d18-1206 (cit. on p. 49).

[66] OpenAI. "GPT-4 Technical Report". In: *CoRR* abs/2303.08774 (2023). DOI: 10.48550/arXiv.2303.08774. arXiv: 2303.08774. URL: https://doi.org/10.48550/arXiv.2303.08774 (cit. on p. 26).

[67] Long Ouyang et al. "Training language models to follow instructions with human feedback". In: *NeurIPS*. 2022. URL: http://papers.nips.cc/paper%5C_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html (cit. on pp. 4, 25).

[68] Lawrence Page et al. "The PageRank Citation Ranking : Bringing Order to the Web". In: *The Web Conference*. 1999 (cit. on p. 9).

[69] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "Glove: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. ACL, 2014, pp. 1532–1543. DOI: 10.3115/v1/d14-1162. URL: https://doi.org/10.3115/v1/d14-1162 (cit. on p. 7).

[70] Matthew E. Peters et al. "Deep Contextualized Word Representations". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*. Ed. by Marilyn A. Walker, Heng Ji, and Amanda Stent. Association for Computational Linguistics, 2018, pp. 2227–2237. DOI: 10.18653/v1/n18-1202. URL: https://doi.org/10.18653/v1/n18-1202 (cit. on p. 7).

[71] Cristian Popa and Traian Rebedea. "BART-TL: Weakly-Supervised Topic Label Generation". In: *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*. Ed. by Paola Merlo, Jörg Tiedemann, and Reut Tsarfaty. Association for Computational Linguistics, 2021, pp. 1418–1425. DOI: 10.18653/v1/2021.eacl-main.121. URL: https://doi.org/10.18653/v1/2021.eacl-main.121 (cit. on p. 9).

[72] Zheng Lin Qingyi Si. *Alpaca-CoT: An Instruction Fine-Tuning Platform with Instruction Data Collection and Unified Large Language Models Interface*. https://github.com/PhoebusSi/alpaca-CoT. 2023 (cit. on p. 29).

[73] Alec Radford et al. "Improving Language Understanding by Generative Pre-Training". In: (2018) (cit. on p. 14).

[74] Alec Radford et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9 (cit. on pp. 14–16).

[75] Colin Raffel et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67. URL: http://jmlr.org/papers/v21/20-074.html (cit. on p. 49).

[76] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Nov. 2019. URL: https://arxiv.org/abs/1908.10084 (cit. on pp. 8, 15).

[77] Nils Reimers et al. "Classification and Clustering of Arguments with Contextualized Word Embeddings". In: *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Ed. by Anna Korhonen, David R. Traum, and Lluís Màrquez. Association for Computational Linguistics, 2019, pp. 567–578. DOI: 10.18653/v1/p19-1054. URL: https://doi.org/10.18653/v1/p19-1054 (cit. on pp. 7, 8).

[78] Stephen E. Robertson et al. "Okapi at TREC-3". In: *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*. Ed. by Donna K. Harman. Vol. 500-225. NIST Special Publication. National Institute of Standards and Technology (NIST), 1994, pp. 109–126. URL: http://trec.nist.gov/pubs/trec3/papers/city.ps.gz (cit. on p. 7).

[79] Victor Sanh et al. "Multitask Prompted Training Enables Zero-Shot Task Generalization". In: *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL: https://openreview.net/forum?id=9Vrb9D0WI4 (cit. on pp. 25, 49).

[80] Teven Le Scao et al. "BLOOM: A 176B-Parameter Open-Access Multilingual Language Model". In: *CoRR* abs/2211.05100 (2022). DOI: 10.48550/arXiv.2211.05100. arXiv: 2211.05100. URL: https://doi.org/10.48550/arXiv.2211.05100 (cit. on p. 26).

[81] Benjamin Schiller, Johannes Daxenberger, and Iryna Gurevych. "Aspect-Controlled Neural Argument Generation". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*. Ed. by Kristina Toutanova et al. Association for Computational Linguistics, 2021, pp. 380–396. DOI: 10.18653/v1/2021.naacl-main.34. URL: https://doi.org/10.18653/v1/2021.naacl-main.34 (cit. on p. 8).

[82] Tim Schopf, Simon Klimek, and Florian Matthes. "PatternRank: Leveraging Pretrained Language Models and Part of Speech for Unsupervised Keyphrase Extraction". In: *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - KDIR*. INSTICC. SciTePress, 2022, pp. 243–248. ISBN: 978-989-758-614-9. DOI: 10.5220/0011546600003335 (cit. on p. 77).

[83] Rico Sennrich, Barry Haddow, and Alexandra Birch. "Neural Machine Translation of Rare Words with Subword Units". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. DOI: 10.18653/v1/p16-1162. URL: https://doi.org/10.18653/v1/p16-1162 (cit. on p. 11).

[84] Jessica Shieh. *Best practices for prompt engineering with OpenAI API*. https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-openai-api. 2022 (cit. on p. 30).

[85] Shahbaz Syed et al. "Frame-oriented Summarization of Argumentative Discussions". In: 2023. URL: https://webis.de/downloads/publications/papers/syed_2023a.pdf (cit. on p. 7).

[86] Chenhao Tan et al. "Winning Arguments: Interaction Dynamics and Persuasion Strategies in Good-faith Online Discussions". In: *Proceedings of WWW*. 2016 (cit. on p. 36).

[87] Rohan Taori et al. *Stanford Alpaca: An Instruction-following LLaMA model*. https://github.com/tatsu-lab/stanford_alpaca. 2023 (cit. on pp. 28, 31).

[88] Hugo Touvron et al. "LLaMA: Open and Efficient Foundation Language Models". In: *CoRR* abs/2302.13971 (2023). DOI: 10.48550/arXiv.2302.13971. arXiv: 2302.13971. URL: https://doi.org/10.48550/arXiv.2302.13971 (cit. on p. 27).

[89] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al. 2017, pp. 5998–6008. URL: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (cit. on pp. 10–12).

[90] Xiaojun Wan and Tianming Wang. "Automatic Labeling of Topic Models Using Text Summaries". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016. DOI: 10.18653/v1/p16-1217. URL: https://doi.org/10.18653/v1/p16-1217 (cit. on p. 9).

[91] Yizhong Wang et al. "Self-Instruct: Aligning Language Model with Self Generated Instructions". In: *CoRR* abs/2212.10560 (2022). DOI: 10.48550/arXiv.2212.10560. arXiv: 2212.10560. URL: https://doi.org/10.48550/arXiv.2212.10560 (cit. on p. 28).

[92] Canwen Xu et al. "Baize: An Open-Source Chat Model with Parameter-Efficient Tuning on Self-Chat Data". In: *CoRR* abs/2304.01196 (2023). DOI: 10.48550/arXiv.2304.01196. arXiv: 2304.01196. URL: https://doi.org/10.48550/arXiv.2304.01196 (cit. on p. 28).

[93] Susan Zhang et al. "OPT: Open Pre-trained Transformer Language Models". In: *CoRR* abs/2205.01068 (2022). URL: https://doi.org/10.48550/arXiv.2205.01068 (cit. on p. 25).

[94] Tianyi Zhang et al. "BERTScore: Evaluating Text Generation with BERT". In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL: https://openreview.net/forum?id=SkeHuCVFDr (cit. on p. 22).

Acronyms

BERT Bidirectional Encoder Representations from Transformers. 6–8, 13–15, 23

GPT Generative Pre-trained Transformer. 14, 15

HDBSCAN Hierarchical Density-Based Spatial Clustering of Applications with Noise. 8, 17, 19, 39

LLaMA Large Language Model Meta AI. 27, 28

LLM Large Language Model. i, 4, 5, 10, 14–16, 24–29, 47, 48, 60, 61, 68, 70, 71, 73, 76

RLHF Reinforcement Learning from Human Feedback. 16, 25, 26, 73, 74

RRF Reciprocal Rank Fusion. 67

SFT Supervised Fine-Tuning. 16, 25, 26

SOTA state-of-the-art. i, 8, 10, 16, 17, 24, 37–39, 76, 77

UMAP Uniform Manifold Approximation and Projection. 8, 38, 39

Glossary

contextualized word embedding Word representations that capture the meaning of a word based on its surrounding context. 8, 9, 22, 23, 37

cosine similarity Given vectors $\mathbf{A}$ and $\mathbf{B}$, the cosine similarity is the cosine of the angle between $\mathbf{A}$ and $\mathbf{B}$: $\texttt{cosine-sim}(\mathbf{A}, \mathbf{B}) \coloneqq \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$. 7, 9, 15, 22

few-shot A model performs a task for which it wasn't explicitly trained but receives a limited number of demonstrations illustrating how the task should be carried out. 16, 25, 61, 73

generic frame Frame found across different topics. i, 3, 4

instruction-tuning Fine-tuning language models on instruction-answer pairs or conversations. 26, 27, 33, 68, 73, 76

issue-specific frame Frame related to specific topics or events. i, 3, 4, 6–8, 36

zero-shot A model preforms task on which it wasn't explicitly trained without task demonstrations. 16, 25, 64