Master Semester Project Report

Fall semester

Characterising the User Interests on YouTube

Author:
Olivier Quôc-Vinh Lam
(olivier.lam@epfl.ch)

Supervisors:
Manoel Horta Ribeiro
Robert West

January 15, 2021

Abstract

Social media has become a part to our daily routine, through multiple social platforms. This work focuses on the YouTube data by creating an embedding space at the video level. We use the Latent Dirichlet Allocation (LDA) algorithm to perform topic modelling, which resulted to be much more sensitive to the input data than to the model parameters. Furthermore, we implemented different methods to evaluate and interpret the results. Finally, a human evaluation task is performed on our optimal models in order to verify the performance of our evaluation metrics. We hope that our work can help future projects which focus on performing embeddings on very large dataset.

Contents

1 Introduction

1.1 The Rise of the Social Media

In the last couple of decades, we are witnessing the rise of the social media. Indeed, in the early 2000's, the first social media MySpace reached a million monthly active users whilst nowadays, the biggest social platform Facebook regroup more than 2.7 billions monthly active users. In addition, five other platforms (YouTube, WhatsApp, Facebook Messenger, WeChat and Instagram) have reached and exceed the 1 billion monthly active users. Not only the number of user keeps growing but also the average time each user spend in social platforms is increasing [Ort19; Cle20]. In 2018, 45% of the US teenagers say that they are almost constantly connected to the internet against 24% in 2015 [AS18] and in 2019, the average daily time (worldwide) spent on social networking was projected as 153 minutes, which is 60 minutes more than in 2012. All the facts above raise some questions such as: "What are we reading/watching on the social media ?". As YouTube is the biggest platform for video streaming and is the second biggest overall social platform, our work will be to study and characterize the evolution of the user interests on YouTube.

1.2 YouTube

YouTube outperformed his direct rivals such as Google Video thanks to his ability to spread and captivate the users thanks to his user based system. Indeed, while Google Video platform was concentrated on making the most profit as possible by uploading only "the better videos", YouTube deployed the platform as being community based where everyone can add his own videos. However, the genius of YouTube remains on the fact that they achieved to do a community based platform having a large number of commerce videos qualified as "the better videos". [SV09]
Therefore, the advanced recommendation system as well as the good equilibrium between community and commerce led YouTube to record the impressive value of 2 billion monthly active users in 2019. [IQB20]

1.3 Goals of the Work

In order to find and analyse the communities in this huge social platform, two directions come intuitively in mind. On one hand performing an embedding at the video level and on another hand performing another embedding at the channel level. Naturally, the two spaces should share common characteristics which would determine the behaviour of users on YouTube. This work focuses on the video embedding while a parallel work done by Stanislas Jouven focuses on the channel embedding. Results are expected to be shared and combined together in the end.

Unfortunately, the results from the work won't be reproducible as the main dataset has changed. However, similar work with other datasets can be produced by following the steps described at https://github.com/epfl-dlab/youtube_embeddings.

2 Datasets

In the project, we are analyzing the videos from YouTube, the largest social platform for videos streaming. There exists millions of channels, each of which is managed by one user account. In a channel, multiple videos can be posted, where each video contains a title, a description, some tags, an uploaded date and a field where other users can interact within the video. They can like or dislike the video, add a comment to it, like a comment of other user and answer to a particular comment.

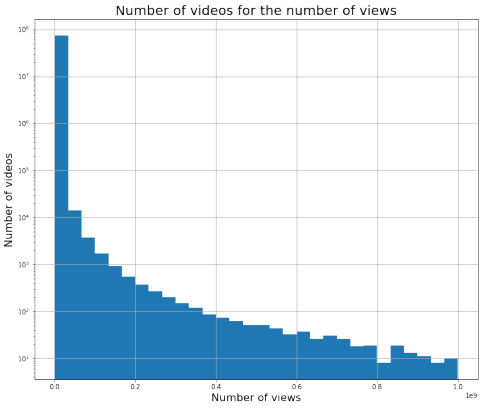

Figure 1: A histogram for the distribution of the videos by the number of views per video.

In this work, we will mainly focus on two datasets. The first contains the informations about the channels, such as the name, the number of subscribers, the number of videos, the category and the date where it joined YouTube. The second one contains all the data about the videos except the informations on the comments in the videos. Those data are crawled into another dataset that won't be used in this work.

A first work has been done on the two datasets in order to keep only the English videos and their corresponding channels. This yields to a dataset of 73'301'516 videos from 136'470 channels,

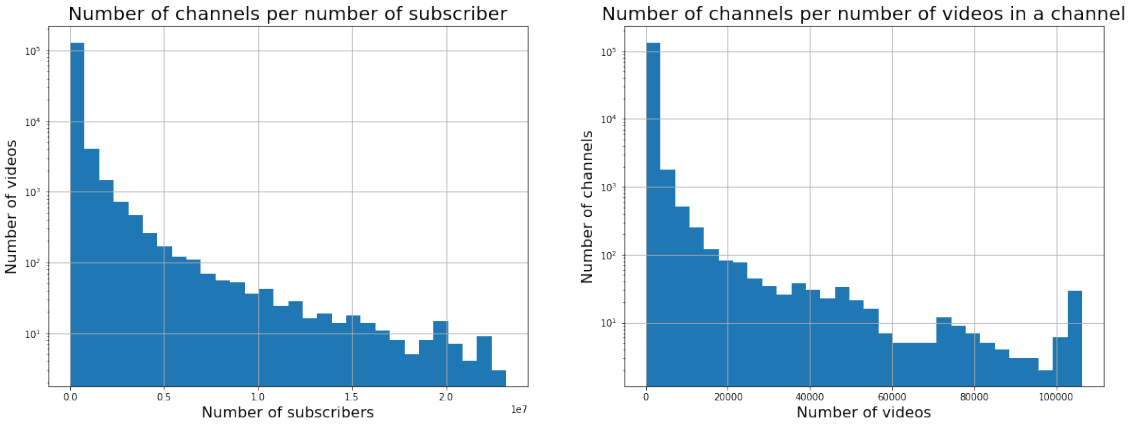

Figure 2: Histograms that show how the channels are distributed according to the number of videos uploaded and the number of subscribers per channel.

which were uploaded on YouTube from 24/05/2005 to 20/11/2019.

On the Figure 1, we can observe the number of videos per number of views in the log scale where we removed the 136 videos that contains more than 1 billion of views, for readability reasons. Figure 2 shows the number of channels, also in a log scale, per number of subscriber and number of videos uploaded per channel. The 50 channels with highest number of subscribers and number of videos uploaded are also removed in the histograms for a nicer representation. Figure 1 and Figure 2 clearly show that the data can be noisy considering the number of videos with few views and the number of channels with few subscribers and/or uploaded videos. For example, there are 24'559'810 videos that have less or equal 1'000 views, which represent more than one third of the dataset. By noisy, we assume that videos with a low number of views may contain random words in its title, tags and/or description. For channels with a low number of subscription or low number of published videos, we also assume that it may be a casual user that just upload casual videos, without the effort of describing them in an adequate way. Furthermore, Figure 3 shows the video counts for each category and it is important to note that the data is not well-balanced for the categories either.

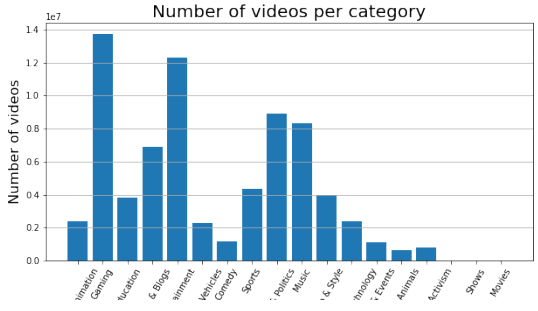

Figure 3: Bar plot for the number of videos per category

3 Methodology

3.1 Data Processing

In order to achieve our goal, we are using the Latent Dirichlet Allocation (LDA) algorithm. More details on it can be found at the subsection 3.2. Many factors can contribute to the quality of the results, especially the data used for training the topic modelling model. As shown later in the section 4, getting the right data for the topic model changes the results significantly while tuning the model doesn't improve the results. Therefore, it is very important to achieve an adequate processing task on the data.

We first construct a document-term matrix where each row corresponds to a video (document) and each column corresponds to a token (term). The tokens are the pre-processed words from a video title, tags and/or description. Considering the title and the tags seems intuitive but since the description may be completely unrelated to the content of a video, we need to verify that using the description yields a better model. By analyzing the document-term matrix (without the description), we have an average of around 48 terms per document, which means that we are dealing with short documents. The literature suggests that LDA may not perform well on short texts, which is why we also try to lengthen the documents by generating and considering bigrams as term [HD10; Tan+14]. As we showed in the previous section, the YouTube dataset is not well-balanced since there are many videos with few views and very few videos with a high number of views. The following question arises: Should we consider all the videos for topic modelling ? If not, how should we choose the videos?

In order to answer that question, we compare 4 models trained on different data and pick the one that yields the best topics. We arbitrarily select the thresholds on the minimum number of views for videos to 1'000 and 10'000 views and the minimum number of subscribers for channels to 10'000 and 100'000 subscribers. That way, we are also able to understand if the LDA algorithm performs better with more data or with better data, as videos with more views from channels with more subscribers are assumed to be more relevant.

In order to tune the hyper-parameter of the LDA model (explained in the next section), we choose to train the different models from a sub-sample of videos for feasibility reasons. Indeed, feeding the model with all the data requires a very large computer resource and time. As the videos from the dataset are already not well-balanced accordingly to the categories, we use the following approach for selecting the sub-sample of videos: for each combination of the category, uploaded year and channel of the videos, select the 20 videos with the most views.

Finally, it is also useful to filter the tokens as initially more than 5.7M tokens have been generated from the data. Therefore, we also need to find the optimal threshold T where we keep only the tokens that appears in at least T videos.

Table Table 1 summarizes the first four model that we train with a fixed number of topics, with the selection for the sub-sample of videos and the tokens filtering with a threshold $T = 100$ as described above. We then manually select the model that yields the most coherent topics and

Model	Min view counts per video	Min subscriber counts for channel	Number of videos used for training	Size of the Vocabulary
Model 1	1'000	10'000	12'140'822	96'902
Model 2	1'000	100'000	4'502'199	50'769
Model 3	10'000	10'000	8'171'467	73'878
Model 4	10'000	100'000	3'899'314	46'481

Table 1: The 4 first built models, based on different sub-sample of videos as input.

check if the model performs better when adding the description of the videos into the vocabulary for documents and/or generating bigrams.

3.2 Topic Modelling

In order to find and characterize the communities from the YouTube dataset, we will be using the well known Latent Dirichlet Allocation (LDA) algorithm for topic modelling proposed by Blei et al. [BNJ03]. LDA is a dimensionality reduction technique that has a very nice and intuitive representation. Indeed, we pass from a space defined by the number of documents and the number of terms to the embedding space, where each topic is described by a distribution over the terms and each document is described by a distribution over the topics. However, LDA requires a number K of topics as a parameter of the model, which is a challenge to find its optimal value. Considering the size of the YouTube data, it is non-feasible to train a model on the whole data for every number K of topics in order to extract the optimal value. This is why we came with an heuristic for selecting a sub-sample of videos, as described in the previous section, that is used for training multiple models. Afterwards, we measure the performance of the models and select the best one before tuning the document concentration (*alpha*) and topic concentration (*beta*) of the LDA model with the optimal number of topics. Performance measures of the model are described in more details in the next section. The higher the document concentration is, the higher the number of topics per document is assumed and the higher the topic concentration is, the more a topic is likely to represent a mixture of terms. We set $alpha \in \{0.1, 0.5, 0.9, 0.95, 1\}$ and $beta \in \{0.01, 0.05, 0.1, 0.5\}$ as these values should work best, as proposed Griffiths et al. [GS04].

With that high number of LDA models, we pick the two best models with respect to the metrics described below, one with a relatively small and one with a large number of topics. A final human evaluation is performed on the two models in order to make ensure the quality of the topic model.

3.3 Metrics for the Topic Model

In this section, we present the methods we used for evaluating a topic model.

3.3.1 Human Ratings for Topic Model

In order to select the best model from the Table 1, we use a simple human evaluation method. As the four models are trained with $K = 50$ number of topics, we give a score for each topic and sum up all the topic scores for each model. The scores $s \in \{0, 1, 2\}$ are given as follows, for each topic that is represented by the 10 terms with highest weight:

- $s = 2$, if a common topic is clearly identifiable from all of the top terms
- $s = 1$, if multiple topics are identifiable or a topic is described by at least 3 terms and all the other terms aren't linked
- $s = 0$, if no topic is recognisable

Since this human evaluation method is non-feasible for measuring a high number of models, we use the coherence score and the accuracy of a simple classifier as metric.

3.3.2 Coherence Score

Introduced by Röder et al., the coherence score for each topic represents how the topic's top term are semantically close to each other [RBH15]. As there are many ways to define the coherence score, we use 2 variants that performed best in previous works. The first one is called the c_v coherence score. It counts the co-occurrences of the terms by using a sliding window and based on these counts, it computes the normalized pointwise mutual information (NPMI) between the terms with highest weight. Therefore, one vector for each top word is computed and the mean of the cosine similarity for every pair of the top terms results in the c_v coherence score. The second variant of the coherence score is the u_mass score, proposed by Mimno et al. [Mim+11]. It assumes that if a document contains a top term from a topic, it should also contain another top term of the same topic with high probability. Hence the logarithm of the conditional probability of every term is computed by using the document co-occurrence counts. Note that the conditional probability is given by considering only terms that have a higher weight in a topic. Taking the arithmetic mean of the probabilities yield the u_mass coherence score.

3.3.3 Classifier

The YouTube dataset includes a category feature for every video, which takes value in a set of 16 elements. As the LDA algorithm is a dimentionality reduction technique, the idea is to recover the already trained LDA model and for each video, we get the topic distribution as the vector of features. Then, we train a simple classifier which can recover the correct category of the video. Finally, the accuracy of the classifier is a metric for the performance of the LDA model. The

intuition comes from the fact that the better the topics are defined, the less the distribution over the topics is mixed for a video. Hence, a good model means the distribution over the topics for each video should be concentred in one topic.

For this task, we select the videos that are considered as relevant but that have not been used in the topic modelling training task. A second filtering step is done in order to get well-balanced data with respect to the categories.

For the classification task, we use the SVM method, with a simple tuning in the coefficient for the regularizer term. As the goal of the classifier is only to measure the performance over the LDA models, we haven't spend time to improve the accuracy on the results.

3.3.4 Human Evaluation

Finally, the best way to measure the performance of the LDA model is still the human evaluation. In order to make it feasible, we proceed as follows:

1. Retrieve the 5 terms with the highest weight for every topic.
2. For each topic t, consider the top 1% of the terms and select the term with the lowest weight such that it appears at least in one of the top terms from the other topics $t' \neq t$. This term is the intruder term for the topic t.
3. In every topic, join its intruder term to the top 5 terms and shuffle the order.
4. Pick 20 random topics and ask a person to retrieve the intruder for each topic.

The better a topic is described by its top terms, the easier it should be to find the intruder. Constraining the intruder to be in the 1% top terms ensure that the intruder is still a good term, since without this constraint, it becomes too easy to spot the intruder - it suffices that one of the topic contains a meaningless word, this word has high probability to be the intruder of the other topics - and setting the constraint to the top 50 terms is not sufficient to distinguish the intruder to the top terms.

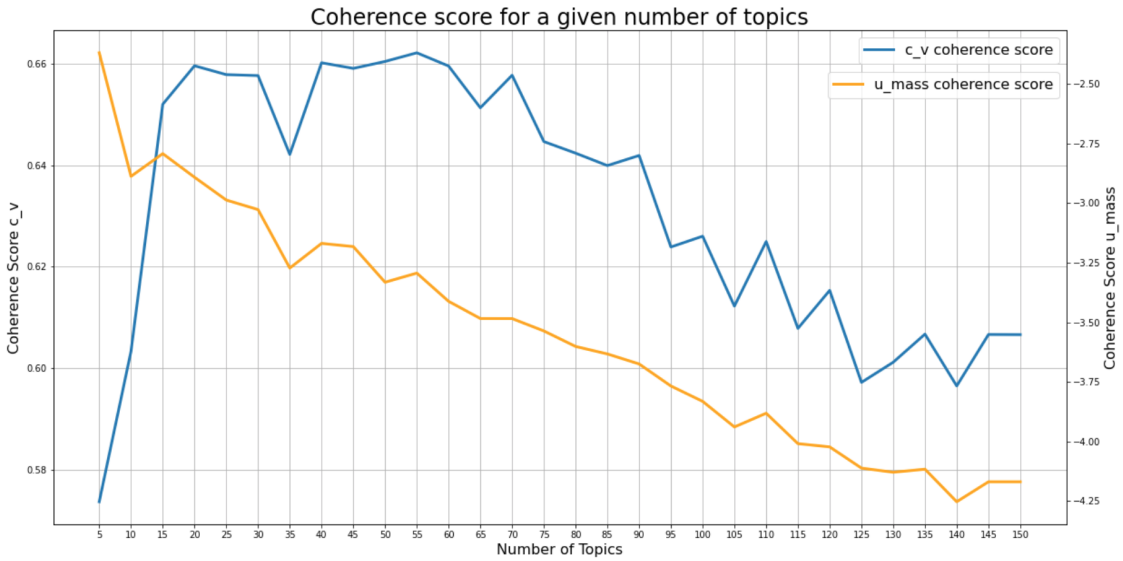

Figure 4: *c_v* coherence and *u_mass* coherence scores for LDA models with k number of topics.

4 Results

The first task is to determine which of the 4 models from Table 1 works best. Recall that all the four models are trained with the same number of topics $K = 50$, arbitrarily chosen. The model 4 get the highest score of 39, compared to a score of 25, 30 and 28 for the first, second and third model, respectively. Therefore, the fourth model is picked in order to find the optimal K number of topics.

4.1 Model Tuning

Figure 4 shows the *c_v* coherence and *u_mass* coherence for every models with a specific number of topics. By their definition, the behaviour of the *c_v* coherence score is expected to follow a logarithmic growth and the opposite is expected for the *u_mass* coherence score. As for both metrics, the higher score means the higher the coherence is in their respective definition. Therefore, we look for the number of topic where the *c_v* score is high and begins to grow slowly within bigger K and where there are some peaks for the *u_mass* score. Oddly, as the number of topics increases, the *c_v* coherence score decreases. This may imply that for a high number of topics, the top terms of each topic become too specific that they lose their common context.

Figure 5 presents the accuracy score for every models with a specific number of topics based on the classifier defined in subsubsection 3.3.3. Using this metrics, the higher the number of topics is, the higher is the accuracy with a maximum at $K = 150$ topics, which can look as a contradiction with the c_v coherence score. As the accuracy is based on a classifier, the more features there are the

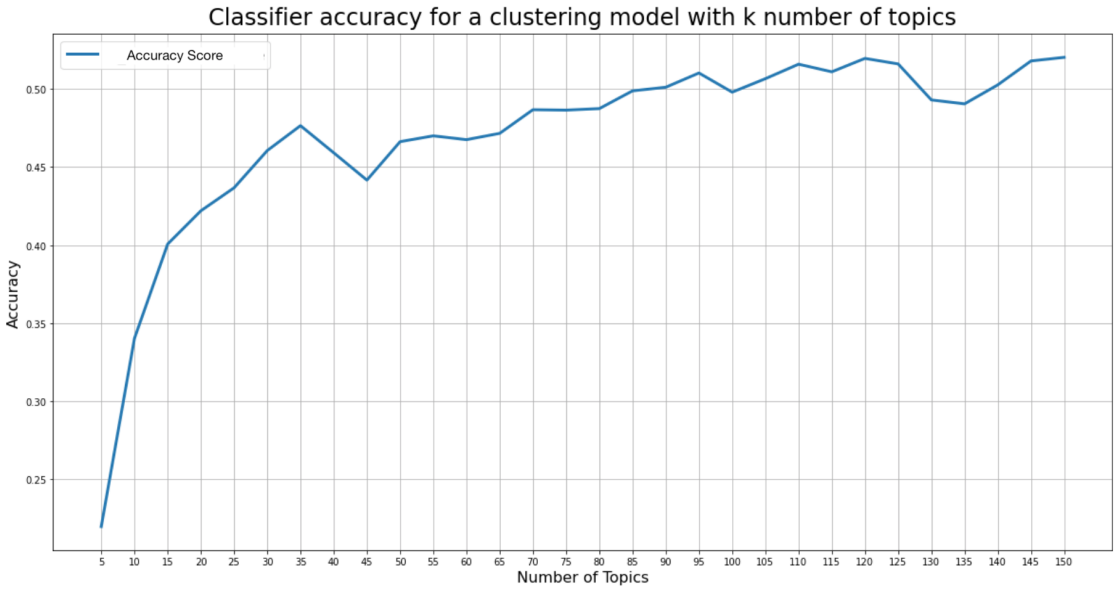

Figure 5: Accuracy scores for LDA models with K number of topics based on the classifier defined in section 3.3.3

better the model performs and hence the higher the accuracy is. Therefore, we look for the model with K number of topics such that the accuracy is higher than the one of its two closest models, which are the models with $K-5$ and $K+5$ number of topics.

Based on the observations above and on the results of the metrics, we pick $K = 55$ and $K = 110$ for future analyses for the topic model. Indeed, fot the model with $K = 55$ number of topics, the c_v score reach the maximum value and the u_mass score and the accuracy of the classifier are higher than the closest models. As we want to verify that these metrics are correlated with the human evaluation, we also pick the model with $K = 110$ topics, where the used metrics suggest that the topics are not represented as well as the model with $K = 55$ topics. However, it performs better than its closest model in all of the three metrics.

Figure 6 shows the coherence scores for the LDA model with $K = 55$ topics with given values for document concentration and topic concentration. Surprisingly, tuning the hyperparameters of the LDA model do not improve nor the c_v score, nor the u_mass score. The same observation can be done with the model with 110 topics, where the results are shown in the Figure 8 in the appendix. As for the accuracy of the classifier, results are not improved either. Figure 9 and Figure 10 from the appendix show the results.

Adding the description of the videos in the corpus of the LDA model results in topics that were impossible to identify the corresponding context when inspecting the top terms of each topic and lengthening the text using bigrams results in the same way. Therefore, we apply the final human evaluation on the two models with $K = 55, 110$ topics and without specifics hyperparameters.

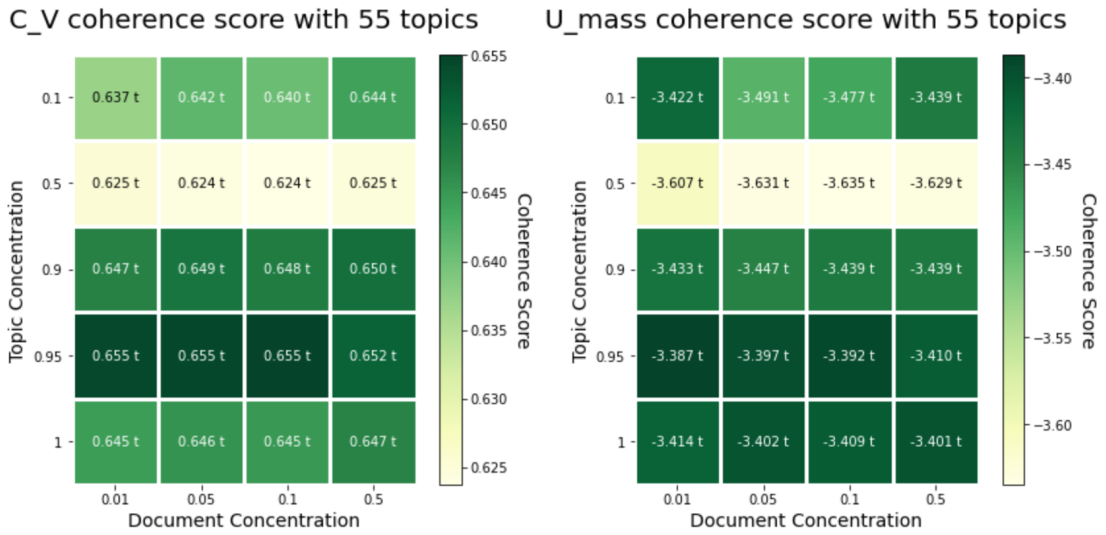

Figure 6: *c_v* coherence and *u_mass* coherence scores for LDA models with $K = 55$ topics and with specific values for document concentration and topic concentration.

We asked 14 people to complete the final task. In the appendix, demographic information and the score for each participant is shown in Table 2. The accuracy for each user is computed as follows:

$$avg\_accuracy = \frac{\#correct\_intruders}{\#intruders}$$

The final score for a model is given by the average accuracy among all the participants. The average accuracy for the model with 55 topics is of 0,47 while the model with 110 topics gets an accuracy of 0,32. It is important to keep in mind that participants 1, 2, 3 are involved in this work, whose results could be biased since they are used to the vocabulary and have some prior insights about the topic model. It is also interesting to notice that all the participants who come from a non-IC related field get a score lower than the average. One of the reason may be the fact that the terms we give the participants are the tokenized version of the words. Therefore, those participants may not be used to deal with this form of the words.

From the results of the human evaluation, we can state that the topics from the first model are better represented by the top terms since it is easier to spot the intruder. Therefore, we can conclude that the *c_v* coherence score is the metric which is the most correlated to the human evaluation.

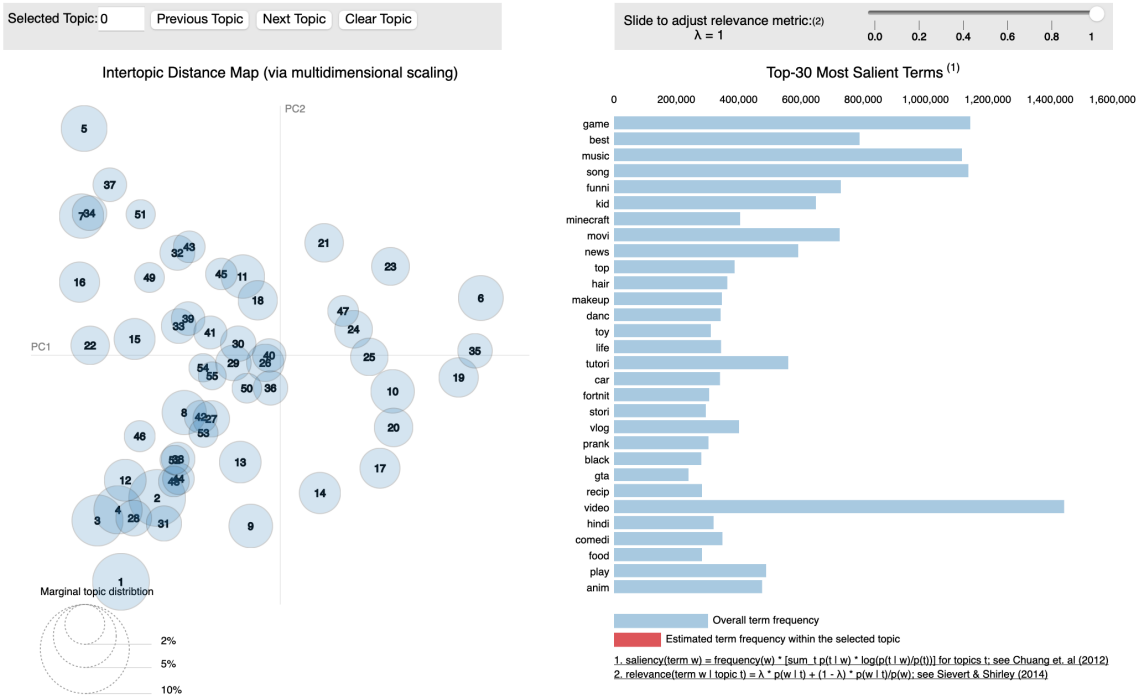

Figure 7: Screenshot from the interactive plot of the topic model.

4.2 Analysis of the Optimal Model

The final representation of the topics from the model with $K = 55$ topics can be visualized here. The same representation of the model with $K = 110$ topics can be found here. The following analysis will be done only on the model with $K = 55$ topics since this is clearly the better model. PCA is applied on the embedding space in order to represent the clusters on a two-dimensional axis. Figure 7 shows a screenshot of the interactive plot if the reader is not able to access the interactive representation. We notice some encouraging results, for example, when checking the term *game*, we can see that the topics which represents it are located in the upper-left of the graph. When checking the topics in this area, we have the topics 18(5), 22(7), 25(34), 50(51) from the Figure 11 and the topic number in parenthesis corresponds to the cluster number from the interactive plot, or in the Figure 7. The same can be observed for the lower-left part of the graph where the topics in that area mainly correspond to the *Entertainment* category, where the topics represents the music, the movies and the news. In the right area of the graph, we can find well-described topics around the *Howto & Style* category, where we can find topic for makeup tutorial and hairstyle for example.

We notice that, in the graph, the further the topics are represented from the center, the better they are described by their top terms and for these topics, their neighbourhood are really likely to share a common wider concept. As for the topics located in the center of the graph, it tends to be harder to extract a nice categorization of the top terms.

Furthermore, as suggests Sievert et al., taking account the most probable terms given a topic to characterize a topic may not be optimal [SS14]. A slider in the interactive plot allows to adjust the relevance metric of a term to the topic. A lambda value of 1 means that we consider the weight of each term by its probability distribution inside the topic while a lambda value of 0 ranks the

terms by their lift. Unfortunately, in our case, playing with the term relevance did not improve the interpretation of the topics.

Finally, Figure 11 shows the top 10 terms of each topics, ranked by the probability distribution of the term in the topic, which we consider it as the weight. The word counts are taken from the entire corpus of the input data of the model. By analysing the topics, we can identify that the gaming category is represented by the topic 18, 22, 41, 50, 51 and 54, which is clearly the category that is the most represented. When comparing to the Figure 3, we also have that the category that is the most represented in the YouTube dataset is the Gaming category.

5 Discussions

5.1 Potential of our Topic Model

The observations from the subsection 4.2 show that our embedding at the video level for the YouTube dataset shows some real potential. Indeed, some very specific topics as the topic 41 in Figure 11, which describe the video game *Five Nights at Freddy's*, are very easy to interpret while the topic 52 in Figure 11 is more difficult to interpret, even tough it talks about subject that are more known as *minecraft* and the smartphones *Samsung Galaxy, Samsung Note*. This could be improved by finding the optimal value for λ for the term relevance in the topics. Indeed, we can observe in Figure 11 that for almost every topics, the word counts of some terms are very high compared to the other counts inside the same topic. Even though we checked manually the value for λ if the results would improve, a more conventional way to set the value of this parameter may lead to a better interpretation of the topics.

5.2 Reflections on the Project

By looking back at the beginning of the work, we started by trying to construct an embedding on the whole YouTube dataset and the original goal of the project is to characterize the evolution of the user interests in YouTube. Separating the videos by their uploaded year attribute and find an embedding for every year could be another way to start the project. Indeed, we would have dealt with smaller dataset and wouldn't encounter the issues with the available computer resources. On another hand, the insights gained with these smaller embedding could lead us to perform a better final topic model on the whole data.

6 Conclusion

From the YouTube dataset, we created an embedding space at the video level by using the Latent Dirichlet Allocation algorithm. The main challenge in this task is to select the correct data for the topic model and to evaluate and interpret the results. Unfortunately, we haven't been able to merge our results to the ones of the embedding at the channel level. Handling the large dataset and generating a good topic model revealed to be more challenging than expected. However, a lot of good insights from the YouTube data can be derived from this project and future work could use the implementations done here.

References

[AS18] Monica Anderson and Aaron Smith. "Social Media Use in 2018". In: *Pew Research Center* (2018).

[BNJ03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation". In: *J. Mach. Learn. Res.* 3.null (Mar. 2003), pp. 993–1022. ISSN: 1532-4435.

[Cle20] J. Clement. *Global social networks ranked by number of users 2020*. 2020. URL: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/.

[GS04] Thomas L. Griffiths and Mark Steyvers. "Finding scientific topics". In: *Proceedings of the National Academy of Sciences* 101.suppl 1 (2004), pp. 5228–5235. ISSN: 0027-8424. DOI: 10.1073/pnas.0307752101. eprint: https://www.pnas.org/content/101/suppl_1/5228.full.pdf. URL: https://www.pnas.org/content/101/suppl_1/5228.

[HD10] Liangjie Hong and Brian D. Davison. "Empirical Study of Topic Modeling in Twitter". In: *Proceedings of the First Workshop on Social Media Analytics*. SOMA '10. Washington D.C., District of Columbia: Association for Computing Machinery, 2010, pp. 80–88. ISBN: 9781450302173. DOI: 10.1145/1964858.1964870. URL: https://doi.org/10.1145/1964858.1964870.

[IQB20] MANSOOR IQBAL. "YouTube Revenue and Usage Statistics (2020)". In: (2020). URL: https://www.businessofapps.com/data/youtube-statistics/.

[Mim+11] David Mimno et al. "Optimizing Semantic Coherence in Topic Models". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11. Edinburgh, United Kingdom: Association for Computational Linguistics, 2011, pp. 262–272. ISBN: 9781937284114.

[Ort19] Esteban Ortiz-Ospina. *The rise of social media*. 2019. URL: https://ourworldindata.org/rise-of-social-media.

[RBH15] Michael Röder, Andreas Both, and Alexander Hinneburg. "Exploring the Space of Topic Coherence Measures". In: *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. WSDM '15. Shanghai, China: Association for Computing Machinery, 2015, pp. 399–408. ISBN: 9781450333177. DOI: 10.1145/2684822.2685324. URL: https://doi.org/10.1145/2684822.2685324.

[SS14] Carson Sievert and Kenneth Shirley. "LDAvis: A method for visualizing and interpreting topics". In: *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*. Baltimore, Maryland, USA: Association for Computational Linguistics, June 2014, pp. 63–70. DOI: 10.3115/v1/W14-3110. URL: https://www.aclweb.org/anthology/W14-3110.

[SV09] Pelle Snickars and Patrick Vonderau. *The YouTube reader*. Pelle Snickars /Patrick Vonderau National Library of Sweden, 2009.

[Tan+14] Jian Tang et al. "Understanding the Limiting Factors of Topic Modeling via Posterior Contraction Analysis". In: *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. ICML'14. Beijing, China: JMLR.org, 2014, I–190–I–198.

Appendices

Participant	Gender	Study Background	Accuracy with M1	Accuracy with M2
1	M	Ms IC	0.55	0.35
2	M	Ms IC	0.45	0.35
3	M	PhD IC	0.6	-
4	M	Ms IC	0.6	-
5	M	Ms IC	0.25	-
6	M	Ms IC	-	0.3
7	M	Ms IC	-	0.3
8	M	Ms IC	-	0.2
9	M	Bs HEC	0.35	-
10	F	Ms HEC	-	0.3
11	F	Ms Bio	0.4	-
12	M	Ms IC	0.55	-
13	M	Bs IC	-	0.45
14	M	Ms Law	-	0.3
Average	-	-	0.47	0.32

Table 2: Results on the human evaluation experiment. $M1$ defines the topic model with $K = 55$ number of topics and $M2$ defines the topic model with $K = 110$ number of topics.

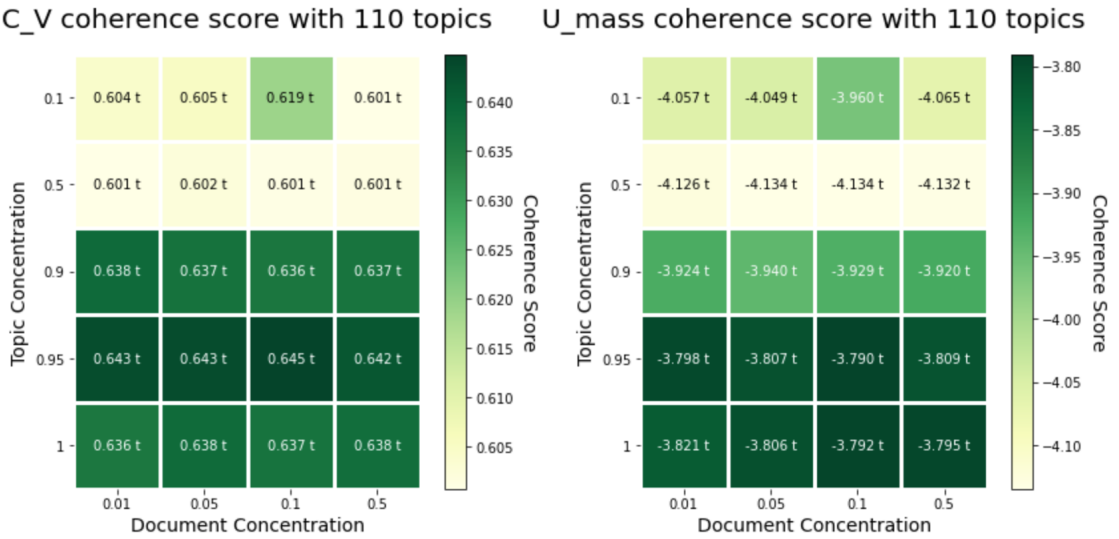

Figure 8: *c_v* coherence and *u_mass* coherence scores for LDA models with $K = 110$ topics and with specific values for document concentration and topic concentration.

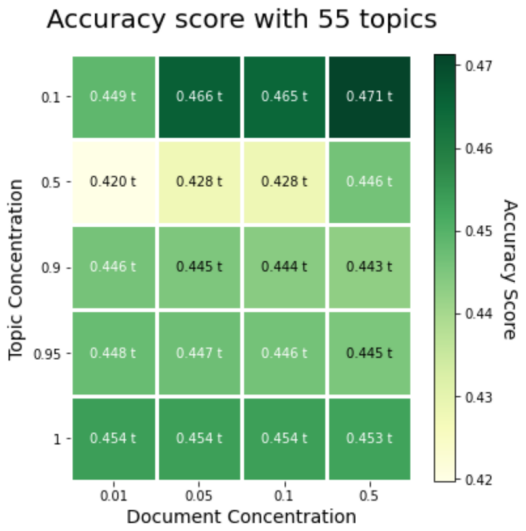

Figure 9: Accuracy score from the classifier for topics models with $K = 55$ topics and specific values for *alpha* and *beta*.

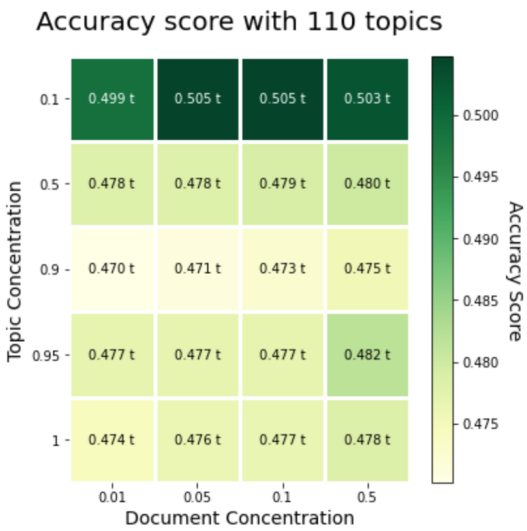

Figure 10: Number of different users by year who use their Camipro card at least one time during the concerned year.

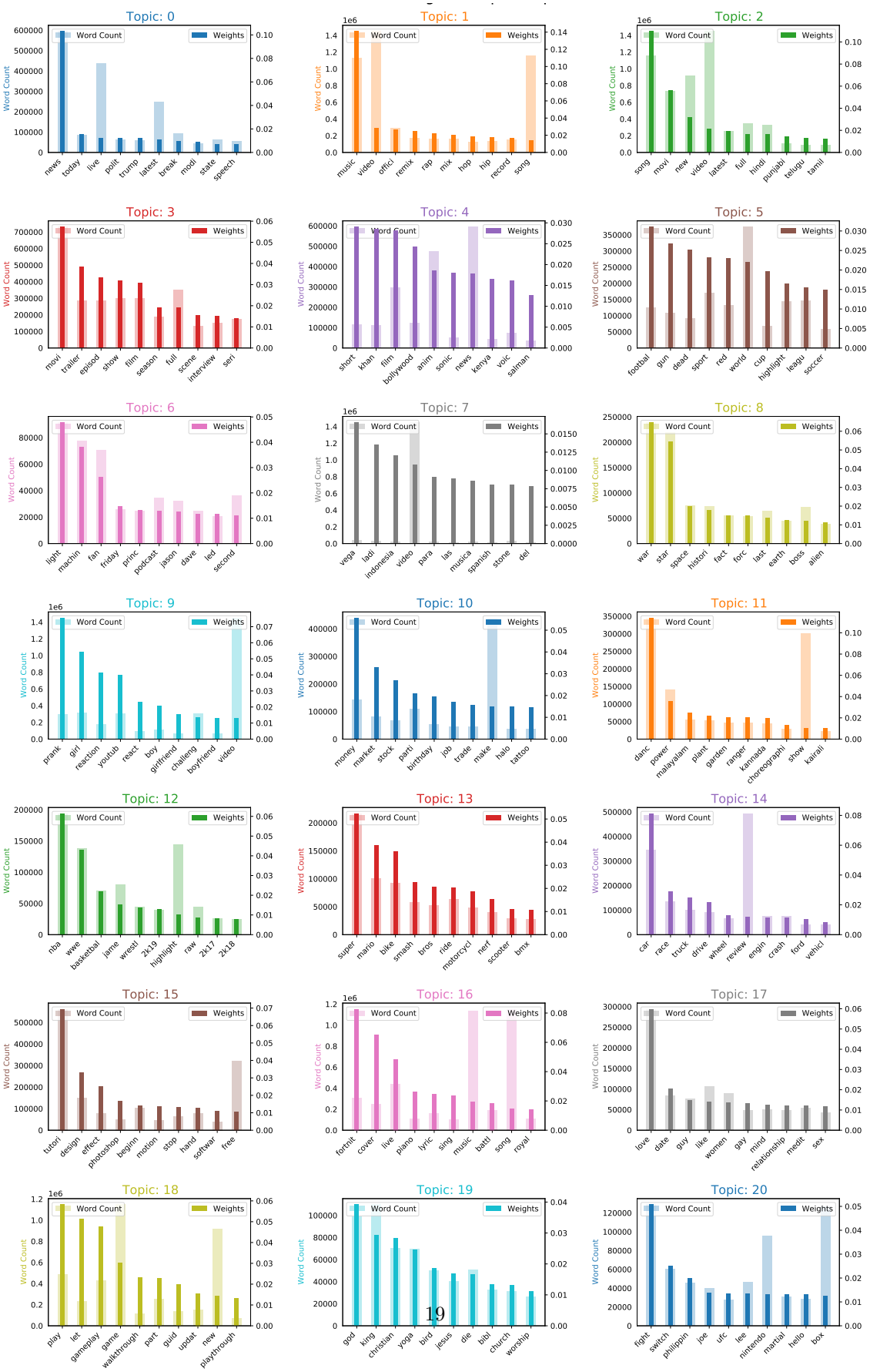

Figure 11: A histogram for the distribution of the videos by the number of views per video.

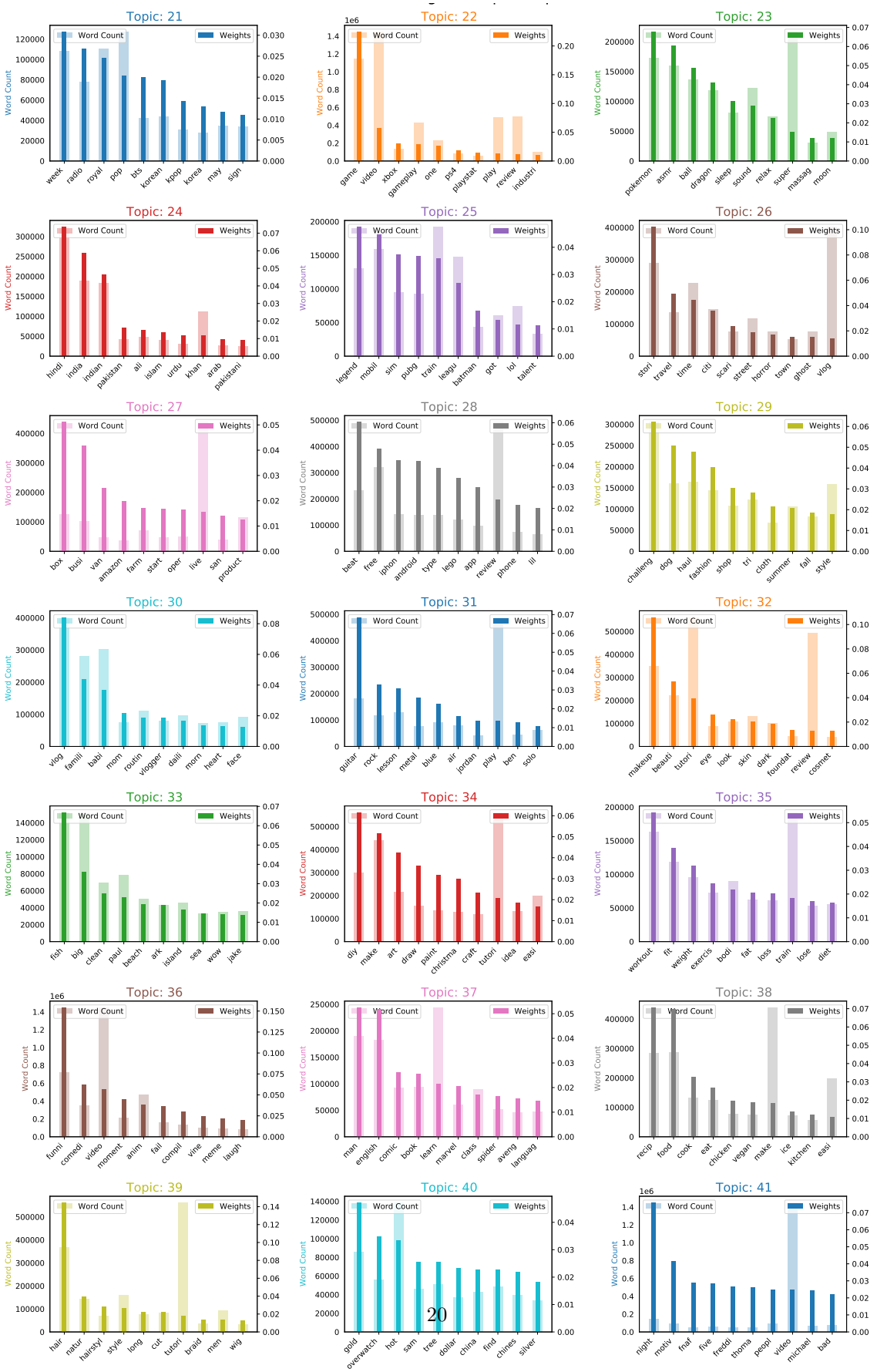

Figure 11: A histogram for the distribution of the videos by the number of views per video.

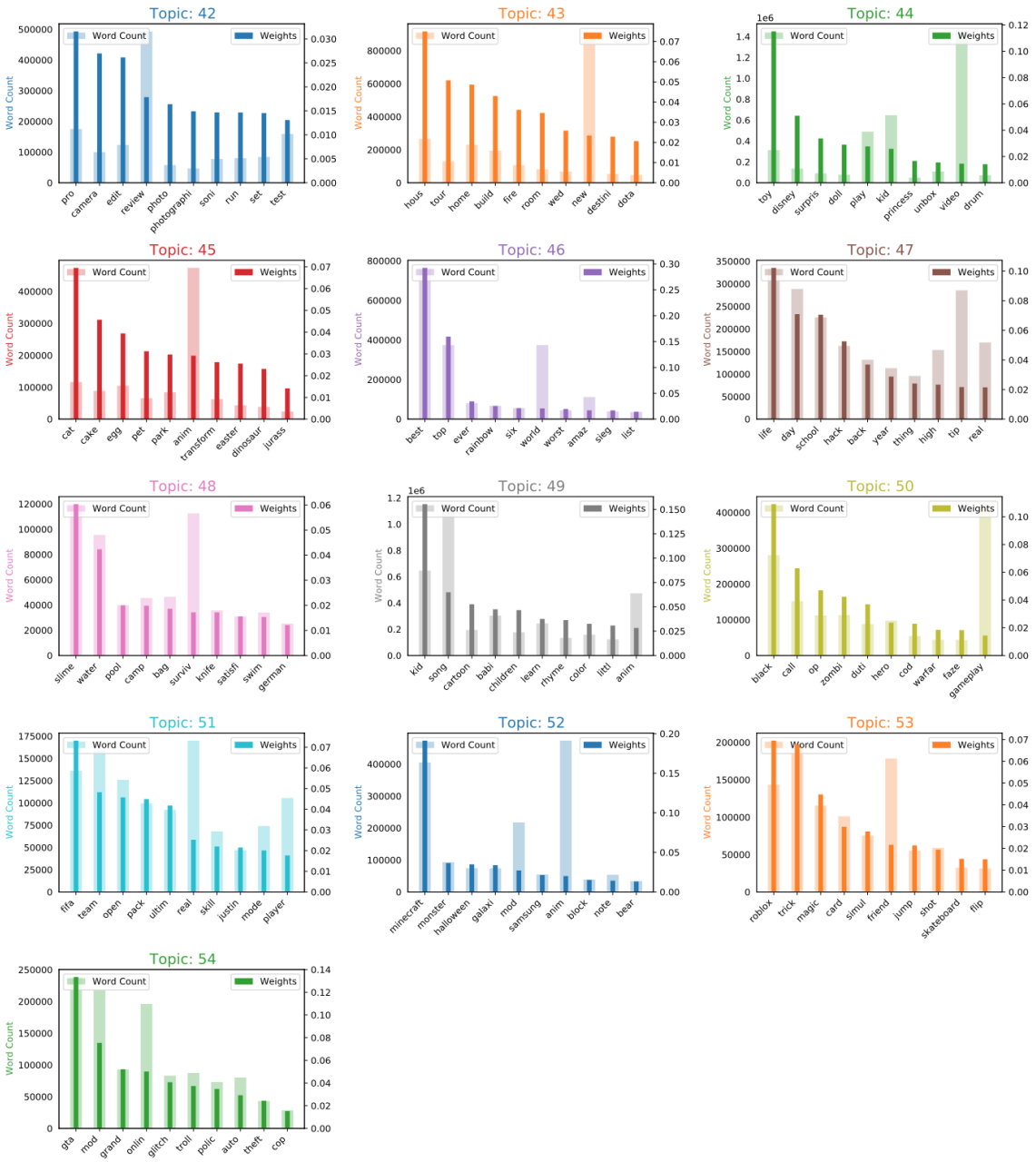

Figure 11: A histogram for the distribution of the videos by the number of views per video.