ContextIQ: A Multimodal Expert-Based Video Retrieval System for Contextual Advertising - Supplementary Material

Ashutosh Chaubey*, Anoubhav Agarwaal*, Sartaki Sinha Roy*, Aayush Agrawal*, Susmita Ghose*
Anoki Inc.
achaubey@usc.edu {anoubhav,sartaki,aayush,susmita}@anoki.tv

A. Alternative Modality Queries to ContextIQ

The flexibility of our system allows us to effortlessly perform queries across different modalities, including video, audio, and image, ensuring any-to-any search capabilities.

Image Query: The process begins by encoding the input query image using the vision encoder of the vision-text model f_{θ_1}, resulting in an image embedding. This embedding is then compared against the vision embeddings of all available content $\{\mathcal{F}_i^v : i = 1, 2, ..., N\}$ using cosine similarity. The system retrieves and ranks content based on these similarity scores.

Video Query: For video queries, the system first extracts frame-level embeddings from the sampled frames of the query video using the vision encoder of the vision-text model f_{θ_1}. These frame-level embeddings are then aggregated using the previously defined aggregation function $\mathcal{A}_v$ to generate a single video embedding. This video embedding is compared directly with the vision embedding database $\{\mathcal{F}_i^v : i = 1, 2, ..., N\}$ using cosine similarity. The content is then ranked according to similarity, with the most relevant videos appearing at the top of the results.

Audio Query: The process for audio queries begins by segmenting the query audio into a fixed number of segments. Each segment is encoded using the audio encoder of the audio-text model g_{θ_2}. These segment-level encodings are then aggregated using the previously defined aggregation function $\mathcal{A}_a$ to form a single audio embedding. This aggregated embedding is compared against the audio embeddings database $\{\mathcal{F}_i^a : i = 1, 2, ..., N\}$ using cosine similarity. The results are then ranked based on these similarity scores.

B. Video Action Recognition

B.1. Simplifying Kinetics 710 classes

Reducing Kinetics 710 [2,3,6] classes to minimize inter-class confusion can be done by either discarding irrelevant classes or combining similar ones. A hierarchical approach to combining Kinetics classes was explored in [8] using a clustering method. However, this approach only provides examples rather than hierarchical clustering for the entire Kinetics dataset. In our ContextIQ system, we reduced the number of classes by collecting various signals and manually determining which classes to discard or combine. As a result, the number of classes was reduced from 710 to 185. The result is captured in this sheet (as referred in the following paragraphs) present in our GitHub repository https://github.com/AnokiAI/ContextIQ-Paper/. The signals used were:

1. **Relevance to contextual advertiser**: Some classes, like ”stretching arm” or ”shuffling feet” may be too mundane, while others, like ”playing oboe” or ”clam digging,” are too niche for a broad audience targeting. Using GPT-4, we identified and marked about 50% of classes as irrelevant for audience targeting (highlighted in red in the attached sheet). Examples of discarded classes include ”Playing oboe” (niche instrument with limited audience), ”Pole vault” (niche sport), and ”Stretching leg” (too general for segmentation).

   GPT4 Prompt:
   I have a list of 710 actions. Create a downloadable sheet with three columns, 710 actions, Discard, Reason. The discard should be Yes, but only if it seems less useful for detecting an action. If Discard is yes, also mention the reason (1-2 lines). I want to discard about 50% of the actions to keep the most useful half. An action is less useful if it does not seem helpful for creating audience segments for ad targeting. E.g, pinching action does not seem useful to target. Do not make any action class. Use the 710 as it is.
   [[Paste the list of 710 actions]]

2. **Groupings and correlated classes**: Many classes in the Kinetics 710 set have high overlap, and their occurrence is highly correlated. For example, there are three separate classes for playing guitar, strumming guitar, and tapping guitar, which the model finds difficult to

[*] These authors contributed equally to this work

differentiate, leading to higher inter-class confusion. To address this, we used two approaches: one extends the existing Kinetics 400 [6] groupings to 710 classes, and the other examines the top correlated classes during prediction.

K400 Groupings: The K400 set [6] provided groupings of the 400 classes into 37 groups. However, these groupings were not extended to the additional 300 classes in the K710 set. For these additional 300 classes, we inferred their groupings by finding their text similarity with the 37 groups using the text encoder h_{θ_3} used in the main paper and tagging the class to the most similar group. These classes are marked with a double asterisk in the K400 grouping column in the sheet.

Top-3 correlated classes: The VideoMAE2 model [11] generates logits for 710 classes during inference. We build a co-occurrence matrix by counting every pair of classes in the Top-10 logit scores, then compute the correlation matrix. This process is applied to both the Kinetics validation set (50,000 videos) and our internal movie/TV clip set (Sec. 5.2 in the main paper). For instance, classes like dunking, dribbling, shooting, and playing basketball are highly correlated, allowing us to merge them into a single class, such as playing basketball.

3. **Accuracy on the K710 validation set**: Some classes perform poorly on the simpler Kinetics validation set (likely the same data distribution they were trained on), making them less likely to perform well on our movie clip dataset. We calculate the Top-1 and Top-3 accuracy for each class on the Kinetics set and highlight those in the bottom 25th percentile in the sheet. For instance, photobombing has a 38.8% Top-1 and 51% Top-3 accuracy, making it a candidate for discarding to reduce false positives and inter-class confusion.

4. **Class occurrence ranking**: Kinetics includes many classes that rarely occur in the wild, such as wood burning (art), stacking die, and wrestling alligator. In our large, diverse internal dataset, we found that 90% of the Top-3 search results come from only 218 classes 1. In the attached sheet, we list the occurrence count rank of each class (from 1 to 710) and highlight those beyond the top 218.

Using the above four signals, the 710 classes were manually screened and reduced to **185 classes**:

- 182 classes were discarded.

- 418 classes were combined into 89 classes (see Tab. 1 for some of the obtained combined classes).

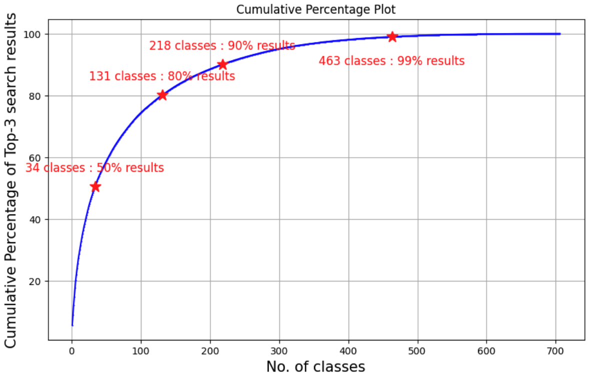

Figure 1. Cummulative Percentage Plot of Top-3 predicted actions on the internal set

Table 1. Few examples of obtained combined classes

Combined class	Actions belonging to the class
playing cards	playing poker, shuffling cards, card stacking, card throwing, dealing cards, playing blackjack
drinking alcohol	uncorking champagne, bartending, drinking beer, drinking shots, tasting beer, playing beer pong, pouring beer, tasting wine, pouring wine, opening bottle (not wine), opening wine bottle
riding animal	riding camel, riding elephant, riding mule, riding or walking with horse
playing board game	playing monopoly, playing checkers, playing dominoes, playing mahjong, playing scrabble
cleaning floor	cleaning floor, mopping floor, sweeping floor, brushing floor, vacuuming floor, sanding floor

- 96 classes were retained.

Combining classes involves a trade-off between losing specificity and improving average precision @ K and prediction confidence. For instance, while predicting a broader class such as “drinking alcohol” (refer to row 2 of Table 1) can yield higher precision, it sacrifices the ability to differentiate between specific types like wine and beer.

C. Emotion Recognition

Text-based Emotion Recognition. We use a pre-trained Emoberta-Large [7] model, which is trained on the MELD [10] and IEMOCAP [1] datasets, for text-based emotion recognition as it is a speaker-aware model and shows better performance empirically on movie scene subtitles.

Leveraging Visual and Audio Cues for Emotion Recognition. The text-based models work only when there is enough text for the model to make a prediction. Moreover, it is difficult to find subtitles for some content, but still, we need to predict emotions in them for better retrieval and brand safe filtering. Since we already use the vision-text model and audio-text models for different parts of the ContextIQ system, we use these models to get some extra signals for predicting emotion. For example, we tagged the emotion *joy* with text queries like *people smiling* and *people dancing*, and assigned the emotion *joy* to all videos retrieved through the video (vision) modality using these queries. Concretely, we associate textual concepts that can be linked to different emotions and then find the scenes that have high video embedding similarity with the emotional text concept. Assume $Q_t = \{t : e\}$ to be the text concept dictionary which contains strings t associated with different emotions e. Then, for a particular video scene x, we say that it is associated to an emotion e if,

$$f_{\theta_1}(x) \cdot f^T_{\theta_1}(t) > \tau_e \tag{1}$$

where f_{θ_1} and $f^T_{\theta_1}$ are the video and text encoders, respectively, of the vision-text model [9], and τ_e is a predefined threshold for the concept-emotion pair $t : e$.

Empirical results show that textual emotion concepts work well only for *joy* emotion. For other emotions, either it is difficult to find emotional text concepts which are relevant to that emotion, or the text concept associated to the emotion is not well represented by the vision-text model.

Similar to visual concepts, we associate audio concepts to different emotions given by $Q_a = a : e$, which contains audio files a and corresponding emotion e associated with that audio file. Then for a particular video scene x, we say that it is associated to an emotion e if,

$$g_{\theta_2}(x_a) \cdot g_{\theta_2}(a) > \tau_e \tag{2}$$

where g_{θ_2} is the audio encoder of CLAP [13], x_a is the audio for the given video and τ_e is a predefined threshold for the concept-emotion pair $a : e$. Note that we do not use the text encoder of CLAP because text-audio matching did not result into as good results as audio-audio matching. We have only linked audio emotion concepts to *sad* emotion because the rest of the emotions do not show good results empirically.

D. Hate Speech Detection

Aggregation Strategy: To combine predictions from the BERT model, the scores for the Hate Speech and Offensive classes are summed. This aggregated score is then compared against a threshold of $\theta = 0.7$. The final prediction is obtained by applying a logical OR operation between the thresholded BERT prediction and the predictions from the LLM to boost recall.

Prompting Strategies: We implement various prompting techniques to enhance the predictive performance of the LLM [5].

1. **Few-Shot Learning**: A few examples are provided to the model to establish task context, improving its ability to accurately identify hate speech. Specifically we use three examples for the same.
2. **Definition of Hate Speech**: A precise definition of hate speech is included in the prompt to ensure consistent detection aligned with the dataset annotations. We use the following definition of hate speech : *Language that disparages a person or group on the basis of protected characteristics like race, gender, and cultural identity.*
3. **Structured JSON Output**: The model is instructed to return its response in JSON format, enabling easy parsing and seamless integration with the contextIQ system.

4. **Chain of Thought Reasoning**: The model is prompted to generate intermediate reasoning steps before determining whether content qualifies as hate speech, enhancing prediction accuracy. [12]

   Various analyses were performed to evaluate the effectiveness of these strategies by using a combination of them for detection. Table 3 presents the results of these analyses. The results demonstrate that incorporating all the prompting strategies enhances detection performance, leading to improvements in accuracy, precision, and F1 score.

Validation Data and Results: We conducted validation using two datasets: an internal dataset and the implicit-hate dataset [4]. For implicit-hate, we sampled 250 examples each of Explicit Hate Speech, Implicit Hate Speech, and Normal Speech to ensure a balanced evaluation across different types of speech. In contrast, the internal dataset consisted of 11,645 examples, which, after applying a profanity filter, was reduced to 10,645. Given the unbalanced distribution of hate speech versus normal speech on internal dataset, calculating recall was challenging. As a result, we only focused on the positive predictions generated by each model.

On the internal dataset, the BERT model identified 397 out of 10,645 examples (3.7%) as positive, while the LLM predicted 509 examples (4.8%) as positive. To assess these predictions, we randomly sampled 40 examples from each set of positive predictions, which were reviewed by two independent curators, given the subjective nature of the task. While precision varied significantly between curators owing to the subjective nature of the task, the LLM consistently outperformed the BERT model, with an average delta of 7.5%.

Table 2. Classification metrics for LLM, BERT and Ensemble Model

	Explicit Hate vs Normal Speech				Implicit Hate vs Normal Speech			
Metric	LLM	BERT	Ensemble (OR, $\theta = 0.7$)	Ensemble (AND, $\theta = 0.2$)	LLM	BERT	Ensemble (OR, $\theta = 0.7$)	Ensemble (OR, $\theta = 0.2$)
Accuracy	83.9	77.7	81.5	85.3	75.3	63.4	73	73.2
Precision	78.9	75.9	74.5	82	75.2	66.8	70.7	76.9
Recall	92.3	81.1	95.1	90.2	74.9	52.4	78.1	66
F1 Score	85.1	78.4	83.5	85.9	75.1	58.8	74.2	71

Table 3. Differential Analysis for different prompting strategies

	Explicit Hate vs Normal Speech					Implicit Hate vs Normal Speech				
Reasoning	Yes	No	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
Definition of Hate Speech	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No
Number of Examples	3	3	1	0	3	3	3	1	0	3
Recall	94.6	**97.2**	95.2	93.5	94.9	76.5	**85.6**	74.8	77.8	75.5
Precision	**73.9**	65.8	72.3	70.2	71.0	**70.3**	62.9	67.3	66.3	67.4
Accuracy	**80.8**	73.4	79.4	77.1	78.7	**71.9**	67.6	69.2	69.3	69.5
F1 Score	**83.0**	78.4	82.2	80.2	81.2	**73.3**	72.5	70.8	71.6	71.2

For the implicit-hate dataset, we evaluated various prompt templates and temperature values to enhance the performance of the LLM. A temperature value of 0.6, combined with the prompt template described here, yielded the optimal results. Table 2 presents the results for the best parameter combinations for both the LLM-based and BERT models, along with the outcomes for the ensemble models. The ensemble model outperformed the individual models, offering the flexibility to fine-tune precision and recall according to specific requirements. Additionally, the table also provides results for the ensemble model using both AND and OR operations across two different threshold values. The selection of these parameters can be guided by the desired balance between precision and recall in different scenarios.

References

[1] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N. Chang, Sungbok Lee, and Shrikanth S. Narayanan. Iemocap: interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42(4):335–359, Nov. 2008. 2

[2] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. Submitted on 3 Aug 2018. 1

[3] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. Submitted on 15 Jul 2019 (v1), last revised 17 Oct 2022 (this version, v2). 1

[4] Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang. Latent hatred: A benchmark for understanding implicit hate speech. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 345–363, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. 3

[5] Keyan Guo, Alexander Hu, Jaden Mu, Ziheng Shi, Ziming Zhao, Nishant Vishwamitra, and Hongxin Hu. An investigation of large language models for real-world hate speech detection, 2024. 3

[6] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. Submitted on 19 May 2017. 1, 2

[7] Taewoon Kim and Piek Vossen. Emoberta: Speaker-aware emotion recognition in conversation with roberta, 2021. 2

[8] Nicolas Lemieux and Rita Noumeir. A hierarchical learning approach for human action recognition. *Sensors*, 20(17), 2020. 1

[9] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023. 3

[10] Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. MELD: A multimodal multi-party dataset for emotion recognition in conversations. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–

536, Florence, Italy, July 2019. Association for Computational Linguistics. 2

[11] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking, 2023. 2

[12] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc. 3

[13] Yusong Wu*, Ke Chen*, Tianyu Zhang*, Yuchen Hui*, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023. 3