Milestone Report on Replication of YouTube's User-Generated Content Summarization and Topic Extraction Model

Tommy Shang Wu, Jake McFarland, Abiel Jeong-Joon Kim

Abstract

The objective of our project, to recreate our own variant of YouTube's comment section AI (artificial intelligence) generator, is proceeding along as stipulated in the project proposal. Utilizing the dataset mentioned previously, we have created a clustered set of data that is adequately separated by topic. Using this and our upcoming steps, our proposal should be completed on time.

1 Introduction

Using k-means clustering and a transformer model, we intend to perform summarization of large amounts of data. This topic has many uses, usually to do with reducing the amount of time needed to read through large amounts of text, as well as highlighting the larger ideas while discarding the smaller and less pertinent details.

2 Related Work

While the target of our work, YouTube's comment topics, is still in testing and a very new feature, there are numerous examples of text summarization being used on a day-to-day basis. Scholarcy, a summarizing AI, is used to create digestible versions of otherwise lengthy reports and textbooks (Scholarcy, 2024). This is intended to reduce the amount of time needed to study previous examples and new material as well as create more time to work on projects. Other related works take things beyond purely summarization. While this performs summarization, like our work, it does not perform it in the same manner. There is a large difference between summarizing a large, cohesive amount of text, and summarizing a lot of unrelated messages. For example, Recall, another summarizing tool, claims to create a 'knowledge graph' that takes higher level concepts from summarized content and saves them to a database, then makes connections on content you have watched or read. It then helps you recall these topics using active recall strategies (Recall Documentation, 2025). This example is more in line with what we intend our work to result in, where information from multiple sources is compiled into a single summarization.

3 Approach

The high-level approach and system pipeline remained consistent since the project proposal with heightened details with respect to low-level implementation decisions. To reiterate, the ultimate project objective is to engineer an NLP system that takes in a swath of user-generated comments and outputs a summary of key overlapping topics as a means of communicating general viewership thought or sentiment.

Our proposed system pipeline comprises two primary systems which abstractly factor the project directive into two more tractable tasks. Namely, the identification of key overlapping topics via K-means clustering followed by the summarization of each key topic via a deep-learning NLP model. The following sections elaborate on the implementation-specific decisions that we made to bring this system to life.

3.1 Topic Extraction Subsystem

3.1.1 Sentence-Embedding Projection Method

As discussed in our project proposal, the core idea was to implement take the set of user comments, project them into the embedding space, and then run K-means clustering across the embedding vectors to identify key overlapping topics to which can then be iteratively fed into the summarization model.

Our original implementation intent was to project the user-generated comments into the embedding space using Word2Vec or GloVe. Specifically, we originally planned on representing each comment - which can be thought of as a sentence

with respect to our dataset - as a weighted sum of its token embeddings.

After further alternative methods investigation, we found this approach to be too naive and hypothesized a significant loss of information especially as the comment length grows larger. Instead, we implemented the mapping of sentence embeddings directly via a popular pretrained model from Hugging Face found here: (Hugging Face, 2025).

As per their description, the transformer-based model, all-mpnet-base-v2, is specifically designed for sentences and paragraphs revolving around tasks like clustering or semantic search which matches the needs of our topic extraction subsystem perfectly. Note that the sentence embeddings have 768 dimensions as opposed to GloVe or Word2Vec's typical 300.

3.1.2 K-Means Clustering: Finding Optimal K

Once each comment has been mapped to a 768-dimensional space, we implemented K-means over the sentence-embedding space as a means to find key overlapping topics with respect to general viewer sentiment or thought. However, in order to determine the number of clusters, K, that our system should identify, we implemented 3 main approaches: Predefined K, search K via elbow method, search K via silhouette score.

The first approach, using a predefined K, is the simplest approach and least computationally costly. We simply manually set the number of clusters K that our system should interpret before running SKLearn's KMeans algorithm over the set of sentence embeddings.

$$K_{opt} = C$$

The second approach searches for an optimal K value via the elbow method. It works by defining a range of K values to try, and picking the optimal K which minimizes the sum of squared distance from a centroid to its member sentence embeddings, called WCSS (Within Cluster Sum of Squares). This approach discovers a more suitable K than the predefined-K approach at the cost of greater computational cost.

$$\text{WCSS} = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$

The third approach searches for an optimal K value via leveraging silhouette scores. It works by defining a range of K values to try, and picking the optimal K which maximizes the average silhouette score, which is a global measure of how cohesive clusters are and how well they are separated. The average silhouette score requires computing the silhouette score for all embedding vectors which is calculated by considering how cohesive it is in the context of its associated cluster (average distance to every other vector in the same cluster) and how well separated it is from unassociated clusters (average distance to every other vector in complementary clusters). i.e. Maximizing the average silhouette score sees that every point is highly cohesive (close to every related member point from the same cluster) and well separated (far from every other unrelated point from different clusters).

$$S_{avg} = \frac{1}{N} \sum_{i=1}^{N} s(i)$$

Our experimentation and analysis suggest that the final implementation should leverage the second approach, approximating optimal K via the elbow method. See the experimentation section for associated reasoning and evidence to support our findings.

4 Experiments

4.1 Comparison of the Elbow Method and the Average Silhouette Score

To compare the reported values of optimal K between the elbow method and average silhouette score approaches, we ran both approaches on varying K ranges and compare their outputs. More concretely, we define an upper bound threshold, K_{max}, which defines a range of K-values $1, ..., K_{max}$. Different upper bound thresholds produce different ranges of candidate K values. Each range of candidate K values yield different optimal K solutions when running either elbow method or silhouette score approach. Thus, we want to compare the optimal K solutions that we get from both approaches as K_{max} increases or decreases.

For this experiment, we tested $K_{max} = 10, 20, ..., 200$ over a subsample of 984 sentence embeddings. Observe Table 1 for a summary of our findings when invoking both the elbow method and average silhouette score approach for varying upper bounds of candidate K values.

Our findings show that the elbow method and average silhouette score do not agree on what the

Max k	Silhouette n-Clusters	Elbow n-Clusters
10	4	5
20	4	9
30	4	9
35	4	9
40	4	9
45	4	10
48	4	10
50	4	10
60	59	10
90	59	10
100	59	10
200	199	10

Table 1: Cluster Analysis Results

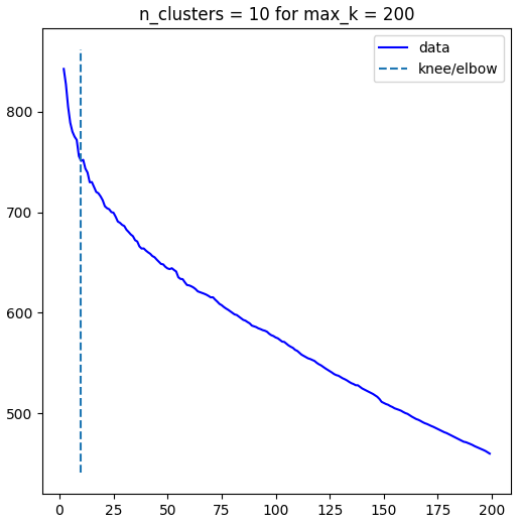

Figure 1: Elbow Plot

optimal K should be for varying ranges of candidate K values. In fact, *both methods appear to exhibit almost opposing behaviors insofar that the elbow method approach converges to an optimal K as we increase the upper bound of candidate Ks whereas the silhouette score approach converges as we decrease the upper bound* under the sensible range.

4.2 Insights of the Elbow Method

As reported in section 4.1, the elbow method converges to an optimal K of 10 as we test a greater number of candidate clusters over the 984 samples (Figure 1). Conversely, the elbow method decreases its optimal K count as the number of tested candidate clusters decrease (we lose the optimal K of 10 at around $K_{max} = 50$).

This finding suggests that, across the 984 sample space, the elbow method requires a certain upper bound of candidate Ks in order to identify a clear elbow of the WCSS vs K-Clusters plot. Otherwise, the distance curve becomes too flat and the inflection point becomes less apparent. **It therefore follows that the elbow method is a stable and promising approach that yields the optimal K (likely around or equal to 10) but only when provided a sufficient K_{max}.**

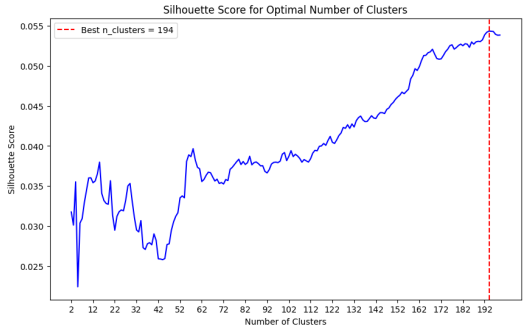

Figure 2: K-Means Silhouette Plot

Note that the optimal K reported by the elbow method corresponds to this elbow-like inflection point and is calculated using the kneedle algorithm, implemented by KneeLocator from Python's kneed module.

4.3 Insights of the Average Silhouette Score

Typically, a well-clustered dataset should see the average silhouette score peak at an optimal K with its scores degrading in value for varying candidate Ks diverging from the optimal. Interestingly, in the context of our 984 sample, the silhouette score approach favors K=4 clusters when testing smaller ranges of candidate Ks, $K_{max} < 50$ but diverges for large ranges of K, $K_{max} > 50$. In fact, the data tends to yield optimal K = $K_{max} - 1$ for large K_{max} (Figure 2).

This finding suggests that as we scale the upper bound threshold of our K range, the K-Means model overfits to the 984 sample with clusters reducing to singleton clusters where each embedding vector becomes its own centroid. In the limit, one can realize this by imagining the uniform distribution across all sentence embeddings in 768 dimensions. This aligns with the theoretical literature underpinning the behavior of the average silhouette score where a rough upper bound threshold is suggested at about:

$$K_{\max} \approx \sqrt{\frac{N}{2}}$$

Where N represents the sample size. At some point past this recommended threshold, the space

becomes weakly clustered, exposed by its tendency to overfit for large K upper bounds. **Consequentially, the silhouette score approach reveals that a clustering structure indeed emerges with an optimal K around 4 when testing comparatively lower upper bounds thresholds, but loses stability and over fits as we scale K_{max}.**

4.4 K-Means Final Approach

Given that we have answered the 3 questions proposed in the beginning of section 4, evidence appears to indicate that the best method to approximate optimal K during inference is to employ either a standalone implementation of the average silhouette score approach or a hybrid combination of the silhouette score approach and elbow method.

Given that, as we are replicating YouTube's summarizer AI, any given comment section for a YouTube video can differ dramatically in content and number of clusters, we decided that a predefined K is too naive. *Thus, the predefined K approach is eliminated.*

A standalone implementation of the elbow method is promising, but requires a sufficient K upper bound threshold in order to identify a clear inflection point and hence optimal K. That is, comparatively larger compute costs are required to recognize a clear convergent optimal K, otherwise the WCSS vs K-clusters curve becomes too flat - likely a function of the underlying dataset's complexity. *Thus, the standalone elbow method is eliminated.*

The standalone silhouette approach empirically suggests the requirement of a relatively smaller upper bound threshold using our aforementioned rule of thumb formula which is promising. A hybrid approach that combines the silhouette score and elbow methods are also promising i.e. consider taking the arithmetic average between both approaches.

Therefore, by process of elimination, we decided that the final K-means approach to be either a standalone implementation of the average silhouette score method or a hybrid combination of both the average silhouette score approach and elbow method. For the final project, we plan on defining certain thresholds for sample size N that invokes either or as a piecewise function.

5 Future Work

The results we have so far are promising. Moving forward, we intend to implement the summarization of each k-mean using two different transformer models, and then evaluate the performance of each model using both BLEU score and ROUGE score.

5.1 Summarization

The most important part of our work will be the text summarization. In order for our work to function according to design, it is entirely necessary for the summarization to produce meaningful results. Working within the deadline, it is our intent to implement this summarization model in two ways: once by building our own transformer model from the ground up, and once by utilizing the T5 model from the transformer module. We intend to follow existing documentation on implementing such models, using the 'Summarization' guide on HuggingFace (Hugging Face, 2024). We intend to use a Topic Extraction system using the weighted-average embedding vectors. Given a set of comments c1, c2 ... cn, a set of corresponding embedding vectors will be created, i.e. e1, e2 ... e2. Each of these vectors is the weighted average of their corresponding vector c. When we map out these vectors in a multidimensional space, we will be able to notice semantic relations using our k-means clustering to define shared topics. Once we have created these clusters, we will give each of these clusters to the summarizer model.

5.2 Evaluation

When measuring the performance of a summarization model, there are two main considerations: precision and recall. We have decided to use a separate evaluation method to measure each of these, namely, BLEU and ROUGE.

5.2.1 BLEU scores

The BLEU score is a method used to measure how close our model will be to the training data. It is performed by checking how many n-grams in the output summarization are also in the training data to receive a precision value, as well as a brevity penalty which is used to punish the machine for extremely short translations. YouTube's Topic summarizations are very short, usually 1-2 sentences at most. Therefore, although we will be testing a number of values for the brevity penalty, it is likely that a smaller than usual value will be used.

5.2.2 ROUGE scores

The ROUGE score is a method used a lot in text summarization tasks. It is similar to the BLEU

method, but instead of checking how many n-grams in the output are also in the training data, it checks how many n-grams in the training data are also in the output to receive a recall value. Due to the length of YouTube's Topics being relatively short, and the size of the text being relatively large, it is likely we will only use somewhere between ROUGE-1 to ROUGE-3, where the number represents the maximally sized n-gram being considered. This is to cut down on the computation necessary in cases where it is likely to be a trivial result.

5.2.3 Model Comparison

Finally, we intend to compare our model to the actual model it is being built in likeness of, the YouTube Topic generator. In order to measure our models against this generator, we will need to use API calls to get all the comments from several videos with this feature in testing below them, then calculate YouTube's BLEU and ROUGE score compared to our own. We will make the assumption that the YouTube Topic summarization model also uses k-clustering in order to make this comparison fair, and we will also assume that there are not blocks on certain words being included in topics or individuals who cannot contribute to these topics, even though that is likely not the case.

6 Timeline

Our work has currently been completed up to the k-means clustering stage of the project. We are satisfied with the clustering and embeddings created in this manner, and expect to move on to the summarization in the coming days. We expect to stay on track of our timeline previously mentioned in the project proposal, completing all of our work by the due date.

References

Hugging Face. 2024. Summarization. Accessed: 2025-03-20.

Hugging Face. 2025. all-mpnet-base-v2. Accessed: 2025-03-20.

Recall Documentation. 2025. Docs. Accessed: 2025-03-20.

Scholarcy. 2024. About-us. Accessed: 2025-03-20.