A Sampling Approach to Assessing Expected Reach of YouTube Videos Using Natural Language Processing

Jeremy Savill, Olaf Tomiuk
jeremysavill2023@u.northwestern.edu, otomiuk@gmail.com

2022-12-03

Abstract

Organizations spend millions of dollars on marketing and advertising with the goal of trying to gain the attention of their target audience - sometimes with limited success. The research conducted examines whether an organization can better reach its audience through analysis of the language for the content they generate. Our conclusion is that the combined use of information extraction, recurrent neural networks, and natural language processing can extract meaningful insights about the language used in YouTube videos and create accurate predictions around how many people the videos reach. More specifically, this research explores the development of an open-source generalizable process for parsing explainer videos using PyTube and for creating meaningful context from how-to videos using Keras/Tensorflow. These techniques combined allow for the assessment of reach given the language being used, and for providing a score to rank the reach of the video being evaluated with the top performing model achieving 75% accuracy in the test data set.

Keywords: Knowledge graph, natural language processing, information extraction, viral videos

Introduction

The internet is one of the primary sources for sales and marketing campaigns. This research analyzes the language used in YouTube "How to change a car battery" videos to determine if it impacts the reach a video has.

The model is designed to identify the key characteristics of language used in the how-to videos to help make videos go viral or not. The model converts language to text and generates triples and knowledge graphs to extract key subject, predicate, and object combinations. Pytube is used to extract metadata and video transcripts while Keras within Tensorflow is used for natural language processing. Natural language processing models are combined with metadata from the videos to successfully assess reach. Future research could include video/image analysis, but is currently out of scope.

Literature Review

Much of this research builds upon the tools developed through the Stanford CoreNLP systems. Written in Java, the core pipeline intakes a corpus of raw text and can add several annotations based on the needs of the researchers including the tokenization and lemmatization of words, parts of speech tagging, and named entity recognition. Released as open source software in 2010, the Stanford researchers aimed to "quickly and painlessly get linguistic annotations… behind a common API… [with] a minimal conceptual footprint, so the system is easy to learn" (Manning, 2014).

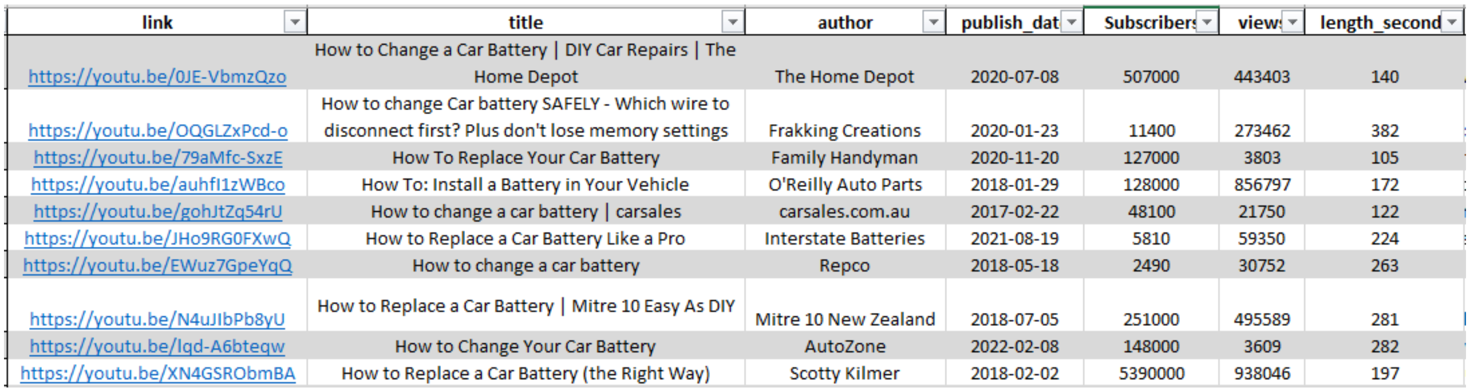
The development of knowledge graphs for model training and development is accomplished through the open domain information extraction (open IE) application of the CoreNLP pipeline. This extraction produces tuples in the form of two arguments and a relationship between them creating a triple that is visually represented in the knowledge graph by a subject node, object node, and relationship edge line connecting them. A team of researchers from the University of Manchester found several limitations with this method including that the "systems must rely on unsupervised extraction strategies … [of] possible relations of interest … automatically detected while making only a single pass over the corpus" (Niklaus, 2018).

Another team of Stanford researchers led by Gabor Angeli quickly observed that the triples extracted were of limited use in situations where longer sentences that have multiple self-contained clauses were not being captured. Their solution, and one that should be under consideration for this research, was to “pre-process the sentence in a linguistically motivated way to produce coherent clauses which are logically entailed to the original sentence and easy to segment into open IE triples” (Angeli, 2015). This underlines the importance of exploratory data analysis and processing of corpus text to produce meaningful data patterns for the model.

Data

Subject matter videos available on YouTube were hand-selected for this project. To focus modeling efforts in a single concentration area, 'How to' videos related to auto parts were chosen.

Metadata associated with each video was extracted using the Python package - Pytube. The metadata captured includes length of video, publish date, number of views, and number of subscribers. The number of subscribers were not retrievable using Pytube, so they were populated manually. The following table shows a snapshot of metadata captured for the first 10 videos about 'How to change a car battery'.



link	title	author	publish_dat	Subscriber	view	length_second
https://youtu.be/0JE-VbmzQzo	How to Change a Car Battery | DIY Car Repairs | The Home Depot	The Home Depot	2020-07-08	507000	443403	140
https://youtu.be/OQGLZxPcd-o	How to change Car battery SAFELY - Which wire to disconnect first? Plus don't lose memory settings	Frakking Creations	2020-01-23	11400	273462	382
https://youtu.be/79aMfc-SxzE	How To Replace Your Car Battery	Family Handyman	2020-11-20	127000	3803	105
https://youtu.be/auhfI1zWBco	How To: Install a Battery in Your Vehicle	O'Reilly Auto Parts	2018-01-29	128000	856797	172
https://youtu.be/gohJtZq54rU	How to change a car battery | carsales	carsales.com.au	2017-02-22	48100	21750	122
https://youtu.be/JHo9RG0FXwQ	How to Replace a Car Battery Like a Pro	Interstate Batteries	2021-08-19	5810	59350	224
https://youtu.be/EWuz7GpeYqQ	How to change a car battery	Repco	2018-05-18	2490	30752	263
https://youtu.be/N4uJIbPb8yU	How to Replace a Car Battery | Mitre 10 Easy As DIY	Mitre 10 New Zealand	2018-07-05	251000	495589	281
https://youtu.be/lqd-A6bteqw	How to Change Your Car Battery	AutoZone	2022-02-08	148000	3609	282
https://youtu.be/XN4GSRObmBA	How to Replace a Car Battery (the Right Way)	Scotty Kilmer	2018-02-02	5390000	938046	197

Figure 2. Metadata from YouTube videos

Video transcripts were retrieved using YouTube Transcript/Subtitle API. This API returns subtitles for YouTube videos given the YouTube video ID. When music is playing in the video it is shown as ‘[music]’ in the transcription text. Since the focus of this project is on language for creating knowledge graphs, transcripts are cleansed by removing the word ‘[music]’. Transcripts are cleansed by performing lemmatization, removing stop words and contractions. Data visualization was performed on the 10 most frequent words in each video. Appendix B shows the scatterplot with words on x-axis and number of occurrences on y-axis. The color of the dot notates the video link name, and size of the dot represents the number of 'likes' for that video.

A knowledge graph is a combination of subject, object and predicate that constitutes a semantic triple. Subject and object are represented as nodes in the graph while the predicate describes the relationship between the nodes. Stanford OpenIE, which is a part of Stanford CoreNLP, is a natural language processing library written in Java. A python wrapper for Stanford OpenIE, authored by Philippe Remy, is used to extract triples from video transcripts. The Python package NetworkX is used to visualize knowledge graphs.

The generated triples and the original transcript were compared manually to check if the semantic triples were giving the correct reasoning. Several triples were missed, and wrong relationships were created in the knowledge graphs primarily because the whole transcript was one long sentence without any punctuation.

A bidirectional recurrent neural network model with attention mechanism for restoring missing punctuation in unsegmented text developed by Ottokar and Tanel (2016) was used to add punctuation to the transcripts. Triples generated after punctuating were able to correct several misconceptions of relationships between subject and objects. Figure 2 shows a knowledge graph visual created from a YouTube video on 'How to change a car battery'. Appendix – A shows its corresponding triples in tabular format.

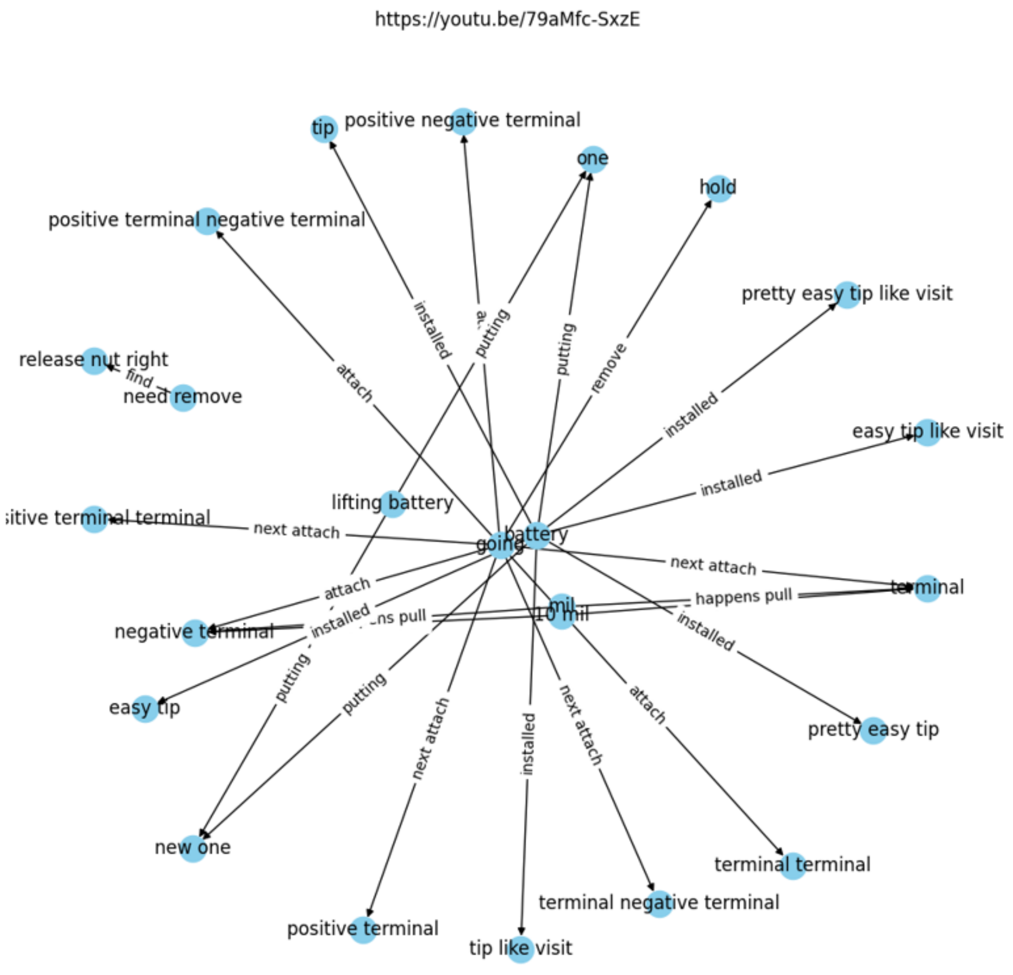


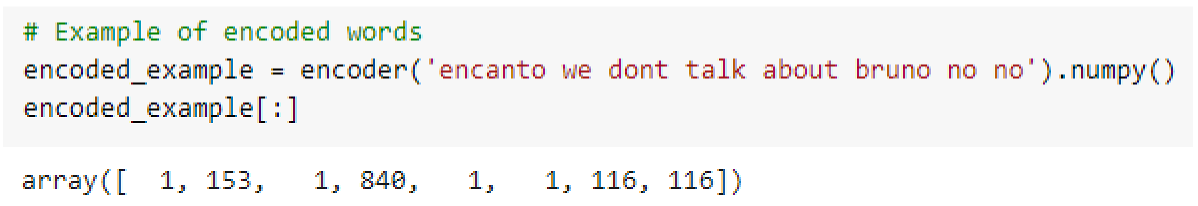
Figure 3. Knowledge graph for YouTube video ID - 79aMfc-SxzE

Methods

When setting up the project for experimentation with the data developed from generated knowledge graphs, the first step is to define the target flag variable upon which the model makes its prediction. The flag variable is created by taking the number of views and dividing them by the number of subscribers to create a ‘Reach’ score. The higher the ‘Reach’ score, the better the video rates in getting views against what would be expected from the number of subscribers the author of the video has. Reach was highly variable, with a low score of 0.02 and a high score of 619. To reduce this variation, the scores were banded into quartiles with 4 category ratings that could are associated with descriptive ratings of 'poor', 'okay', 'good', and 'excellent'. This banding technique also eliminates the issue of unequal sampling bias. The higher the score or rating, the more likely the model believes the video maximally reaches its audience controlling for the number of subscribers with the language contained in the video.

With the target flag variable and text of the video copied into a new data frame, the SKLearn train_test_split package is applied to the data frame to provide the model a set of videos to train on, a validation set to check against training data, and a test set of new data the model has not seen to make predictions. There are 40 total videos which were split into 32 training records, 4 validation records, and 4 test records. The dataframe is then converted to a TensorFlow dataset and the data pipelines are created by batching, shuffling, and optimizing the TensorFlow slices.

Finally, the transcript must be vectorized before a model can use the data for predictions. The Keras package within TensorFlow contains a TextVectorization encoder to adapt text into a one-dimensional vector. There are numerous advantages to this method including the ability to alter the maximum vocabulary size, standardize text by removing capitalization and punctuation, and padding extra vocabulary tokens if more vocabulary is needed. See image below for an example output array of an encoded sentence.



```
# Example of encoded words
encoded_example = encoder('encanto we dont talk about bruno no no').numpy()
encoded_example[:]
```

```
array([  1, 153,   1, 840,   1,   1, 116, 116])
```

Figure 3. Code and output example of vectorized and encoded vocabulary

For the model architecture itself, the initial embedding layer is the length of the defined vocabulary with the initial number of nodes an experimental consideration. The complete lemmatized vocabulary of the 40 video transcript set is 1,755 words and adjusting this vocabulary is another experimental consideration. A recurrent neural network (RNN) with two bidirectional long short-term memory layers enables the model to understand the text both forward and backward to better understand word associations. The first bidirectional layer includes the return sequences which the second bidirectional layer interprets along with the original sequence. A dense layer is added to prepare the model for its output classification activated using the ReLU function. The final output layer utilizes the SoftMax activation function to provide a probability of what the model believes the rating of the video should be with each of the output nodes receiving a score that collectively adds up to one. The highest probability score is the model's prediction for the expected Reach of the video based on the transcript. The model compiles with the Adam optimizer and the loss function is measured utilizing sparse categorical cross entropy.

Other considerations for model optimization include adding dropout layers which will inactivate a certain percentage of nodes in a layer to reduce model overfitting. The model is set to train for 50 epochs. However, the model is also equipped with an early stopping procedure that will end the training epochs early if the validation accuracy score does not improve after 3 training iterations.

Success of the model is observed by measuring the training and validation accuracy metrics generated by the model. Diverging values indicate a degree of overfitting whereby a model is less able to interpret new information from the corpus of data upon which the model is trained. A confusion matrix plotting predicted values versus actual values can help identify areas where the model is weaker and provide insight into what can be adjusted to improve model accuracy.

Results, Analysis and Interpretation

Ten models were developed and trained on the 32 video training transcripts. The hyperparameters that were adjusted in service of improving the model were the size of the vocabulary, the number of nodes in the embedding layer which also affected the total number of parameters, and the dropout rate for the LSTM layers. The table below summarizes these results.

Model	Vocab Size	Nodes in Embedding Layer	Total Parameters	Node Dropout Rate	Training set accuracy	Validation set accuracy	Test set accuracy
1	500	32	106,020	0%	72%	50%	0%
2	500	32	106,020	20%	75%	50%	0%
3	1,000	48	211,060	20%	47%	50%	50%
4	1,000	48	211,060	40%	75%	75%	75%
5	1,000	64	415,908	20%	53%	25%	0%
6	1,000	64	415,908	40%	88%	75%	25%
7	1,250	64	431,908	20%	84%	50%	25%
8	1,250	64	431,908	40%	75%	50%	25%
9	1,000	128	1,201,860	20%	63%	50%	25%
10	1,000	64	406,212	40%	72%	50%	50%

During these experiments, an overfitted model was the single greatest impediment to achieving positive results. This is a drawback to an LSTM model as retaining information from previous layers makes it more prone to "memorizing" training data which hinders performance on the testing data set. The figure below shows the AUC/ROC curves of the training and validation accuracy scores of model 1.

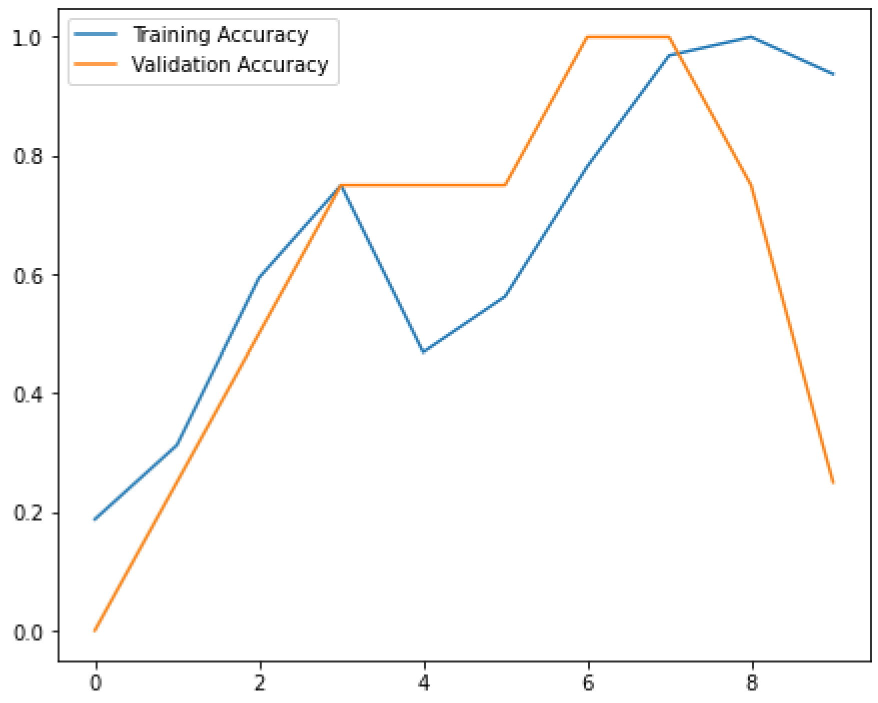


Figure 4: Model 1 suffers from overfitting

The divergence of the two lines indicates overfitting and the model's 25% accuracy rate on the test data set confirms this. Most of the experimental models suffered the same overfitting problem despite a dropout rate of up to 40% for some models.

The second issue was determining the ideal vocabulary size needed to train the model. With 1,755 lemmatized vocabulary words across the corpus, intervals of 500 were selected for experimentation. In true Goldilocks fashion, the middle value of 1,000 vocabulary words yielded the most positive results. Models utilizing only the top 500 words were not able to make the associations needed to improve on the training data set, much less the testing data set. Models with a vocabulary length of 1,500 words appeared to have too much data at its disposal and could only "memorize" the training data and suffered when predicting the test data set. The models training with 1,000 vocabulary words performed best with model 2 achieving 75% accuracy on the training, validation, and testing data sets. The AUC/ROC curve for model 2 is shown below.

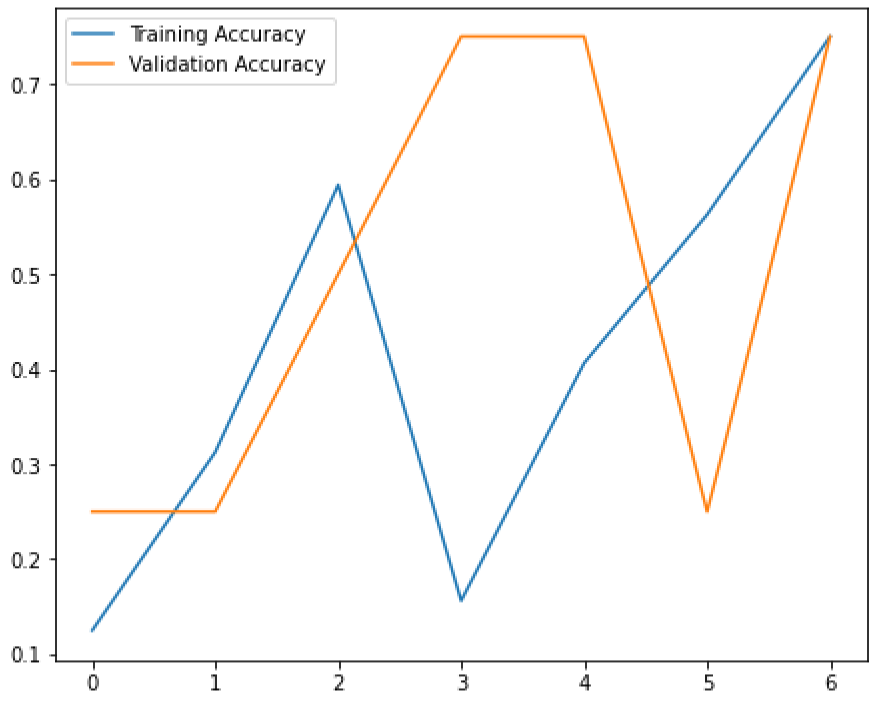
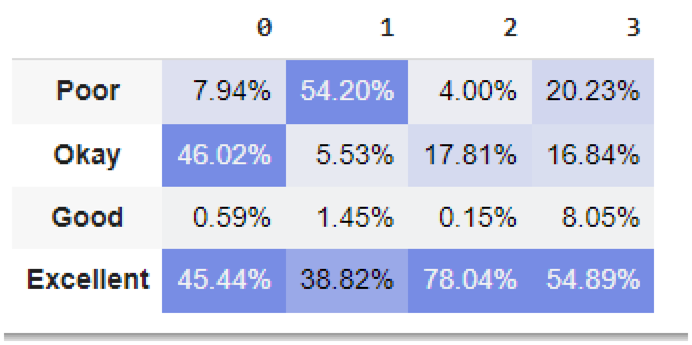


Figure 5: Model 4's Training and Validation Accuracy Scores

While the data is somewhat noisy, the values of both the training and validation accuracy scores end up converging towards the 75% mark. Some of this is attributed to a small set of 4 validation videos so there can be only 5 validation test scores in increments of 0.25 from 0 to 1.

The relevant questions become, how can this data be utilized and what does it mean? Model number 4 outperformed what would be expected from pure chance alone. This suggests that the model is able to make associations between how well the videos are conveying the information and the expected reach of the video based on the number of subscribers the author currently has. This makes the model useful as a sanity check for organizations that would feed the transcript to the model and be reasonably sure the resources spent to produce the content would be most effective. The figure below shows the weighted confusion matrix of Model 10's probability of each rating for each of the test videos.



	0	1	2	3
Poor	7.94%	54.20%	4.00%	20.23%
Okay	46.02%	5.53%	17.81%	16.84%
Good	0.59%	1.45%	0.15%	8.05%
Excellent	45.44%	38.82%	78.04%	54.89%

Figure 6: Weighted Confusion Matrix of Model 10

Model 10's first 2 predictions were accurate, but it can be seen that the model tends to overweight the highest classification and underweight the second highest classification. This could be due to the fact that the entire set of videos are all based on the same subject and expanding the content of subjects while retaining the Reach variable could lead to increased differentiation and more accurate results. While not yet a minimum viable product, the model serves as a solid proof of concept which merits further development and consideration.

Conclusions

An organization's ability to reach its target market is one of the most critical components for its success. As such, many organizations spend millions of dollars per year trying to find better ways to connect with their customers. Our research set out to help these organizations by determining if natural language processing can be used to help predict whether or not a video will go viral. While results were encouraging, a more robust approach would likely include a larger sample size and a more broad spectrum of subject matter. Furthermore, while natural language processing alone has the potential for creating an accurate model, a model that considers computer vision would almost certainly provide improved accuracy given the question we've set out to address.

Directions for future work

Future work expands upon the subject matter and video transcripts included in the corpus. Transformer technology such as BERT, which uses matrix multiplication to reweight relationships between words in a sentence, may produce more accurate results than the RNN bidirectional model which only reads the text data forwards and backwards. Incorporating data from sources other than YouTube, such as Instagram or TikTok, may provide a more generalizable application into the overall concept of a viral video. Combining video elements through a convolutional neural network with the existing natural language processing is a logical next step to help improve model comprehension and prediction accuracy. Also, there are experimental opportunities to utilize the model architecture to make other predictions such as the classification of the core competency subject or the number of likes a video could generate based on its core competency rating and other included metrics. The number of possible applications for this model architecture make for a compelling research subject.

References:

Angeli, Gabor, Melvin Premkumar, and Christopher Manning. n.d. "Leveraging Linguistic Structure for Open Domain Information Extraction." Accessed October 19, 2022. https://nlp.stanford.edu/pubs/2015angeli-openie.pdf.

"Biggest Social Media Platforms 2022." n.d. Statista. Accessed October 22, 2022. https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/.

Bosselut, Antoine, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. "COMET: Commonsense Transformers for Automatic Knowledge Graph Construction." ArXiv:1906.05317 [Cs], June. https://arxiv.org/abs/1906.05317.

Goodrow, Cristos. 2017. "You Know What's Cool? A Billion Hours." Blog.Youtube (blog). YouTube Official Blog. February 27, 2017. https://blog.youtube/news-and-events/you-know-whats-cool-billion-hours/.

Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McCloskey. 2014. "The Stanford CoreNLP Natural Language Processing Toolkit." in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60

Martinez-Rodriguez, Jose L., Ivan Lopez-Arevalo, and Ana B. Rios-Alvarado. 2018. "OpenIE-Based Approach for Knowledge Graph Construction from Text." Expert Systems with Applications 113 (December): 339–55. https://doi.org/10.1016/j.eswa.2018.07.017.

Mondal, Ishani, Yufang Hou, and Charles Jochim. "End-to-End Construction of NLP Knowledge Graph." ACL Anthology. Accessed October 12, 2022. https://aclanthology.org/2021.findings-acl.165/.

Niklaus, Christina, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. "A Survey on Open Information Extraction." ACLWeb. Santa Fe, New Mexico, USA: Association for Computational Linguistics. August 1, 2018. https://aclanthology.org/C18-1326/.

Pujara, Jay, Hui Miao, Lise Getoor, and William Cohen. 2013. "Knowledge Graph Identification." In Advanced Information Systems Engineering, 542–57. Berlin, Heidelberg: Springer Berlin Heidelberg.

Stanovsky, Gabriel, and Ido Dagan. 2016. "Creating a Large Benchmark for Open Information Extraction." Association for Computational Linguistics. https://aclanthology.org/D16-1252.pdf.

Tilk, Ottokar and Tanel Alumäe. "Bidirectional Recurrent Neural Network with Attention Mechanism for Punctuation Restoration." in *Proceedings of the 17th Annual Conference of the International Speech Communication Association,* 2016, 3047-3051. https://doi.org/10.21437/Interspeech.2016-1517.

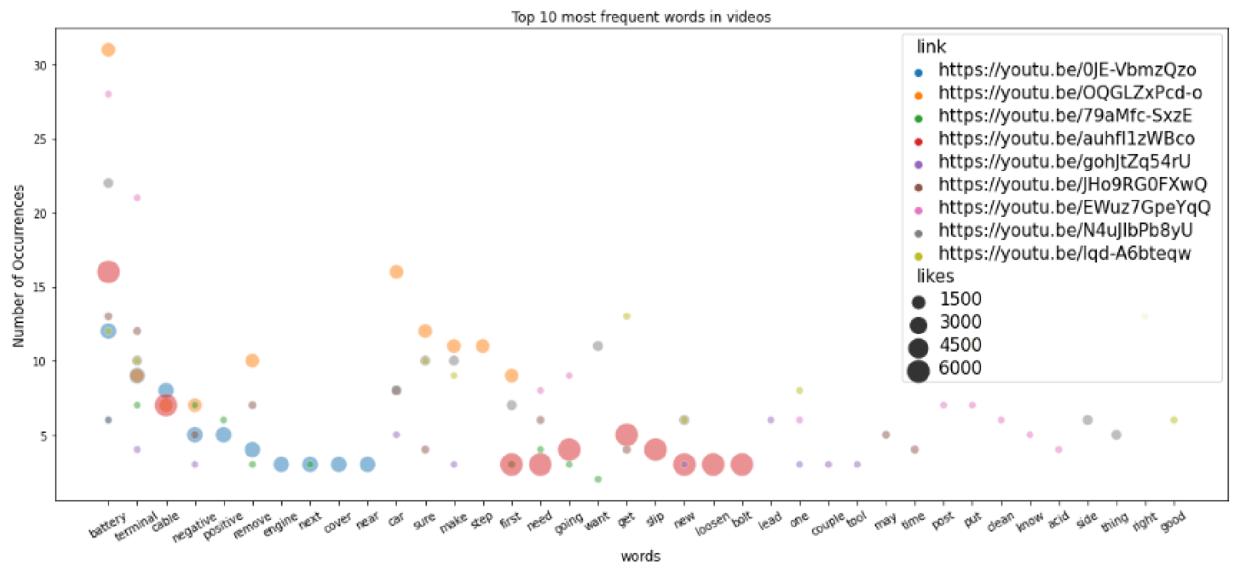
Appendix - A:

Triples for YouTube video ID - 79aMfc-SxzE

subject	object	predicate
10 mil	terminal	happens pull
mil	negative terminal	happens pull
mil	terminal	happens pull
10 mil	negative terminal	happens pull
need remove	release nut right	find
going	hold	next remove
going	hold	remove
lifting battery	one	putting
battery	new one	putting
battery	one	putting
lifting battery	new one	putting
going	positive negative terminal	next attach
going	positive terminal terminal	attach
going	terminal terminal	next attach
going	terminal	attach
going	negative terminal	next attach
going	positive negative terminal	attach
going	positive terminal	attach
going	positive terminal negative terminal	next attach
going	positive terminal negative terminal	attach
going	positive terminal terminal	next attach

subject	object	predicate
going	terminal terminal	attach
going	positive terminal	next attach
going	terminal negative terminal	attach
going	negative terminal	attach
going	terminal negative terminal	next attach
going	terminal	next attach
battery	easy tip like visit	installed
battery	pretty easy tip	installed
battery	pretty easy tip like visit	installed
battery	tip like visit	installed

Appendix - B:


Top 10 most frequent words in videos
Number of Occurrences
words
link
https://youtu.be/0JE-VbmzQzo
https://youtu.be/OQGLZxPcd-o
https://youtu.be/79aMfc-SxzE
https://youtu.be/auhfl1zWBco
https://youtu.be/gohJtZq54rU
https://youtu.be/JHo9RG0FXwQ
https://youtu.be/EWuz7GpeYqQ
https://youtu.be/N4uJlbPb8yU
https://youtu.be/lqd-A6bteqw
likes
battery
terminal
cable
negative
positive
remove
engine
next
cover
near
car
sure
make
step
first
need
going
want
get
slip
new
loosen
bolt
lead
one
couple
tool
may
time
post
put
clean
know
acid
side
thing
right
good