vExplorer: A Search Method to Find Relevant YouTube Videos for Health Researchers

Hillol Sarker, PhD, Murtaza Dhuliawala, MS, Nicholas Fay, MS, Amar Das, MD, PhD
IBM Research, Cambridge, MA, USA

Abstract

Patients and caregivers are increasingly using online video sharing services, such as YouTube, to document experiences with a health condition. Despite the richness of shared information and social commentaries in such videos, there have been few systematic studies focused on the health content of such media. Finding related videos on YouTube can be challenging for researchers because of inadequate search and ranking methods. In this paper, we present initial work on an unsupervised information retrieval method that supports the mental model of a researcher while he or she is exploring a topic area and lets the user examine extracted metadata to identify relevant videos. An experimental comparison and evaluation of our approach to YouTube for searching for videos on autism personal stories finds that using title, description, and tags of the videos in our approach produces more relevant videos than the ones that YouTube suggests.

1 Introduction

Many patients and caregivers are using online social media platforms, such as YouTube and Facebook, to document and share experiences of a health condition and its treatments. Comments they receive on such materials may provide feedback and create social support from other patients or caregivers. This use of social media also opens up a new opportunity for researchers to find and analyze such health-related content and conduct surveillance and analysis of the topic area.

For example, a researcher may want to use videotape examples of verbal protests in children diagnosed with Autism Spectrum Disorder (ASD) to develop methods of identifying such behaviors. Traditionally, the researcher would need to create a study protocol to obtain human subjects approval and recruit children for such research. The subsequent review by an Institutional Review Board may impose significant time delays, and the researcher may be required to ensure the privacy of the study subject by removing Protected Health Information (PHI)[1] from any captured video. In contrast, examples of such behaviors in video materials provided in public online social media platforms, such as YouTube, are exempt from human subjects review requirements. Thus, researchers may readily search videos to obtain relevant examples and, as a result of the reduced overhead, may be able to more rapidly undertake their work.

We note several challenges, however, in using YouTube as a source of health-related videos. Researchers first need to understand how the information retrieval space is structured and what terms and their combinations are likely to provide an optimal search. The iterative steps of applying search terms, sorting out results and keeping track of relevant videos can be time consuming. Second, researchers who discover a single YouTube video that contains the content he or she needs may face problems finding a set of similar videos. YouTube can suggest related videos but their similarity may not be based on the featured health behavior. Instead, it may be ranked based on unrelated factors, such as the usage patterns of the user.

To address such challenges, we have developed a system, called Video Explorer (or vExplorer), that allows a user to input an initial seed video and discover related videos based on extracted metadata. The system uses terms in the title, description, user comments, and tags to obtain relevant concepts and prioritize them as search tokens. The system generates new search strings, accesses the YouTube API, and suggests a list of related videos based on ranked similarity. To the best of our knowledge, this is the first method that searches metadata on YouTube videos from an initial seed video and that then generates relevant keywords to search for relevant videos in an unsupervised manner. We hypothesize that the system better mirrors the mental model of a researcher when he or she is trying to find relevant health-related videos based on a seed video. In this paper, we describe the methods we use for the search and ranking process in vExplorer. We then create an experimental evaluation using videos on ASD-related personal stories and provide results comparing the relevance of retrieved videos for different configurations of vExplorer to that of the standard YouTube search engine.

2 Related Works

Machine learning relies heavily on the quantity and the quality of the labeled dataset. Collection of this dataset can incur significant time and money. Researchers in numerous cases have shared their annotated datasets to help the research community. These datasets include annotated text[2–4], image[5–7], audio[8–10], video[11,12], and wearable sensors[13,14]. These datasets are static, however, and only limited to a specific problem space. User-generated content on online platforms, such as YouTube, are a dynamic source of datasets. Users are continuously adding new and diverse content. Active user communities augment this content by providing feedback, such as comments and likes. We propose a new approach to create datasets based on the metadata of the video content and to use the metadata in a coherent way for search.

Clustering text-based data of users undertaking video search has been studied widely. The majority of the cases researchers start with are by having a given corpus. The problem then reduces to the categorization of these documents. Xiao et al.,[15] proposed a hierarchical approach to cluster the videos that are nearly identical based on the content of the video, overlooking the user comments of the online video. Another method is to use a scatter/gather[16] approach through an iteration of two steps. The first step scatters the entire corpus. The second step gathers only those documents which are relevant to the current concept. This approach is relatively faster and more appropriate for browsing a large database[17]. Daan et al.,[18] have proposed a Markov decision process method to mine the subtitles of live television broadcasts and suggest relevant video contents to the user. An another approach is to use agglomerative and hierarchical clustering[17,19], which starts with each document as an individual cluster. Based on best-case, average-case, or worst-case pairwise similarity score, the method takes two clusters and merges them into one cluster at a time. At the end, the entire corpus becomes one big cluster providing an explainable and intuitive category of clusters. In cases when we have prior knowledge of the number of clusters, k-means or k-medoid can be a candidate clustering approach. However, this approach is sensitive to the selection of initial seeds. In addition, computation may take indefinite time due to nondeterministic iterations. In a similar case, when we have prior knowledge about the number of topics in the corpus, we can use topic model based approaches. Latent Dirichlet Allocation (LDA)[20] and similar algorithms can generate the distribution of words in topics and the distribution of topics in a corpus. They provide, as a result, a human interpretable distribution of words and topics. All of these prior approaches start with the assumption that the corpus is given and the computational process proceeds bottom-up. On the other hand, vExplorer is a top-down approach that develops the corpus in a hierarchical and iterative manner.

Other works related to our evaluation of videos for ASD research have attempted to discover similar types of YouTube videos to assess a research hypothesis. Vincent et al.[21], tested a hypothesis that it is feasible to use unstructured home videos for the early detection of autism, outside of the clinical setting. As a potential source of home video data, the authors used autism-related YouTube videos. They then applied their domain knowledge to manually search YouTube videos, identify the related videos, and develop a dataset. Non-clinical raters were able to identify the cases of autism with high accuracy, proving the validity of the hypothesis. Our approach can augment this related work[21] by discovering related videos as well as the search strings in an unsupervised way.

3 Method

In Figure 1, we provide a brief overview of the proposed system from the perspective of the user. The user first selects a YouTube video which he or she finds related to the search content and seeks to find related videos. The system uses this video as a seed video, finds the most important descriptor tokens for the video, searches YouTube on behalf of the user, scores search results, and provides the user with a list of relevant videos. In this section, we discuss the computational methods used in our system.

Search Token Extraction: The search token extraction begins with the seed video. The system makes API calls to the YouTube data API service to obtain the title, description, tags, comments, and replies of the seed video. We tokenize the text and apply the Penn Treebank (PTB) based part of speech tags[22]. On completion we retain nouns, verbs, and adjectives along with their different forms. Common stop words or slang terms are filtered out. The remaining terms are lemmatized and we create a bag of words representation (C). However, we keep a mapping of source/zone of those tokens (e.g., title, description, tags, and comments) so that we can evaluate their relative importance. Each video, including the seed video, is represented as a vector of the token frequencies (TF). Token frequencies of the vector

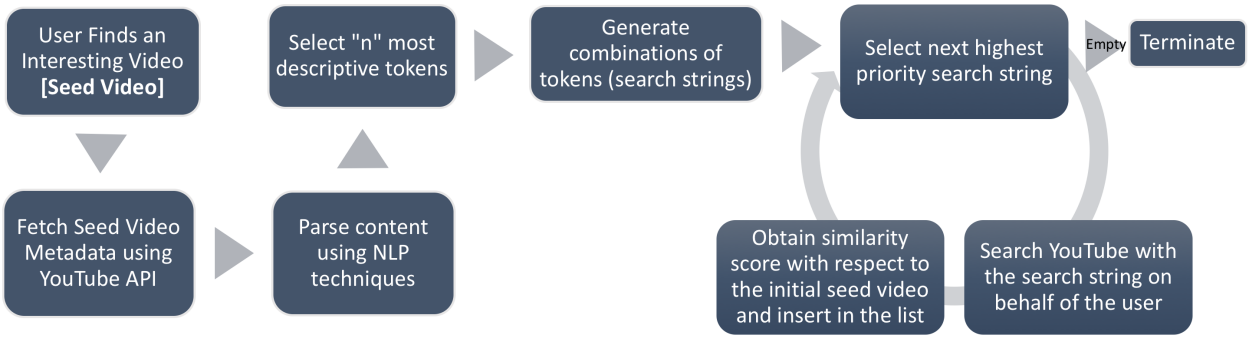

Figure 1: The overview of the system. User provides a video. System generates an ordered list of possible tokens based on title, description, tags, and user comments. The system generates different search strings by combination of the selected tokens, searches YouTube on behalf of the user and provides user with a list of relevant videos.

are normalized using formula $Augmented\_TF(t, Z) = 0.5 + 0.5 * \frac{TF(t)}{max\{TF(t')|t' \in Z\}}$, where Z is the set of terms in one of the four possible zones (title, description, tag, and comment). Coefficient of each token is the mean of four $Augmented_TF$s for the four zones.

Scoring (Similarity Measure): Based on the normalized frequency of the tokens in the seed video, we identify the *n* most significant tokens. The system creates different combinations of these *n* tokens to form candidate search strings. The number of such combinations are exponentially large. We prioritize the exploration process by considering the most relevant search string first based on a similarity score. A YouTube API call produces *n* (50) videos for a certain search string. Each video (including the seed video) is represented as a vector of normalized term frequencies of 100 most important tokens. We compute pairwise cosine similarity between the seed video vector (SVV) and each of the search result videos' vector (RVV). Limiting SVV and RVV to a length of 100 introduces many uncommon words. We take a union of the two sets and form a new pair of vectors with coefficient 0 in case the word is not present in the corresponding vector. We use the mean cosine similarity score as the overall score for the search string. This similarity measure helps us to explore the more relevant search paths first.

Expansion of the Search Space: Exploration of the video search space uses a greedy approach. Initially each token is treated as a one-word search string. We insert them in a priority queue. In each iteration, we pick the top scoring search string, concatenate other one word tokens with the search string, perform the search with the YouTube API, compute the mean similarity score from the search results, and push the new search string back to the priority queue. We make sure that the token sequences for which we are searching are unique and that no tokens are repeated in a search string as we expand the search. The process terminates when we either run out of all possible combinations of search tokens or explore a user-specified number of search strings. Throughout the process, we keep track of the search strings that yields the best set of videos that the user is seeking. An ordered list of search strings and corresponding videos are returned to the user.

4 Experimental Setup

Our approach allows us to consider the merit of including various types of extracted metadata in the search process. We designed an experiment to evaluate which configurations provided the most relevant search results and how these results compared to those returned from YouTube. We first searched YouTube using the search string "*autism personal stories*." We used the browsers' incognito mode feature and disabled the location sharing feature so that YouTube does not provide any personalized search result. Based on manual inspection of the top 10 search result videos, we observed that each of them is relevant to the search string. We later used these 10 videos for the validation of the proposed solution. As the baseline condition, for each of these 10 videos, we used YouTube API to obtain a list of 10 other recommended videos. We maintained a system-wide circular queue of 10 YouTube API keys. Each YouTube API call picks an API key from this circular queue and move the pointer to the next. We believe that such system reduced a potential confounder created by the fact that YouTube may store cookies or other kind of tracking information to personalize the search result. To show the efficacy of the proposed vExplorer system we designed our experiment to

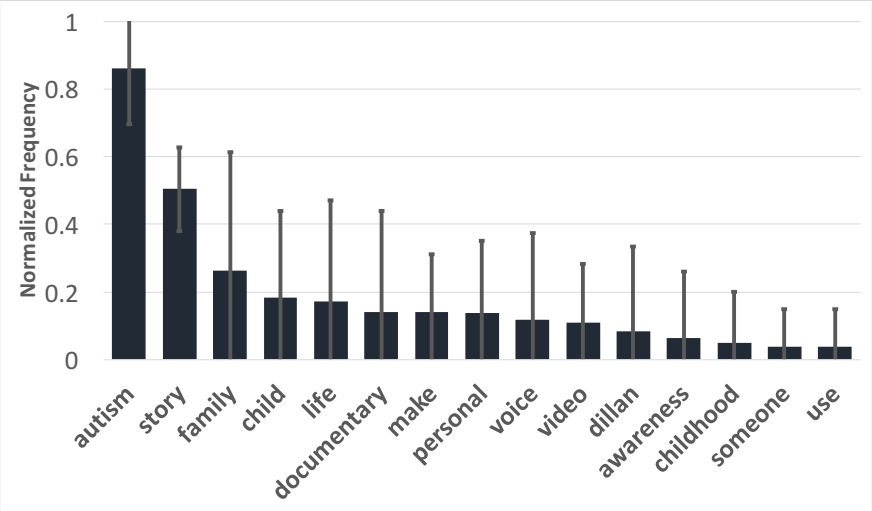

Figure 2: A set of 15 tokens aggregated from 10 seed videos considering the title and description. Y-axis shows the mean normalized TF ($\pm$ SD).

prove the following hypothesis.

H01. vExplorer suggested videos are more relevant to a seed video than YouTube recommended videos.

From the previous list of 10 selected YouTube videos, we picked one video at a time and set as the seed video. We generated 15 most relevant tokens based on normalized term frequency in three experimental settings. First, we only used title and description as a baseline condition for vExplorer. Second, in addition to title and description, we also included tags which are provided by the video uploader. Third, we also included user comments and their replies. 15 tokens can generate $2^{15} - 1$ search strings which may take significant time to complete. Given that we used a greedy approach to generate search strings, we believe that the vExplorer system can reach the desired search string fast. We limited our system to run only for 100 iterations (see figure 5). We obtained 10 vExplorer suggested relevant videos in each of the three settings. Three experimental settings, 10 seed videos, and each suggesting 10 related videos produces 300 videos in total. In addition, there are 10 YouTube suggested related videos for each of the 10 seed videos making a total of 100 videos. Overall, we created a set of 400 related videos that were suggested by YouTube and vExplorer.

We took a union of YouTube recommended videos and our system recommended ones (n=400) for all three experimental settings. Due to duplicate videos appearing across different experimental settings, we had only 200 unique YouTube videos in this set. Before having human raters assess the relevance of the returned videos, we randomized the ordering of the videos to remove potential bias. Two independent raters, not involved in the development of vExplorer, were recruited to rate the relevance of autism videos, and were given a list of the 10 initial seed videos. They were asked to watch these videos, and they were told of the initial search string, "*autism personal stories*." After the participants gained confidence in understanding the health-related topic presented in those initial seed videos, they proceeded independently to rate each video in the randomized list into three categories, "*relevant*," "*not relevant*," and "*unsure*." In cases of disagreement between rated categories, a third reviewer, not involved in the development of the system, was brought into adjudicate the discrepancy.

5 Results

We present the results of our evaluation where the selected 10 seed videos had each produced 15 tokens based on the normalized token frequency. In the first experiment setting, we use the title and description of videos to find the 15 most informative tokens. The title and description are also used to compute the similarity score between a seed video and any given video. Figure 3 shows the mean ($\pm$ SD) normalized score for each of the set of 15 selected tokens from 10 seed videos. We observe that autism, story, family, child, life, and documentary are the most informative tokens. The system then tries different combinations of these tokens to form candidate search strings. Table 1 shows a list of 5

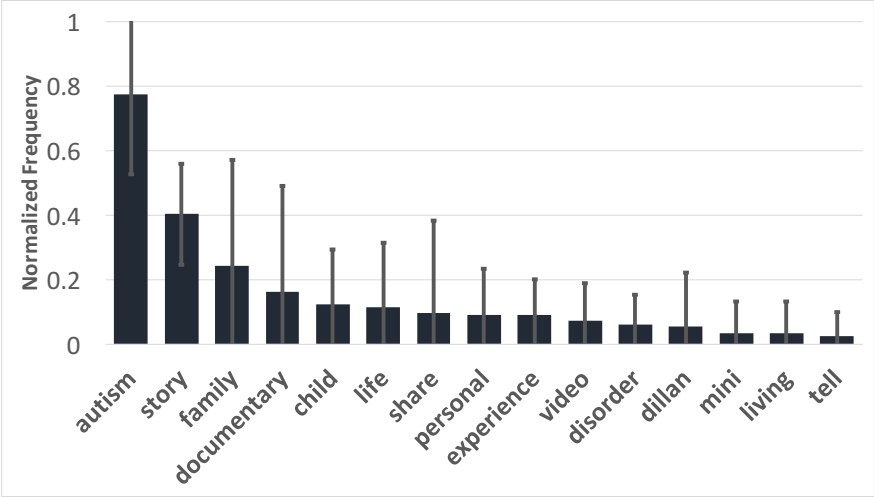

Figure 3: A set of 15 tokens aggregated from 10 seed videos considering the title, description, and tag. Y-axis shows the mean normalized TF (± SD).

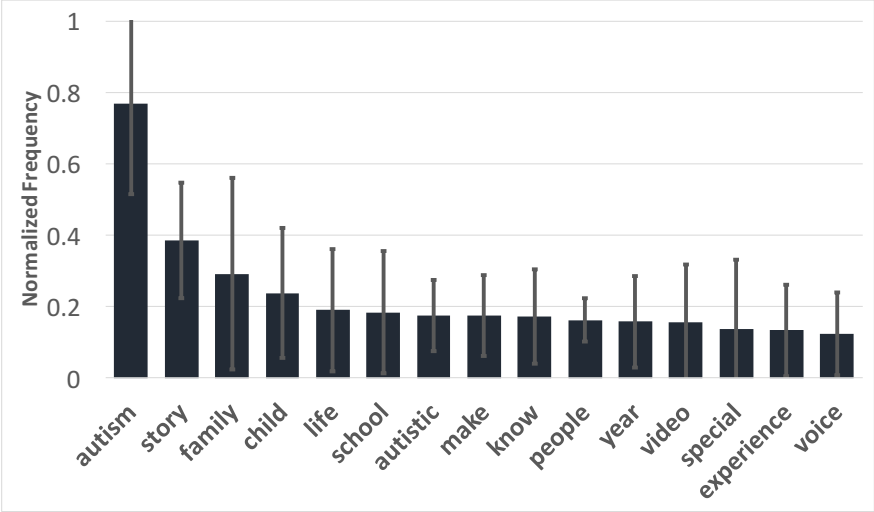

Figure 4: A set of 15 tokens aggregated from 10 seed videos considering the title, description, tag, and comment. Y-axis shows the mean normalized TF (± SD).

example search strings suggested by vExplorer. Although "*dillan*" and "*voice*" independently, are not among the most common tokens, we observe that "*dillan voice*" is the highest scored search string based on the content it produces when searched on YouTube. Figure 3 list tokens generated by including the tags into consideration while Figure 4 considers all four types of metadata.

Figure 5 provides a comparison of the relevance of YouTube recommended videos against the three configurations of our system generated recommendations. 60% of the cases YouTube provides relevant videos while 20% cases they are completely non-relevant. Our baseline configuration, which uses only title and description of the videos, provided results that are relevant 46% of times, much lower than YouTube recommendation (60%). However, when we take tags into consideration, vExplorer outperforms YouTube recommendation in relevance (70% versus 60%). The remaining 30% videos were categorized as "unsure," meaning our system never suggested any non-relevant video. Surprisingly, including comments into the configuration did not add any value. Video results are still rated 70% relevant and 30% unsure, and no videos are unrelated.

Table 1: Top 5 queries generated by vExplorer in 3 different experiment settings

Title+Description	Title+Description+Tag	Title+Description+Tag+Comments
dillan voice	autism documentary mini living	autism family life
autism story personal	autism documentary mini life	autism family special
dillan voice autism	autism documentary mini living life	autism family story
autism story personal video someone	autism story personal tell	autism story
autism story personal video use	dillan voice	autism story make

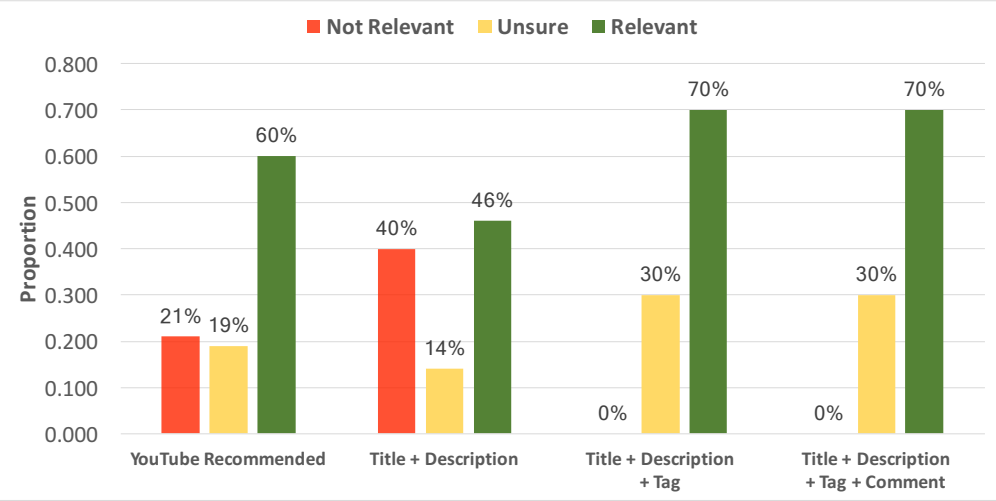

Figure 5: Proportion of videos "relevant," "not relevant," or "unsure" based on YouTube based recommendation or based on vExplorer. Considering title, description, and tag of the video proposed system outperforms YouTube based recommendation.

6 Conclusion and Future Work

In this paper, we propose a novel search system to help researchers discover videos of health-related behaviors based on an initial seed video of interest in YouTube. Our system helps researchers avoid common challenges in this type of search. For example, the user may not know which keywords provide optimal results in their manual search of related videos. In addition, certain tags on videos cannot be used directly as search terms in the public YouTube video page. To address these challenges, we have developed an unsupervised information retrieval approach that extracts and uses tags and other metadata to search and rank videos based on relevance to a seed video.

Our evaluation of different metadata configurations of vExlporer found that adding comments to the search strategy did not improve relevance of the results. This unexpected finding for social media data may be attributed to numerous factors. First, the presence of slang terms and inflammatory trolls in user comments may make it hard for the proposed system to capture the innate concepts in a specific video. Detection of trolling[23] may have the potential to improve the performance of our system. Second, our search space is limited to the token list available in the initial seed video. While computing the similarity score, our current system may overlook missing but relevant tokens in a seed video. This may be a likely scenario in the case where the seed video's title, description, tags, and comments are not sufficiently verbose. The incorporation of a thesaurus may improve the performance of the proposed system. We can also generate close caption text for each video and include the content in the bag of words representation. In addition, we may be able to improve the performance further by including speech and video frame-based features. Third, YouTube provides related videos based on analytics built into their native platform (e.g., database) whereas YouTube API adds the network latency. Our system is significantly slower in comparison to a YouTube search. Systems like vExplorer could be integrated into these online video services to provide a better user experience. Precomputed hierarchies of related videos may also reduce the search time significantly. Fourth, our evaluation was confined to the

concept of "*autism personal stories*" where only a limited set of YouTube videos are available. Many videos did not have any comments and for some the uploader of the video disabled user comments. In summary, we plan to expand our current work to address these challenges and to conduct more comprehensive evaluations of the system in different health domains.

References

1. David T Fetzer and O Clark West. The hipaa privacy rule and protected health information. *Academic radiology*, 15(3):390–395, 2008.

2. Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(2009):12, 2009.

3. Jure Leskovec and Julian J Mcauley. Learning to discover social circles in ego networks. In *Advances in neural information processing systems*, pages 539–547, 2012.

4. Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. ACM, 2015.

5. Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

6. Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 529–534. IEEE, 2011.

7. Trishul M Chilimbi, Yutaka Suzue, Johnson Apacible, and Karthik Kalyanaraman. Project adam: Building an efficient and scalable deep learning training system. In *OSDI*, volume 14, pages 571–582, 2014.

8. Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1041–1044. ACM, 2014.

9. Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Ismir*, volume 2, page 10, 2011.

10. Betul Erdogdu Sakar, M Erdem Isenkul, C Okan Sakar, Ahmet Sertbas, Fikret Gurgen, Sakir Delil, Hulya Apaydin, and Olcay Kursun. Collection and analysis of a parkinson speech dataset with multiple types of sound recordings. *IEEE Journal of Biomedical and Health Informatics*, 17(4):828–834, 2013.

11. Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

12. Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

13. Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *International workshop on ambient assisted living*, pages 216–223. Springer, 2012.

14. Timo Sztyler and Heiner Stuckenschmidt. On-body localization of wearable devices: An investigation of position-aware activity recognition. In *Pervasive Computing and Communications (PerCom), 2016 IEEE International Conference on*, pages 1–9. IEEE, 2016.

15. Xiao Wu, Alexander G Hauptmann, and Chong-Wah Ngo. Practical elimination of near-duplicates from web video search. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 218–227. ACM, 2007.

16. Douglass R Cutting, David R Karger, Jan O Pedersen, and John W Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 318–329. ACM, 1992.

17. Nachiketa Sahoo, Jamie Callan, Ramayya Krishnan, George Duncan, and Rema Padman. Incremental hierarchical clustering of text documents. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 357–366. ACM, 2006.

18. Daan Odijk, Edgar Meij, Isaac Sijaranamual, and Maarten de Rijke. Dynamic query modeling for related content finding. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 33–42. ACM, 2015.

19. Maria-Florina Balcan, Yingyu Liang, and Pramod Gupta. Robust hierarchical clustering. *The Journal of Machine Learning Research*, 15(1):3831–3871, 2014.

20. David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

21. Vincent A Fusaro, Jena Daniels, Marlena Duda, Todd F DeLuca, Olivia DAngelo, Jenna Tamburello, James Maniscalco, and Dennis P Wall. The potential of accelerating early detection of autism through content analysis of youtube videos. *PLOS one*, 9(4):e93533, 2014.

22. Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

23. Patxi Galán-García, José Gaviria de la Puerta, Carlos Laorden Gómez, Igor Santos, and Pablo García Bringas. Supervised machine learning for the detection of troll profiles in twitter social network: Application to a real case of cyberbullying. *Logic Journal of the IGPL*, 24(1):42–53, 2016.