# Reverse-engineering Self-selection into YouTube Video Categories

*CAPP 30255: Advanced Machine Learning for Public Policy*

Ta-Yun Yang & Patrick Lavallee Delgado[*]

20 April 2020

## 1 Introduction

When forced to choose an identity, how do we reconcile everything about ourselves into the confines of a label? And despite everything that seems to separate some of us, how is it that we still identify the same? Broad categories are convenient for individuals to sort their preferences into cultures, politics, and other social phenomena, but messy because the ambiguity of language allows each to attach his or her own meaning to those labels. While there may exist quantitative markers with which to explain why an individual subscribes to one category and not another, qualitative language is interesting because its deliberate word choice demonstrates additional stated preference for substantiating the same.

More precisely, we are interested in how individuals self-select into arbitrary categories using other instances of self-expression. YouTube offers a simple version of this challenge. As a user uploads video content, he must describe the work in the title, description, tags, and category. The user may use any words he wishes, so long as he remains consistent with YouTube's community guidelines; but he is limited by the categories available and he must choose exactly one category. These categories are broad and several may apply to his video. For example, a funny video of a dog playing volleyball could conceivably exist in the "Pets & Animals", "Sports", and "Comedy" categories. Which does the user choose and why? We attempt recreate his decision using the language he chooses to describe his content along with examples of others in the YouTube community.

Our copy of the YouTube data collects the descriptions and activity statistics of the most-watched videos for every day between November 2017 and June 2018 by country. This is freely available from the YouTube API. The data has two levels: words in videos and videos in categories. Table 1 lists the 30 categories from which the user chooses. Table 2 summarizes the number of videos users upload by country and the size of the corpus of text available to our study.

[*] Candidates, MS Computational Analysis and Public Policy, {tayuny, pld}@uchicago.edu.

Table 1: YouTube video labels

| | | | | |
|---|---|---|---|---|
| Action/Adventure | Drama | Gaming | People & Blogs | Shows |
| Anime/Animation | Education | Horror | Pets & Animals | Sports |
| Autos & Vehicles | Entertainment | Howto & Style | Sci-Fi/Fantasy | Thriller |
| Classics | Family | Movies | Science & Technology | Trailers |
| Comedy | Film & Animation | Music | Short Movies | Travel & Events |
| Documentary | Foreign | News & Politics | Shorts | Videoblogging |

*Note: the United States has an additional category, Nonprofits & Activism.*

Table 2: Summary of observations in YouTube video data

| Country | Total videos | Unique videos | Average tag count | Unique tag words |
|---|---|---|---|---|
| Canada | 40881 | 24427 | 19.58 | 75459 |
| Germany | 40840 | 29627 | 17.97 | 109309 |
| France | 45090 | 31459 | 14.70 | 83825 |
| India | 37352 | 16307 | 18.76 | 49084 |
| Japan | 20523 | 1705 | | |
| Mexico | 40451 | 321 | | |
| Russia | 40739 | 1528 | | |
| South Korea | 34568 | 443 | | |
| United Kingdom | 38916 | 3272 | 18.00 | 21930 |
| United States | 40949 | 6351 | 19.74 | 31951 |

*Note: missing values were not available in time for submission of this proposal.*

In the sections that follow, we discuss the related work that is the foundation for our study and then the plan of action pursuing and evaluating the same.

## 2 Related work

Our challenge is fundamentally an exercise in *text segmentation*, grouping language into coherent topic clusters using the lexical cohesion that arises from the semantic relationships between words. An early attempt to group text with shared meaning is lexical chaining [17], which links nearby words on whether they exist in related thesaurus categories and evaluates the strength of the resulting chains on frequency and density. This inspired years of work on unsupervised text segmentation, which uses the frequency and co-occurrence of words to identify topic boundaries in a text. Among the first of these algorithms is TextTiling [10], which compares the lexical similarity of adjacent sentence groups from the words those sentences share, and finds a topic boundary where the similarity of words between those groups is low. The choice in topic boundary from several possibilities improves with Latent Semantic Analysis (LDA) [5], which uses principle component analysis to cluster the frequency of co-occurring words in order to reveal semantic dissimilarities between sentence groups.

Beyond the neatness of written text, this area of research also explores language with multiple participants at different times. Addressing this variation in meeting transcripts is the Lexical Cohesion-based Segmenter (LCSeg) algorithm [8], which identifies lexical chains on word frequency alone and compares the cosine similarity of lexical chains among adjacent sentence groups to identify potential topic boundaries in a manner similar to TextTiling. Other work extends LCSeg to asynchronous conversations in emails threads and blog comments [12], which draws paths to sequential fragments in different texts and consolidates topic clusters that LCSeg identifies with those that have a high cosine similarity among their sentences. A generative approach is the TopicTiling algorithm [19], which uses latent Dirichlet allocation (LSA) [3] to estimate topic-word and topic-document probability distributions from a corpus of text that can associate topics to words in a document; in an extension of TextTiling, it finds a topic boundary where the cosine similarity of topics between sentence groups is low.

We wish to emphasize the representation of words as it influences our conceptualization of how we establish semantic relationships in our work. The algorithms we describe so far map the vocabulary of a corpus to a matrix in which each dimension represents one word. The most popular implementations include bag of words (BOW) and n-grams. But, these representations do not preserve word order and the informative structure of language; text segmentation algorithms go to great lengths to recover semantic relationships. Improvements to text representations include increasing the size of the n-gram or even allowing n-grams of varying lengths to encode phrases [15]. Other approaches [2] revise the BOW representation and estimate the distribution of each word and predict that of new words. The experiment in the large variety of English texts from the Brown corpus (1,181,041 words) and Associated Press News (13,994,528) shows that the new model is significantly more efficient in terms of text perplexity comparing to traditional n-grams. Moreover, the improved versions of the text representation algorithms optimize their run time for more sophisticated text [13] [16].

Yet, as the size of the vocabulary increases, so do the number of dimensions to the point where the the accuracy of our estimates deteriorate and the calculation becomes intractable. A simple solution is TF-IDF, which preserves words with high frequency among all documents in a corpus to maintain those that offer the most information [18]. Dimensionality reduction techniques provide a feasible path to balance the trade-off between complexity and information preservation. A low-dimensioned projection can be derived via singular value decomposition (SVD) with limited loss of information from the text. Zhang, Yoshida, and Tang [21] compared the efficiency and accuracy of TF-IDF and latent semantic indexing (LSI). After processing 14,150 Chinese academic articles and 21,578 English news from Reuters, LSI method is most effective in terms of information retrieval from text classifications with support vector machines. However, the defect of dimensionality reduction is also clear. Since the low-dimensioned features are the projections from the origin text representation, the interpretability of the weights declines.

Recent advances in text segmentation use word embeddings, a departure from one-hot encodings that represent each word as its own dimension towards dense encodings that represent words as vectors in space. Training continuous vector representations of words with domain-specific text becomes its own task [14]. Among algorithms that demonstrate this improvement is GraphSeg [9], which draws a semantic relatedness graph of each sentence in a document to all others and creates segments composed of the adjacent sentences in maximal cliques; It calculates similarity of sentences as the sum of cosines of the embedding vector of each word in one sentence to that of each word in the other.

The discipline builds to deep neural networks, perhaps the best approach to the problem we pursue. In a very similar case recovering the subsection titles of Wikipedia articles, the Sector algorithm [1] learns an embedding of latent topics from ambiguous headings, segments the the document into coherent sections, and labels each section with a learned topic. It uses bidirectional long short-term memory (BLSTM) networks to generate the dense topic embedding matrix. This method allows us to "remember" previous information in a long sequence of text [11]. For the text segmentation phase, Sector traces movement in the topic embedding vector space over sentence groups and identifies a new section where that movement is fastest. For the section label decoding phase, it identifies the heading whose words have the strongest association with the topic embedding matrix. In a further departure from previous work, the end-to-end framework in Sector recasts text segmentation as a semi-supervised task: learning the semantic similarities of the text to define the scope of topics that in turn segment the text.

We are also aware of significant research in the area of text classification, applying labels to the identified text segments. The YouTube data is different than many natural language processing challenges in that it is not strictly sequential: while we expect some semantic coherence at the video level, the order of videos may not be meaningful at the category level. We can, however, make inference to the change in composition of a category because we have time series data. This is all to say that the text classification should not label a video based on the relatively arbitrary position it holds in the corpus, but solely on its language features. Among strategies is minimizing the sum of distances between n-gram frequency rankings of the targeted category and the document [4]. Since the method is simple and requires small amount of time, we will consider it as the baseline in comparison to more complicated classifiers, such as that proposed by the Sector algorithm.

# 3 Action plan

The purpose of this research is to automate and optimize the process of video classification for YouTube with its explicit text content. To be more specific, Title, Description and the Tags of the videos are used to train the classifiers determining whether a video is explicitly related to politics or not. Different text representation strategies, classifiers and combination of features are used to generate the predicted probability of each class.

After the labels are assigned or given to text entities, text features with corresponding class could be applied to various tasks. As the categories of the YouTube video are given in our data, we categorize the video with the explicit text attached to the video. Best practices offered by other researches suggest we use multiple groups of classifiers [6] [20]. These groups include logistic regression and naive Bayes from among probabilistic classifiers, random forest from among decision tree classifiers, as well as the nonlinear support vector machine and the stochastic gradient boosting methods [7].

Cross validation is implemented to guarantee the predictions are robust, which will be performed by dividing the original data into five group of equal size. For every run of the model, a group will be selected as the testing set to evaluate the performance of the models trained by the remained data. Finally, among all the possible classifiers, the one with the highest average precision, recall and F1 score will be selected as the optimal model.

## 3.1 Programming language and packages

| Purpose | Python packages |
|---|---|
| Supervised and Unsupervised Learning | scikit-learn, PyTorch |
| Deep Learning | PyTorch |
| Pre-processing Text | nltk, scikit-learn, PyTorch |
| Visualization | matplotlib, seaborn |

## 3.2 Text representation

In order to transform text content to trainable vectors, four popular methods will be implemented in our analysis. Bag of words, TF-IDF with N-grams contain the information of frequency for words and text entities where the absolute order is not meaningful. Moreover, considering the high dimension from these two methods, Latent Semantic Indexing is used as an improvement via dimensional reduction. However, when we are tokenizing the text in title and description beside tags, the relation of each sentence also matters. Therefore, word embedding with Word2Vec is used to involve implicit connections between sentences and documents into trainable matrix. The following table shows the summary of the text representation strategies and the column on which they are executed.

| Text Representation | Columns implemented |
|---|---|
| Bag of Words | Tags, Title, Description |
| TF-IDF | Tags, Title, Description |
| Latent Semantic Indexing | Tags, Title, Description |
| Word2Vec | Title, Description |

## 3.3 Classifiers

Following the guidelines from the related work, three different groups of algorithms will be executed and compared. Because of its simplicity, we use the n-gram model as our baseline, which requires the least computational capacity followed by tradition supervised learning models. Finally, we will optimize the algorithms with LSTM and BLSTM methods and evaluate whether the extra capacity spent in the YouTube video classification is cost-effective.

| Classifier family | Classifiers |
|---|---|
| Baseline | Simple N-gram distance |
| Supervised Learning Models | Logistic Regression, Random Forest<br>Support Vector Machine, Gradient Boosting |
| Deep Learning | Long Short Term Memory (LSTM), BLSTM |

## 3.4 Features

Since tags and titles are often directly related to the determined category, we are interested in their independent effect to the accuracy of the model. Moreover, concerning that tags might be missing for considerable number of videos which are not among the most popularity, it is worth discovering whether the models are useful to categorizing videos with only titles and descriptions, not tags. Finally, we include all of the information we have as the full model and evaluate whether if the improvement exceeds the extra time spent on the larger matrices.

| | Features | | |
|---|---|---|---|
| Model | Title | Description | Tags |
| 1 | | | X |
| 2 | X | | |
| 3 | X | X | |
| 4 | X | X | X |

## 3.5 Generalization and sub-categorization

Our initial attempts will be to train the model in a binary classification exercise: whether a video is or is not a particular YouTube category. We intend that our final product categorize videos into the full suite of options available. When we are satisfied with the performance of our model, and time permitting, we hope to extend its application to videos in different languages and regions. The expectation is comparable performance across languages. We would not be surprised to find worse performance of the model with data that spanned regions, for the local context with which people conceive of and speak about a category could vary and cause the model to over-generalize its category definitions.

### 3.6 Task Assignment

| | Ta-Yun | Patrick |
|---|---|---|
| Proposal | Text Representation,<br>Supervised Learning, Action Plan | Introduction, Unsupervised Learning<br>Data Summary |
| Text Representation | Bag of Words, TF-IDF | LSI, Word2Vec |
| Supervised Classifiers | Baseline and Supervised Models | — |
| Deep Learning | LSTM and BLSTM | LSTM and BLSTM |
| Final Report | Experimental Result, Visualization | Experimental Result, Literature Review |

# References

[1] Sebastian Arnold, Rudolf Schneider, Philippe Cudré-Mauroux, Felix A. Gers, and Alexander Löser. Sector: A neural model for coherent topic segmentation and classification. *Transactions of the Association for Computational Linguistics*, 7:169–184, 2019.

[2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[4] William Cavnar and John Trenkle. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1994.

[5] Freddy Y. Y. Choi, Peter Wiemer-Hastings, and Johanna Moore. Latent semantic analysis for text segmentation. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, 2001.

[6] Susan Dumais, John Platt, David Hecherman, and Mehran Sahami. Inductive learning algorithms and representations for text categorization. pages 148–155, November 1998.

[7] Jerome Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38:367–378, 02 2002.

[8] Michel Galley, Kathleen R. McKeown, Eric Fosler-Lussier, and Hongyan Jing. Discourse segmentation of multi-party conversation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 562–569, Sapporo, Japan, July 2003. Association for Computational Linguistics.

[9] Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. Unsupervised text segmentation using semantic relatedness graphs. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 125–130, Berlin, Germany, August 2016. Association for Computational Linguistics.

[10] Marti A. Hearst. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, 1997.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–80, 1997.

[12] Shafiq Joty, Giuseppe Carenini, and Raymond T. Ng. Topic segmentation and labeling in asynchronous conversations. *Journal of Artificial Intelligence Research*, 47:521–573, 2013.

[13] Thang Luong, Richard Socher, and Christopher Manning. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

[14] Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, pages 1–12, January 2013.

[15] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, page 3111–3119, 2013.

[16] Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 246–252, 2005.

[17] Jane Morris and Graeme Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48, 1991.

[18] Juan Enrique Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the First Instructional Conference on Machine Learning*, pages 133–142, December 2003.

[19] Martin Riedl and Chris Biemann. TopicTiling: A text segmentation algorithm based on LDA. In *Proceedings of ACL 2012 Student Research Workshop*, pages 37–42, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

[20] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[21] Wen Zhang, Taketoshi Yoshida, and Xijin Tang. A comparative study of tf*idf, lsi and

multi-words for text classification. *Expert Systems with Applications*, 38(3):2758–2765, 2011.