Supplementary Material for Query-Focused Video Summarization: Dataset, Evaluation, and A Memory Network Based Approach

Aidean Sharghi[†], Jacob S. Laurel[‡,*] and Boqing Gong[†]
[†]Center for Research in Computer Vision, University of Central Florida, Orlando, FL 32816
[‡]Department of Computer Science, University of Alabama at Birmingham, AL 35294
aidean.sharghi@knights.ucf.edu, jslaurel@uab.edu, bgong@crcv.ucf.edu

Here we provide the supplementary materials to support the main text.

Section A thoroughly explains how the queries are constructed from dense concept annotations that are acquired from AMTurk-ers.

Section B describes the algorithm to aggregate multiple user summaries (per video-query pair) into one *oracle* summary to facilitate training supervised approaches.

Section C analyzes the proposed evaluation metric's behavior when we randomly *replace* some shots in a user summary.

Furthermore, two video summaries generated by our summarizer are included (cf. Chocolate_Street.mp4 and Drink_Food.mp4 enclosed).

A. Constructing Queries

In this section, we thoroughly describe the process of generating the queries from dense concept annotations. While users often input free text to query videos through search engines, we simulate the real scenarios and construct the queries using the dense concept annotations we have collected (cf. Section 3.2 in the main text) to ease benchmarking different algorithms.

By processing the dense user annotation data, we extract various statistics that enable us to have the queries covering a wide range of varieties. Initially, a concept is assumed present in the video if it appears in at least T shots. This is to filter the present noise in annotations acquired from AMT workers and to make sure the concepts really appear together (to steer clear of the pairs that are tagged together as a result of noise or bias).

As described in the main text, when a user enters a query q (for instance, on a video search engine), which is usually more than one word, we have four distinct scenarios; i) all the concepts in the query appear in the same video shots together, ii) all concepts appear in the video, but never jointly in a single shot, iii) only one of the concepts constituting the query appears in the some shots of the video, and iv) none of the concepts in the query are present in the video (1 such query). A robust video summarizer, must be able to maintain good performance under any of the scenarios. Therefore, by including enough samples of all the scenarios, we build a comprehensive and diverse dataset.

For the scenario i, we create a list of pairs that appear together in the same shots and sort it in descending order. There are two approaches to select concept pairs from this list: 1) to employ a random selection process where the probability of selecting a pair from the list is proportional to the number of times the pair appeared together in the video (this gives higher chance to the concepts that tend to happen together in the video while not completely crossing out the concepts that are not dominant in the video), and 2) to pick few top concept pairs. We opt to use the random selection process to better generalize the dataset and remove bias.

For the scenario ii, we are interested in concept pairs that are present in the video but not in the same shots, e.g., concept pairs such as CAR and ROOM that are unlikely to appear in the same shots of the video. To this end, for each pair we compute their harmonic mean of frequencies:

$$score(f_{c_1}, f_{c_2}) = \frac{f_{c_1} \times f_{c_2}}{f_{c_1} + f_{c_2}} \tag{1}$$

where f_{c_1} and f_{c_2} are the frequencies of concepts c_1 and c_2, respectively. This formulation has two interesting features that make it useful in this regard; 1) the resulting combination of numbers fed to it is always smaller than the smallest entry, 2) it is maximized when both inputs are large and identical. By computing the harmonic mean of frequencies for all the pairs in the list and sorting it in descending order, the concept pairs that have high frequencies for both concepts constituting the query are ranked higher. At this point,

[*] Jacob S. Laurel contributed to this work while he was an NSF REU student at UCF thanks to the support of NSF CNS #1461121.

we employ the same random selection process to randomly choose pairs from this list.

For the third scenario, we concentrate on pairs that only one concept constituting the query is present in the video, e.g., if there is no CAR present in the entire video while there exists shots with COMPUTER appearing in them, the pair CAR and COMPUTER is a candidate for this scenario. To make sure that the constructed dataset is comprehensive and benefits from the versatile dictionary, we first exclude the concepts that were used in the first two scenarios, we put the rest in a list and use their frequencies to randomly sample from them.

For the last scenario, where neither of concepts in pairs must be present in the video, we simply use the concepts that never appear in the video.

For scenarios i, ii, and iii, we select 15 queries. For scenario iv, we only choose one query; summarizing based on any such query consisting of concepts that are not present in the entire video must result in about the same summary. In other terms, when a user wants the model to summarize the video based on a query consisting of non-present concepts, the summarizer must only return *contextually* important segments of the video, that is essentially what a conventional generic video summarization approach (as opposed to query-dependent approaches) generates.

Figure 2 shows that queries play a major role in the summaries that users generate. For a particular video, the same user has selected summaries that have both common (green margin) and uncommon (orange margin) segments.

B. Generating Oracle Summaries

Supervised video summarization methods often learn from one summary per video, or in the case of query-focused summarization per query-video pair. On the other hand, for evaluation purposes, it is better to contrast a system summary against multiple references and report the average. Thus, we collected 3 user summaries per query-video pair to use for evaluation purposes. However, in order to train the model, we obtain *oracle* summaries that have maximum overall agreement with all three reference summaries (per query-video pair).

The algorithm [1] starts with the set of common shots (y^0) in all three reference summaries. Next, at each iteration, it greedily includes the shot that returns the largest marginal gain $G(i)$,

$$G(i) = \sum_{u} \text{F1-score}(y^0 \cup i, y_u) - \sum_{u} \text{F1-score}(y^0, y_u) \tag{2}$$

where u iterates over the user summaries (in our case, $u \in \{1, 2, 3\}$) and F1-score is obtained from our proposed evaluation metric. Table 1 in the main text shows the correlation between the obtained oracle summaries and user summaries, showing the oracle summary has very high agreement with all the user summaries.

C. Evaluation Metric Behavior

As described in Section 5.2 of the main text, we studied the effect of randomly **removing** some video shots from the user summary on our proposed metric, observing that our metric has a linear drop in recall. Due to the fact that captions only capture limited information about the scene (cf. Figure 1), repeating the same experiment and evaluating with ROUGE-SU4 on captions provided by [3], recall showed a non-linear drop. As a side experiment, figure 3 illustrates the effect of randomly **replacing** some video shots in the user summary, studying the effect of noise on performance. Here we are swapping some shots with others that might be very similar or different to the original shots. For reference, we include the ROUGE-SU4 metric in our experiments as well.

D. Summary Examples

In addition to the text, we have enclosed two system summaries to provide qualitative results. The first video summary corresponds to query q={FOOD,DRINK} (scenario i), and consists of 93 shots (each shot is 5 seconds long), making it less than 8 minutes long while the original video is $\sim$ 4 hours. The second, with query of q={CHOCOLATE,STREET} (scenario ii), is a summary of length $\sim$ 5 minutes (56 shots) generated by our model for a 3 hours long video.

References

[1] A. Kulesza and B. Taskar. Determinantal point processes for machine learning. *arXiv preprint arXiv:1207.6083*, 2012. 2

[2] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain, 2004. 3

[3] S. Yeung, A. Fathi, and L. Fei-Fei. Videoset: Video summary evaluation through text. *arXiv preprint arXiv:1406.5824*, 2014. 2, 3

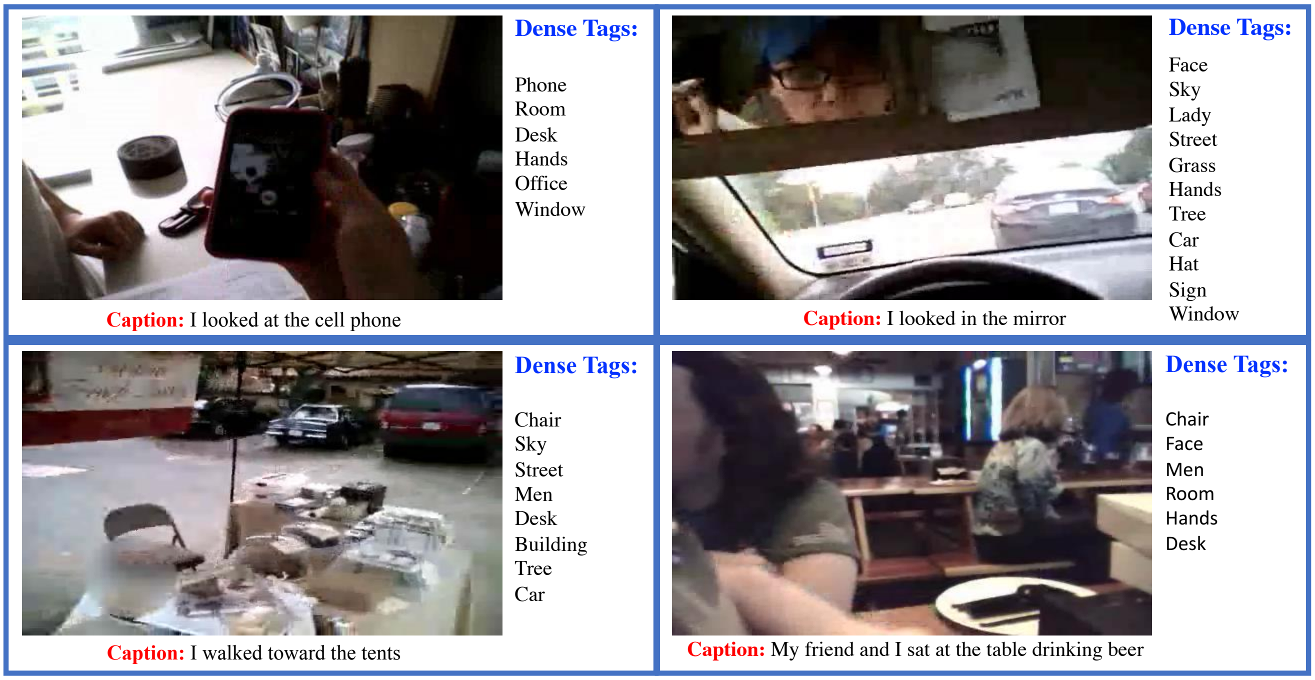

Figure 1: Comparing semantic information in our dense tags vs captions provided by [3]. The figure illustrates that the caption is targeting limited information about the scene, while the dense annotations are able to better explain the characteristics of the scene.

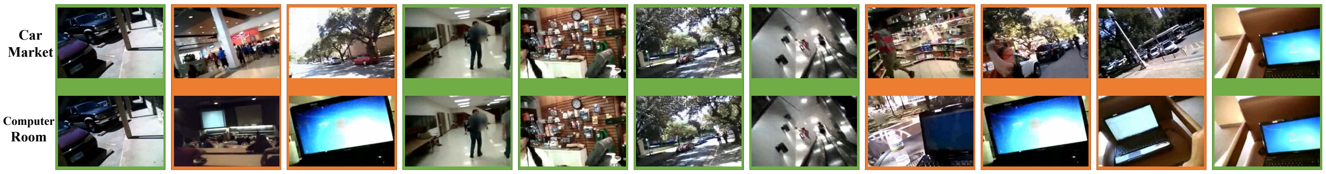

Figure 2: This figure compares two user summaries (generated by the same user) for two different queries. Both summaries contain shared segments, that are assumed important in the context of the video, while they disagree on the query-relevant segments.

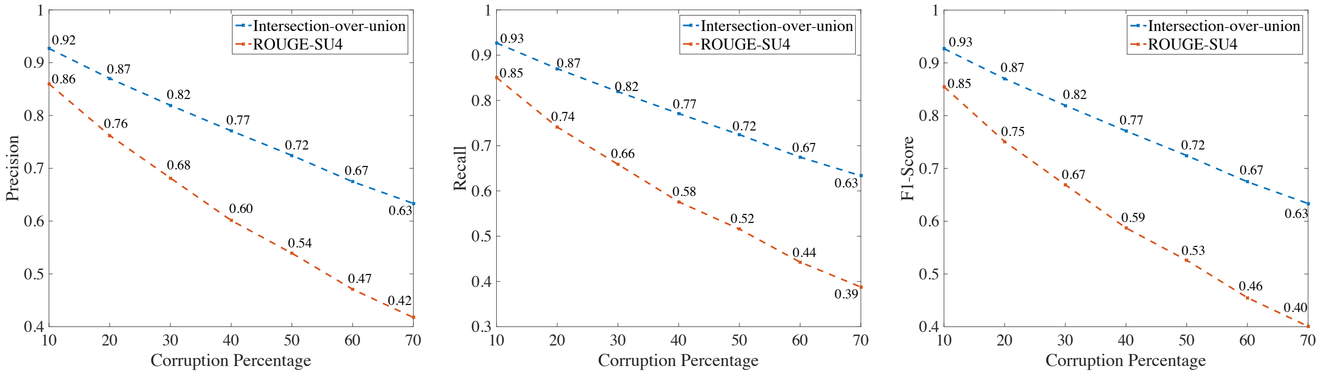

Figure 3: Studying the effect of randomly *replacing* some video shots in the user summary on the performance. The evaluation by ROUGE-SU4 [2] is included for reference.