# Video2Text: Learning to Annotate Video Content

Hrishikesh Aradhye
George Toderici
Jay Yagnik
*Google, Inc.*
*Mountain View, CA 95050, USA*
*hrishi@google.com*

***Abstract*—This paper discusses a new method for automatic discovery and organization of descriptive concepts (labels) within large real-world corpora of user-uploaded multimedia, such as YouTube.com. Conversely, it also provides validation of existing labels, if any. While training, our method does not assume any explicit manual annotation other than the weak labels already available in the form of video title, description, and tags. Prior work related to such auto-annotation assumed that a vocabulary of labels of interest (e.g., indoor, outdoor, city, landscape) is specified a priori. In contrast, the proposed method begins with an empty vocabulary. It analyzes audiovisual features of 25 million YouTube.com videos – nearly 150 years of video data – effectively searching for consistent correlation between these features and text metadata. It autonomously extends the label vocabulary as and when it discovers concepts it can reliably identify, eventually leading to a vocabulary with thousands of labels and growing. We believe that this work significantly extends the state of the art in multimedia data mining, discovery, and organization based on the technical merit of the proposed ideas as well as the enormous scale of the mining exercise in a very challenging, unconstrained, noisy domain.**

## I. Introduction

The traditional machine learning approach begins with a vocabulary of labels to be learned and searches for features and classifiers that can best distinguish these labels. This approach is inherently limited by the label set – which is often niche and hand-selected – because the classifiers by definition can only learn concepts within this vocabulary of labels. It does not scale well given the enormity and diversity of modern multimedia corpora. Popular online video archives such as YouTube contain hundreds of millions of videos with several hundreds of thousands of new uploads every day. The richness and diversity of content of these videos is reflected in the millions of words spanning several different languages in the title, tags, and description sections of these videos. In comparison, any hand-designed niche set of labels (e.g., porn, news, sports, music) falls woefully short in describing the content of these videos.

The approach described in this paper uses a subset of 25 million YouTube videos for training and validation. We extracted a generic set of large dimensional audiovisual features on each video in this corpus. We then formed a pool of concepts by taking a union of all unique words and N-grams from user-supplied metadata text over this corpus. The proposed approach then jointly selects *(a)* a subset of the available features and decision stumps that can best classify a given label and *(b)* a maximal set of labels that can possibly be learned to the desired level of accuracy using a given set of generic features. We developed a novel scheme to iteratively expand the vocabulary of learned concepts by using classifiers learned in prior iterations to learn composite, complex concepts. The semantic descriptive capacity of our approach is thus bounded not by the intuition of a single designer but by the collective, multilingual vocabulary of hundreds of millions of web users. All aspects of the proposed algorithms are parallelizable over a number of CPUs in order to make a machine learning problem of this scale solvable in a reasonable timeframe.

The utility of the proposed approach from a data mining perspective is multifold, although experimental demonstration of some of these applications is beyond the scope of this paper. First, our method provides a parallelizable mechanism to generate multiple content-aware semantic descriptors for a given video. User-uploaded metadata is often incomplete. Our algorithm augments and reinforces it, underscoring tags that are audio-visually meaningful and consistent. Secondly, this mechanism can be perceived as a feature transformation where the native feature space is transformed into the space of classifier scores, which in turn could be used as features for subsequent learning purposes such as video search ranking, determining related videos, and niche genre classification tasks. We demonstrate that for an example supervised learning task of softcore porn detection, the proposed method can achieve slightly better performance than the baseline while requiring only 10% of annotated training data. We also demonstrate that, given a fixed minimum performance criteria, our iterative training scheme can learn up to 92% more concepts that meet or exceed these criteria than the baseline.

## II. Prior Work

**Label Discovery:** In the context of multimedia data mining, label discovery corresponds to unsupervised association of videos/images/audio files and text labels. A set of approaches have studied unsupervised auto-annotation of

images [1], [2], [3], [4] with generic objects of interest. These approaches train over images with weak labels. The images may contain multiple objects and the exact locations of the objects are not known. Each training image is first segmented into a collection of regions. An unsupervised learning algorithm is then applied to estimate the joint probability distribution of labels and visual features (usually local features). Given the same visual features over an un-annotated image, posterior probabilities of labels are estimated. Similar to these approaches, the proposed work also requires weak labels for training purposes, which in our case are the words and N-grams in the video title, tags, and description.

One primary difference is that the space of possible labels (interchangeably referred to as the label vocabulary) in our case is practically infinite and certainly not known a priori. The proposed approach begins with an empty vocabulary and automatically builds it by discovering a set of "learnable" labels given a native feature space. It continues to iteratively discover more composite concepts by using some the simpler concepts in the vocabulary. In contrast, the existing label discovery approaches require a previously known and tractable vocabulary of labels. Secondly, the choice, descriptiveness, and completeness of metadata text in our domain of interest are left to the uploader's discretion. A video labeled "Britney Spears" may be, for example, a performance by the pop star, a video of somebody else talking about her, a slideshow of unrelated photos set to one of her songs, or it may have nothing to do with Britney Spears at all. Conversely, there may be videos related to Britney Spears that are not so labeled. There are videos with spelling errors as well as videos with the tags in a multitude of languages. To the best of our knowledge, the existing auto-annotation approaches assume carefully and exhaustively labeled data. It is unclear how they will fare in a domain with unconstrained content and incomplete/inaccurate/ill-defined/misleading ground truth.

**Learning in the Wild:** A related problem of learning faces in the wild has received some attention in the recent past. Cour *et. al.* [5] match person name tags to faces present in the image achieving disambiguation in case of multiple faces/names. Yagnik *et. al.* [6], [7] proposed a *consistency learning* framework for learning celebrity facial signatures from images in the wild given a named entity extractor and a list of celebrity names of interest. Sargin *et. al.* [8] proposed a similar framework using voice as the biometric signature for celebrity speaker identification. Although these approaches accept unconstrained multimedia input, the label vocabulary is still bounded. Moreover, all of these approaches depend on carefully selected biometric features that are known to correlate well with the label vocabulary of interest (person names). In contrast, our approach works with any native feature space and automatically learns a set of labels that relate to the native feature space. In that sense, our method is a generalization of these learning in the wild approaches.

**Unsupervised Data Augmentation:** These approaches aim to augment small manually labeled datasets by unsupervised or semi-supervised mechanisms. The work by [9], [10], [11] issues queries to popular image search engines to build models for a category. These models are subsequently used to augment the dataset by finding more images of that category or for improving search results. The method proposed by [12] starts with a small manually labeled dataset on a known set of concepts and trains separate classifiers on text and image content. It then employs a heuristic mix of the two to determine when to label unseen images. This scheme is used iteratively to build a set of newly discovered images for each concept.

**Supervised Concept Learning:** A large number of supervised video concept learning approaches have been published using TREC video data (e.g., [13]). While related to some aspects of the proposed work, these approaches differ from the current objective of unsupervised learning without a bounded, pre-set vocabulary of concepts. Some of the lessons learned, however, are applicable. Qi *et. al.* [14] developed a system for labeling videos, which makes use of the fact that many of the labels are correlated. It constructs a correlation graph of the various concept classifiers, which is then used to make a final decision. Montagnuolo *et. al.* [15] proposes a method for categorizing television footage into a eight possible video genres using a parallel neural network.

Yang *et. al.* [16] developed a video genre classification system for YouTube videos using user-supplied video category (e.g., sports, music, etc.) as the ground truth. They used a corpus of 11K YouTube videos and demonstrated that by mixing the classifier outputs from different modalities a more accurate outcome is achieved. Although the their domain of application is the same as this paper (i.e., YouTube videos), their problem space is significantly different for the following reasons. (a) The video category – the ground truth used in [16] – is selected by the user from a fixed set of allowed labels, spanning only 17 categories. Our assumed ground truth, on the other hand, is derived from words in the title, tags, and description which are bounded only by the vocabulary of the user. On one hand, our training problem is thus significantly more difficult with a much larger decision space coupled with much noisier ground truth. One the other hand, the descriptive power of the annotations by our system is far higher as well with output vocabulary size in the thousands of labels. (b) The classification problem for [16] is unilabel, meaning only one label from the 17 categories is associated with one sample, whereas ours is multilabel where any number of tags could be associated with one sample. (c) Given the small target decision space, [16] could afford to manually choose a specific multimodal feature space that works well for the desired classification tasks. In contrast, the proposed method can work with any feature space and will accordingly synthesize a vocabulary

of concepts best suited for that feature space. As an extreme case, we demonstrate that our method can indeed work a single dimensional feature space.

To the best of our knowledge, no other published effort has attempted to solve the problem of interest to this work – large scale unsupervised auto-annotation of multimedia data in the presence of noisy and incomplete ground truth and unspecified, practically unbounded vocabulary.

## III. Approach

### A. Audiovisual Feature Extraction

Our approach is not dependent on any specific underlying feature space, referred to as the *native* feature space in the discussion below. Given a specific feature extraction algorithm, concepts that can be described with this feature space are automatically discovered. A variety of algorithms for extracting video-level features have been described in the literature. For the purposes of this paper, we have focused on one example feature characterizing the motion in the video, one visual feature category characterizing color/texture, and an audio feature category. Our framework can process audiovisual modalities separately or jointly, and multiple feature categories within the same modality are allowed. However, demonstrating results with exhaustive feature engineering is not the purpose of this paper. We now provide a brief description of each of these three feature extraction algorithms. Note that these or similar algorithms have been or will be published elsewhere and are not a part of the contribution of this work.

**Motion Rigidity Feature:** This feature was specifically designed for detection of slideshow videos which are typically made of a series of static photo frames with transition effects of translation, pan, zoom or any combination of these. In other words, there is little non-rigid motion in the video. We use homography transformation error between feature points on two consecutive frames to estimate rigidity of motion. This feature generates one real number per video corresponding to the fraction of frames in the video deemed to have only rigid motion. Slideshow videos tend to have values closer to one and non-slideshow videos closer to zero.

**CONGAS-HS Features:** We use a custom local descriptor [17], [18] that collects Gabor wavelet responses at different orientations, spatial scales, and spatial offsets from the interest point. Four orientations and twenty-seven (scale, offset) combinations are used. We extract these descriptors at sparse interest points determined by a Laplacian of Gaussian feature extractor. We then compute bag-of-words histograms. We build codebooks using hierarchical k-means proposed by Nister [19]. The sparse CONGAS feature histogram is extracted every half second. Once the entire video is processed, the cumulative histogram is thresholded such that each bin will have at least ten activations, corresponding to a feature appearing in the video for at least five seconds. In addition, we compute an 8x8 Hue-Saturation histogram for each frame. Each bin is then interpreted as a time series. We use a 8-level 1D Haar wavelet decomposition, and compute the first two moments and the extrema at each level. This descriptor is then appended to the CONGAS feature to form our final feature vector, which is a sparse vector of real-valued data that implicitly characterizes object shape, color variation over time, and texture within a video.

**SAI Features:** The auditory features that we use are based on models of the mammalian auditory system. Specifically, we use a cochlear-model of filter banks that form a *stabilized auditory image* (SAI) [20]. Computing the SAI starts with a set of band-pass filters, followed by an autocorrelation of each channel, which is then transformed into a sparse code using vector quantization. The end result is a sparse vector of real-valued data that implicitly characterizes several aspects of music and speech of the audio track. For a detailed description of the features, please refer to the work by Rehn *et. al.* [21] which uses these features for ranking and retrieval of sound files.

### B. Training Procedure

The following procedure describes our algorithm for training new models. This procedure is classification algorithm-agnostic. We have chosen AdaBoost for our experiments because of its simplicity, speed, and ability to select features as needed.

*1) Data Preparation:* Our operating set is roughly 25 million videos from YouTube. This set contains a wide variety of contents, formats, languages and is arguably large enough to be considered representative. The demographic that uploads, annotates, or views these videos is very diverse in terms of age, nationality, gender, interests, and so on. The list below outlines the steps for organizing this data for concept learning:

1) Loop over text metadata to collect a pool of unique words and N-grams, excluding stopwords, with N set to 2 for the purpose of this paper. Members of this pool now referred to as *concepts*.
2) Purge concepts that are too infrequent or too frequent.
3) Extract an audio/video/audiovisual feature vector for each video in the corpus. Features could be dense or sparse or mixed.
4) Split the video corpus into train and validation partitions. Split the validation set further into a number of sub-partitions equal to the maximum number of concept learning iterations.
5) For each concept in the pool, form one train and a number of validation datasets: *(a)* Define positive examples as videos that include this concept in the metadata. *(b)* Form negative examples by removing the positive set from a generically large subset of the corpus. This amounts to one-vs-all classification. For a given concept, we selected three times as many

negative samples as the number of its positive samples. *(c)* Randomly subsample train and validation sets to keep the training tractable.

*2) Training and Concept Vocabulary Synthesis:* The training procedure is iterative. In each iteration, we add new one-vs-all classifier models to a set of retained models. Each classifier essentially learns to separate videos that should be tagged with a certain concept (positive examples) from videos that shouldn't be (negative examples). Note that positive and negative training examples are defined on the basis of words in user-supplied metadata, as manual annotation or cleanup is not a feasible option given the scale of the problem and the richness of the vocabulary. Words coming from the text metadata are weak labels at best, as uploaders are free to annotate their videos with any text snippet they choose. However, the extent of ground truth noise varies for each of these one-vs-all classifiers. One factor is the specificity of the label in the context of online videos. For example, the label "video" is generic to the extreme: its presence or absence in the metadata conveys no meaning, as they are all videos. The label "music video", on the other hand, is more specific and its presence or absence is thus less noisy. Another factor is the "learnability" of a label given the native feature category. For instance, given visual-only features, a classifier cannot reliably learn certain labels related to music. By *(a)* discarding labels that are too frequent or too infrequent, *(b)* evaluating on an yet-unseen validation set after every training iteration, and *(c)* requiring that both precision and recall be high for a classifier to be retained, we mitigate the effect of both of these factors. In other words, we jointly select classifiers for those labels that are both semantically specific to a certain degree as well as learnable given the native feature space:

1) Initialize a set of retained classifier models to empty.
2) Train a model for each concept in the pool: *(a)* Train a binary one-vs-all classifier using the train set. *(b)* Evaluate using a partition of the validation set, choosing a different partition for every iteration of steps 2 through 4 to ensure data purity. *(c)* Add model to the set of retained models only if both precision and recall surpass desired thresholds.
3) Update feature data: *(a)* Apply all newly retained models to each feature vector in the corpus. *(b)* Append resulting classifier scores to the feature vector.
4) Iterate over steps 2 and 3 until the set of retained models stops growing, or until the maximum number of iterations is reached.

The only adjustable parameters in this algorithm are the model retainment performance threshold and the number of stumps for AdaBoost classifiers. We nominally set the minimum required precision and recall both to 0.7. Models performing below this threshold are not retained. We used one constant number of stumps for all AdaBoost classifiers in the system. The effect of the choice of number of stumps will be described in the results section.

| Index | Concept |
|---|---|
| 1 | slideshow |
| 2 | pics |
| 3 | sakura |
| 4 | sasuke |
| 5 | runescape |
| 6 | video camcorder |
| 7 | flip video |

Table I
CONCEPT VOCABULARY: MOTIONRIGIDITY

## IV. RESULTS

### A. Vocabulary Synthesis

**Motion Rigidity Feature Space:** As an experiment to validate our vocabulary synthesis process, we applied our algorithm on the motion rigidity feature on a subset of our corpus with 3M videos. Since the feature vector here is unidimensional (i.e., one real number per video), we chose number of stumps to be as small as 2 to avoid over-fitting with only one vocabulary synthesis iteration. Given that this feature was previously used for slideshow detection, we anticipated that our algorithm would discover a vocabulary of labels related to slideshows.

The vocabulary of concepts generated in this exercise is provided in Table I. The concepts `slideshow` and `pics` are clearly consistent with what the native feature space was designed to do. `sakura` and `sasuke` are popular animé characters. *Animé* is a Japanese animation style that relies heavily on pan and zoom effects on a static painting to give the illusion of movement. The end effect is much like slideshows. `runescape` is a video game with somewhat limited movement over static or rigidly moving frames. The remaining two concepts, `video camcorder` and `flip video` are a bit harder to explain. A quick YouTube search with these keywords comes up with close up views of a camcorder with a voice-over discussing its features. There is indeed non-rigid motion but not too much of it. It is likely that, with two stumps, the classifier was able to carve out a niche range of the motion rigidity feature where a majority of these videos belong. This example thus illustrates that the vocabulary of concepts synthesized by our approach adapts to the strengths of the native feature space and at the same time is not bounded by the imagination of the designer of the feature.

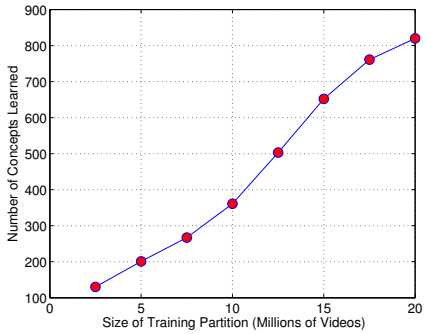
**CONGAS-HS and SAI Feature Spaces:** Tables II and III list a subset of the vocabulary automatically synthesized for CONGAS-HS and SAI features, respectively. For the sake of better readability, the chosen concepts were manually organized into 10 groups each. While these subsets don't provide the complete picture, it gives the reader an idea of the diversity of concepts discovered. Applying our algorithm

| | |
|---|---|
| 1 | amv, gohan, goku, naruto shippuden, manga |
| 2 | bike, bmw, bmx, exhaust, honda, motorcycle, racing |
| 3 | bollywood, hindi, awards |
| 4 | arena, futbol, goals, soccer, match, wwe, skateboarding, skating, skiing, snowboard, snowboarding, highlights |
| 5 | english subtitles, subtitles, subtitulado |
| 6 | bikini, blonde, nude, lesbian, fetish |
| 7 | halo, ps3, xbox, nintendo, kingdom hearts, final fantasy |
| 8 | telenovela, television, general hospital, drama |
| 9 | commercial, documentary, tribute, teaser, interview, music video, cartoon, horror, indie, nature |
| 10 | pics, pictures, pivot, slideshow, montage |

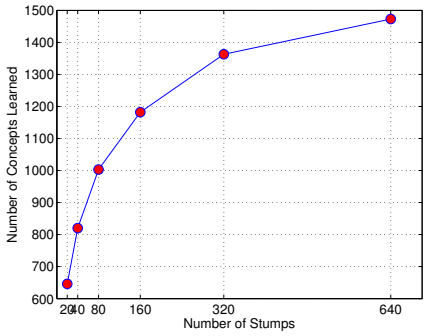
Table II
EXAMPLE CONCEPTS: CONGAS-HS

| | |
|---|---|
| 1 | acceleration, exhaust sound, hockey, skateboarding, game review, amazing highlights |
| 2 | accordion, acoustic guitar, bass, drum, drum solo, drumline, instrumental, jazz guitar, jazz piano, piano solo |
| 3 | acappella, acoustic, classical, chipmunks, club remix, dance techno, death metal, gangsta rap, r&b, radio edit, reggae, soundtrack, uplifting, electronic trance |
| 4 | artists bands, singing, lyrics screen, recitation |
| 5 | adult japanese, amateur college, foot worship, girl erotic |
| 6 | aljazeera, bbc news, standup comedy, commercial, documentary part, drama entertainment, home shopping |
| 7 | islam, quran |
| 8 | films indian, hindi song, entertainment bollywood |
| 9 | audio latino |
| 10 | manga, amv, runescape, latino anime |

Table III
EXAMPLE CONCEPTS: SAI




(a) Effect of Training Corpus Size (40 stumps)




(b) Effect of Number of Stumps ($20M$ training videos)

Figure 1. Factors Affecting Vocabulary Size

on CONGAS-HS features on a train partition of 10M videos with 800 stumps, 348 concepts were discovered in the first iteration. Analogously, using SAI features on a training partition of 20M videos with 640 stumps, 1473 concepts were discovered in the first iteration.

We observed that increasing the corpus size linearly resulted in faster-than-linear rate of increase in the number of concepts learned in the first iteration. (Fig. 1(a)). Conversely, increasing the number of stumps for each AdaBoost classifier brings diminishing returns in terms of new concepts learned (Fig. 1(b)).

### B. Applications

**Label Verification:** The objective of label verification is to verify that a label from user supplied metadata matches with the actual content of the video and disregard the label if it doesn't. This helps improve the precision of video retrieval and helps eliminate spam. The results in Figures 4 and 5 show 6 sample concepts automatically learned using CONGAS-HS features: `cartoon`, `hindi`, `telenovela`, `skateboarding`, `nature`, and `subtitulado`. Each column corresponds to a separate concept with the labels provided at the bottom of the figure. The 5 rows of Figure 4 correspond to the top 5 videos in our corpus – ranked solely on to the concept classifier score – where the user included the concept in the text metadata. These are thus cases where the user-supplied text metadata was *confirmed* by our concept classifier. Conversely, the 5 rows of Figure 5 correspond to the *bottom* 5 videos in our corpus – again ranked solely on the concept classifier score – where the user included the concept in the text metadata. The user-supplied text metadata was thus *rejected* by the concept classifier in these cases.

The examples in Figure 4 can be seen to be by-and-large relevant to the corresponding concept. Given the nature of the native feature space (CONGAS-HS), it is not surprising that it could form reliable models for `cartoon` or `nature`. The model for `skateboarding` possibly learned both the skateboard as an object and the skateboarding as an action. The models for `hindi` and `telenovela` seem to have learned the typical color and motion palette of Bollywood song-and-dance sequences and soap operas, respectively. The model for `subtitulado` was the most interesting, because it seems to have learned not only the texture of overlaid subtitles, but the fact that they are in Spanish, perhaps based on accents.

In contrast, the examples of rejected samples shown in Figure 5 are indeed less representative of the concepts, even though the concept was listed by the user in the metadata. For instance, the rejected samples for `hindi` do include hindi-speaking subjects, but are not representative of the tag

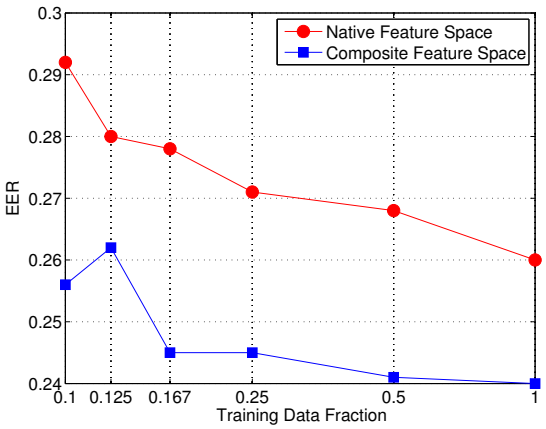



Figure 2. Effect on Supervised Classification

hindi given that they don't have Bollywood themes (See corresponding column from Figure 4). Similarly, many of the rejected skateboarding videos discuss skateboarding without showing the act of skateboarding. The performance of the subtitulado is a bit mixed as it does correctly reject cases where subtitles are not present or not readable but is also seen to incorrectly reject videos with readable, Spanish subtitles but in an unusual font. The rejected examples for cartoon and nature are less contentious and are by large-and-large correct decisions.

## V. Evaluation

### A. Effect on Supervised Classification

The proposed concept learning scheme can be interpreted as a feature transformation where a large dimensional native feature space is transformed into or is appended by a number of classifier scores. Figure 2 shows the effect of such feature transformation on a supervised genre classification problem. We trained AdaBoost models with a fixed number of stumps (40) for separating racy (i.e., not appropriate for minors due to suggestive sexual content) videos from safe videos. The models were trained and evaluated using a manually annotated dataset of nearly $20,000$ videos (of which half were not racy). We compared models trained on the native feature space (CONGAS-HS) against models trained on a composite space (CONGAS-HS + our iteration-1 concept classifier scores) using Equal Error Rate (EER). Note that the concept classifiers were trained using the same number of stumps (40) and the concepts racy or softcore did not become a part of the discovered vocabulary. Several related concepts were in fact discovered and consequently the composite feature space outperforms the native space. We experimented with using only a fraction of the available training data. The difference in EER is even more striking as the amount of annotated training data reduces. Note that the composite feature space achieves better EER with 10% of the training data than the native feature space with 100% of the training data. Manual annotation of thousands of videos is time consuming and expensive. By using knowledge gained by unsupervised learning, this work significantly reduces the dependence on supervised training data.

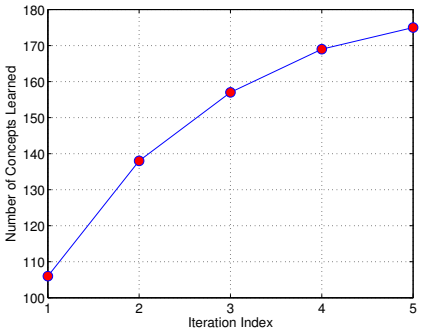
### B. Iterative Multilabel Classification

In this paper we proposed a new method for iterative multilabel classification. For each iteration, classifier scores of the models learned previous iteration are used as input features (referred to as transformed features), in addition to the native feature space (CONGAS-HS or SAI). On the other hand, the baseline approach for multilabel classification is implementing a number of one-vs-all classifiers, which is equivalent of our Iteration 1. Figure 3 (a) shows that this approach with CONGAS-HS features and 40 stumps resulted in 106 labels that met our evaluation threshold of $0.7$ minimum precision and recall. Our iterative scheme improves this number to 175 in five iterations, which is a 65% gain in labels that can be classified with the same criteria of minimum precision and recall. Analogously, Figure 3(b) demonstrates a similar increase using SAI features: from 77 labels in iteration 1 to 148 labels in iteration 5. This is amounts to a 92% gain in the number of labels relative to the baseline. Note that the training data, training parameters, model retainment criteria, and the global pool of candidate concepts remain unchanged from the baseline. The fact that new words are learned with each subsequent iteration implies that some of the transformed features were in fact selected by AdaBoost, replacing features from the native space, which resulted in these models exceeding the model retainment criteria. This attests to the utility of the feature transformation. The number of new words learned diminishes with each iteration, as easier concepts are all learned and the limits of the native feature space are reached.

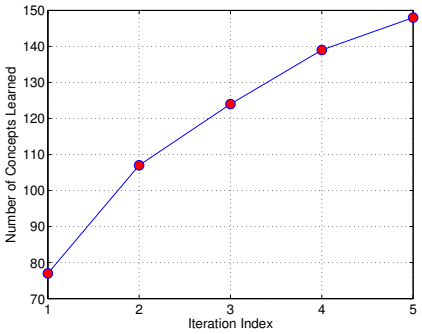
In one experiment, we allowed the models learned in prior iterations to be re-learned in subsequent iterations with the transformed feature space. The F-measure of the re-learned concepts generally increases from iteration to iteration (From $0.86$ in the first iteration to $0.88$ in the fifth for CONGAS-HS and correspondingly from $0.85$ to $0.87$ for SAI), implying that the classification for a given concept gets easier as decisions for related concepts are made available as input.

## VI. Conclusion and Ongoing Work

This paper presented a method for large scale auto-annotation videos without requiring any explicit manual annotation. The domain of application of the proposed method – content analysis of online short form video – is very challenging because of the sheer volume of data, unconstrained nature and diversity of the content, and noisy, multilingual text metadata. One of the novel contributions of this method is its ability to organically grow a vocabulary of concepts that best suits the native feature space




(a) CONGAS-HS




(b) SAI

Figure 3. Iteration vs Vocabulary Size

without requiring any manual involvement. The other is an iterative training scheme that greedily learns a series of high confidence one-vs-all classifiers while iteratively using the posteriors as inputs to build better classifiers for all classes. Our method can be viewed as an attempt to simultaneously solve both the latent layer representation and multilabel classification problem in the case of a large number of classes. We present several illustrative results and use cases of the proposed method. For example, we demonstrate that a composite feature space using our classifier scores achieves better EER with 10% of the training data than the native feature space with 100% of the training data for a supervised learning problem. Manual annotation of thousands of videos is time consuming and expensive. By using knowledge gained by unsupervised learning, this work significantly reduces the dependence on supervised training data. We also demonstrate that, given a fixed minimum performance criteria, our iterative multilabel learning scheme can learn up to 92% more concepts meeting or exceeding these criteria than the baseline.

To the best of our knowledge, this effort is one of the largest data mining exercises on video data. We believe that novelty of the proposed approach and the sheer scale of the analyses significantly enhances the state of the art in multimedia data mining. Our ongoing effort has focused on multimodal feature analysis and learning large scale multimedia semantic networks.

## REFERENCES


[1] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. I. Jordan, "Matching words and pictures," *J. Machine Learning Research*, vol. 3, pp. 1107–1135, 2003.

[2] D. Blei and M. I. Jordan, "Modeling annotated data," in *Proc. ACM SIGIR*, 2003, pp. 127–134.

[3] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary," in *Proc. ECCV*, 2002, pp. 97–112.

[4] S.L.Feng, R. Manmatha, and V. Lavrenko, "Multiple bernoulli relevance models for image and video annotation," in *Proc. CVPR*, 2004.

[5] T. Cour, B. Sapp, C. Jordan, and B. Taskar, "Learning from ambiguously labeled images," U. Penn., Tech. Rep., 2009.

[6] M. Zhao, J. Yagnik, H. Adam, and D. Bau, "Large scale learning and recognition of faces in web videos," in *Proc. FGR*, 2008.

[7] J. Yagnik and A. Islam, "Learning people annotation from the web via consistency learning," in *Proc. ACM SIGMM Workshop on MIR*, 2007, pp. 285–290.

[8] M. Sargin *et al.*, "Audiovisual celebrity recognition in unconstrained web videos," in *Proc. ICASSP*, 2009.

[9] L. Li, G. Wang, and L. Fei-Fei, "Optimol: automatic online picture collection via incremental model learning," in *Proc. CVPR*, 2007.

[10] K. Yanai and K. Barnard, "Probabilistic web image gathering," in *Proc. ACM SIGMM Workshop on MIR*, 2005, pp. 57–64.

[11] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning object categories from google's image search," in *Proc. ICCV*, vol. 2, 2005, pp. 1816–1823.

[12] H. Feng, R. Shi, and T. Chua, "A bootstrapping framework for annotating and retrieving www images," in *Proc. ACM Multimedia*, 2004.

[13] C. Snoek *et al.*, "The MediaMill TRECVID 2008 semantic video search engine," 2009.

[14] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, M. Wang, and H.-J. Zhang, "Correlative multilabel video annotation with temporal kernels," *ACM T. on Multimedia Computing, Communications, and Applications*, vol. 5, pp. 1–27, 2008.

[15] M. Montagnuolo and A. Messina, "Parallel neural networks for multimodal video genre classification," *Multimedia Tools and Applications*, vol. 41, pp. 125–159, January 2009.

[16] L. Yang, J. Liu, X. Yang, and X.-S. Hua, "Multi-modality web video categorization," in *Proc. ACM SIGMM Workshop on MIR*. New York: ACM Press, 2007, p. 265.

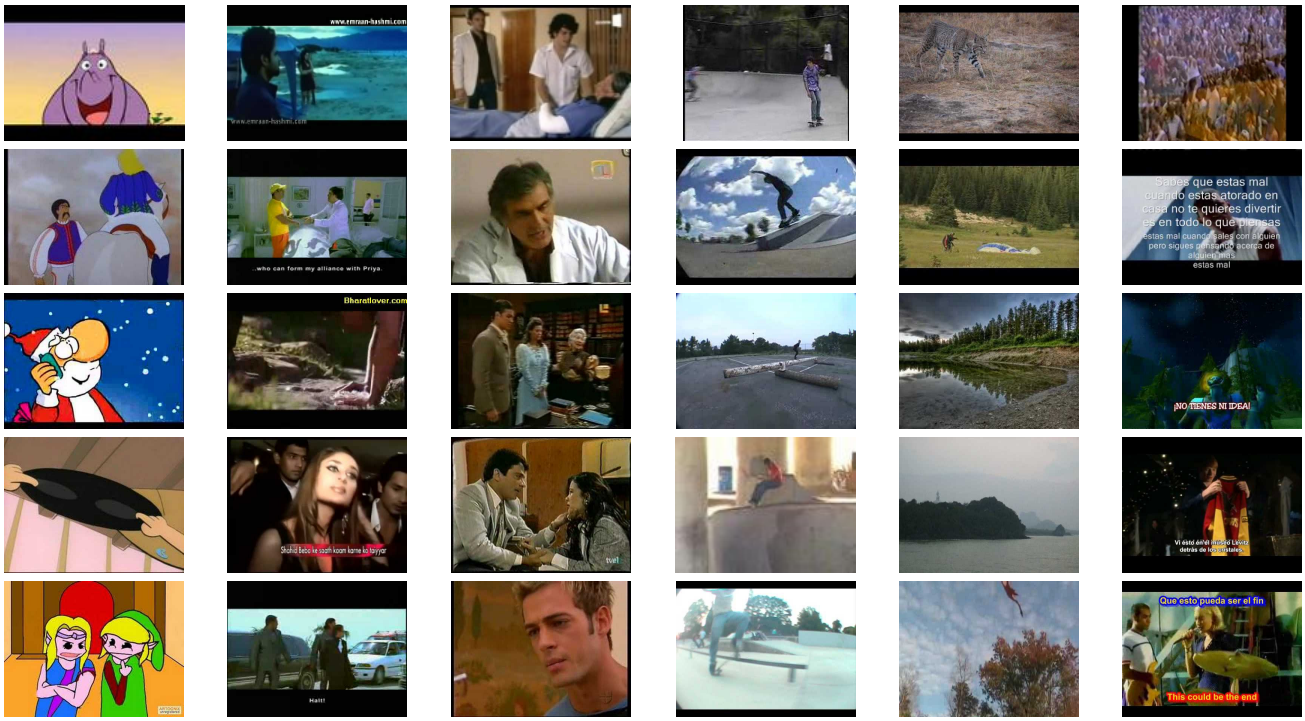
[17] E. Tola, V. Lepetit, and P. Fua, "A fast local descriptor for dense matching," in *Proc. CVPR*, 2008.

[18] H. Neven *et al.*, "Image recognition with an adiabatic quantum computer I. Mapping to quadratic unconstrained binary optimization," 2008. [Online]. Available: http://www.citebase.org/abstract?id=oai:arXiv.org:0804.4457

[19] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proc. CVPR*, vol. 2, Jun. 2006, pp. 2161–2168.

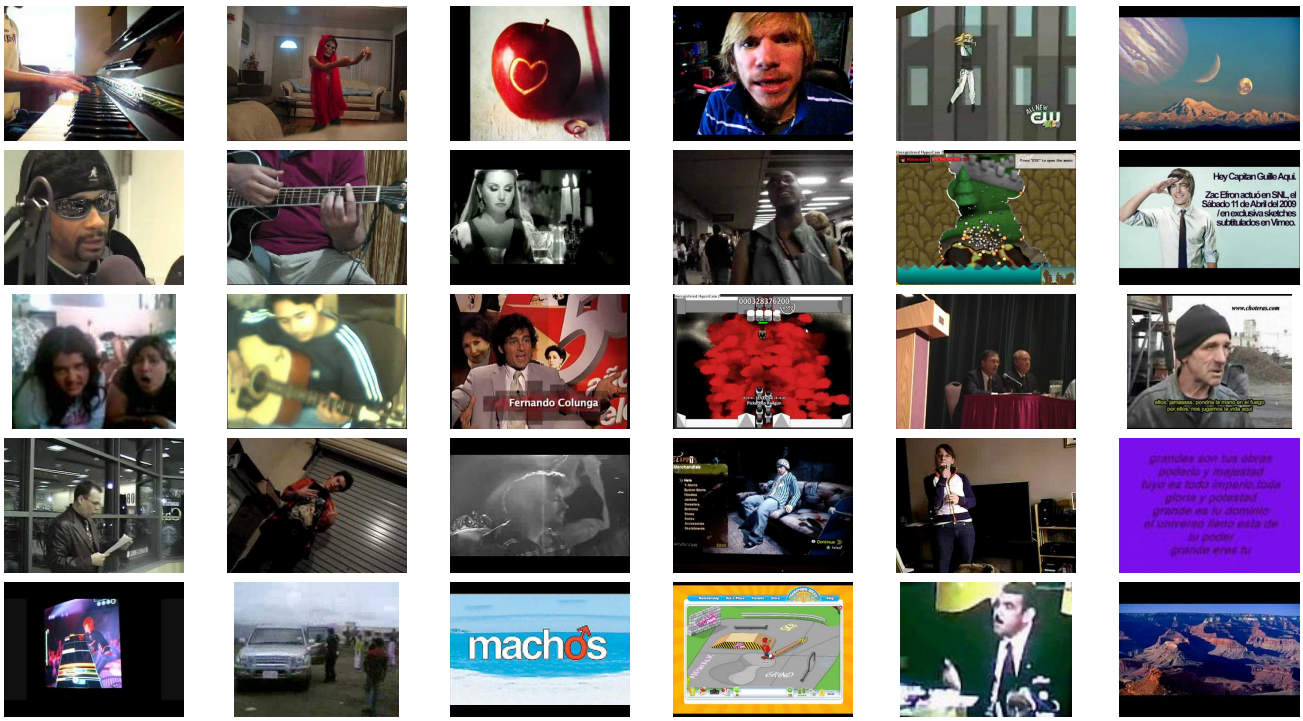
[20] R. D. Patterson *et al.*, "Complex sounds and auditory images," in *Proc. Auditory Physiology and Perception*, 1992, pp. 429–446.

[21] M. Rehn, R. Lyon, S. Bengio, T. Walters, and G. Chechik, "Sound ranking using auditory sparse-code representations," in *Proc. ICML: Workshop on Sparse Methods for Music Audio*, 2009.



(a) cartoon (b) hindi (c) telenovela (d) skateboarding (e) nature (f) subtitulado

Figure 4. Examples of Label Confirmation




(a) cartoon (b) hindi (c) telenovela (d) skateboarding (e) nature (f) subtitulado

Figure 5. Examples of Label Rejection