Improved Video Categorization from Text Metadata and User Comments

Katja Filippova
Google Inc.
Brandschenkestr. 110
Zurich, Switzerland
katjaf@google.com

Keith B. Hall
Google Inc.
Brandschenkestr. 110
Zurich, Switzerland
kbhall@google.com

ABSTRACT

We consider the task of assigning categories (e.g., *howto/cooking, sports/basketball, pet/dogs*) to YouTube[1] videos from video and text signals. We show that two complementary views on the data – from the video and text perspectives – complement each other and refine predictions. The contributions of the paper are threefold: (1) we show that a text-based classifier trained on imperfect predictions of the weakly supervised video content-based classifier is not redundant; (2) we demonstrate that a simple model which combines the predictions made by the two classifiers outperforms each of them taken independently; (3) we analyse such sources of text information as video title, description, user tags and viewers' comments and show that each of them provides valuable clues to the topic of the video.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Linguistic processing*

General Terms

Algorithms, Experimentation

Keywords

Web video, video categorization, text analysis

1. INTRODUCTION

The amount of video content on the web has increased drastically in the recent years. Faster Internet connections, ubiquitous use of filming devices and the popularity of video blogging, all contribute to the steady growth of video content on the Internet. This trend requires more accurate video categorization to make the information searchable and hence more useful for users. Machine learning methods have been used extensively for video, image and text classification as they can handle thousands of features extracted from video and text. For videos, the former include signals computed from video frames. The latter are usually limited to what the uploader provided together with the video: the title and description, although these are not guaranteed to be informative. For example, it is not uncommon to find titles corresponding to file names, like *IMG_2309*, and the description is often left empty. Misspellings and the use of colloquial language make it more difficult to extract helpful features from text than it is in the case of news classification. This also implies that the feature space is larger as a word can be spelled in several ways (e.g., compare *loved* with *looooved, luved, luvd*).

Unlike video features which are universal in that they can be found in videos from any part of the world, text features are obviously language dependent. This makes it impossible to obtain a sufficiently large amount of human-labeled data to train a text-based classifier in a supervised way. Therefore, semi-supervised and unsupervised approaches become particularly attractive as they require no or very little labeled data.

In this paper we describe a way of overcoming the above mentioned difficulty of obtaining labeled data. We present a text-based classifier trained on imperfect predictions of a weakly supervised video content-based classifier [1] which acquires a complementary, text-based view on the data. The three main contributions of our work are as follows:

- Given a large collection of videos together with category predictions made by a weakly supervised video based classifier, we train a text based classifier on a complementary set of features in a supervised way. Surprisingly, the resulting classifier not only performs on par but is even more accurate than the original one, even for categories for which video-based predictions are very noisy.
- We use a simple method of combining predictions of the two classifiers and obtain a system which has higher overall accuracy than either of the two taken separately. We also observe that the two source models complement each other in making better predictions on certain classes of categories.
- We experiment with a very large set of features (hundreds of thousands) and show that a simple bag-of-words approach works well. Furthermore, we demonstrate that the performance improves when we add features from a noisy data source, the viewers' comments. We analyse the results and suggest reasons for why this is so.

The novelty of our work is that we show that integrating the viewers' perspective can be highly beneficial for video classification. Most of the previous work has centered around the information provided by the video (image) uploader only. However, we

[1] `http://www.youtube.com`

SIGIR'11, July 24–28, 2011, Beijing, China.
Copyright 2011 ACM 978-1-4503-0757-4/11/07 ...$10.00.

show in the related work review (Sec. 2), in the blog domain user comments have already proved to be a valuable data source which can be used to improve search, predict sentiment, opinion, popularity or trace the development of the blog story. Analysing video comments is a next step towards better understanding of online user interactions.

The paper is organized as follows: Section 2 gives an overview of related work. Section 3 explains how our study is different from previous work and poses new research questions. Sections 4 and 5 describe the general approach and the features we use. Finally, Section 6 presents the experimental results.

2. RELATED WORK

In this section we review video (image) classification work as well as studies which analysed the usefulness of user comments in the domain of weblogs.

2.1 Video and Image Classification

There has been a lot of research on image and video classification and video annotation [8]. In our brief overview we only focus on studies which considered text features.

Lin and Hauptmann [12] describe an approach where two sets of features – video and text – are used and two SVM classifiers are trained on those to predict whether a news video is a weather forecast or not. The paper introduces a novel way of combining predictions by training a meta-classifier.

Feng et al. [7] are perhaps the first to use co-training [3] for image annotation. They show that a smaller labeled set is required to achieve accuracy comparable with that of a supervised learner. They also address the problem of extracting relevant text features from HTML pages. The classification task they consider consists of assigning one of 15 non-abstract labels, such as *tiger, lion* or *cat* to images.

Cai et al. [4] consider the task of clustering image search results. The three kinds of features they use are extracted from text surrounding images, web page links and images themselves. First, the text and link features are used to identify groups of semantically similar images. After that, images within clusters are reorganized and clusters of visually similar images are formed.

Zhang et al. [20] classify videos with respect to five categories (*movies, music, fun, finance* and *news*) by using binary classifiers trained on two separated feature sets – meta-data (i.e., text) and content (i.e., visual). Their experiments demonstrated that category accuracy depends on the type of the classifier and therefore they take advantage of this prior knowledge by using a voting based category-dependent scheme. According to this scheme, the effectiveness of a classifier for predicting a certain category is estimated during the training phase and is later used during classification.

Yang et al. [17] present a study of video classification with respect to eleven categories (e.g., *animal, entertainment, news, sports, music*). They use four different feature sources: visual, audio, text and semantic. The latter can be expressed with video annotations or "visual words". The text features, which include video title, description and tags provided by the uploader, have *tf.idf* values. To amend the feature vector sparsity problem, Yang et al. use WordNet-based similarity measure to propagate the *tf.idf* among similar words. More than 10K human-labeled videos were used in the experiments. A binary classifier is trained for every category-feature combination. For predictions, scores from all the classifiers trained to recognize a certain category are fused to achieve a final confidence.

Cui et al. [5] classify YouTube videos in one of the 15 YouTube categories (e.g., *travel, news, game, people*) and learn from both text and video features. The novelty of their approach lies in that they use the content features during training only to define a better word similarity metric. This is motivated by the fact that computing video features is usually time and resource consuming. During the training phase, videos are clustered based on their video features. After that, a better word similarity metric is defined which assigns higher similarity to words from the same video cluster. For eleven out of 15 categories, the described method outperforms approaches which use only one of the feature sources, or which fuse the predictions of the two.

Huang et al. [11] analyse the effectiveness of a range of text features in detecting extremist videos on YouTube. Video title, description, tags as well as user comments, uploader's name and video category are used as feature sources. The features are broadly classified into lexical (character and word-based features), syntactic (function words and punctuation) and content-specific (word, character and part-of-speech ngrams). In their analysis they claim that all of the mentioned sources are useful because the features from their best model appear in any of the text source. However, it remains unclear whether any of the sources is indeed indispensable for video classification. Apart from that, extremist videos constitute a very special class and it is not guaranteed that the findings of the study generalize to videos from other categories.

Toderici et al. [16] describe a method for automatic tag recommendation system for YouTube videos where new tags are suggested based on a range of video content features. They also propose a method for assigning categories to videos. In an experiment with human raters they get the precision @5 of 70% for category predictions and tag relevance comparable with that of tags provided by uploaders.

2.2 User Comments Analysis

There has been some interest among researchers in the computational linguistics (CL) community in analysing user comments. Like weblog posts, user comments express personal opinion, and the variety found in the comments on a single post can help to identify controversy, sentiment and explain popularity of many topics. The CL research motivates us to look more closely at the comments left on videos and analyse their utility for video categorization.

Herring et al. [10] were the first to include user comments in their weblog analysis. In their study, they look at weblogs from the genre perspective and report comments on posts as one of the characteristic properties of weblogs; no further analysis of user comments is given.

Mishne and Glance [14] present a first in-depth analysis of weblog comments and consider the relationship between the post and the comments it gets. The questions they pose relate to the amount of comments, the post popularity and the use of comments for weblog search and content analysis. Their study indicates that, compared with posts themselves, user comments constitute a considerable data source on their own which is useful for weblog indexing and search. They also present an analysis of how comments account for controversy in the post and how they can help understand different opinions that the post provokes.

Yano and Smith [18, 19] consider the political blogs domain and aim at predicting which users are likely to respond to a post as well as the size (in the number of comments) and the topics of user responses. The unsupervised model they use is a variant of LDA which models the relationship between the post, the commenters and their responses [2, 6].

Recently, Popescu and Pennacchiotti [15] considered the task of discovering controversial events from Twitter. In their supervised

CATEGORY	SUBCATEGORIES
film	bollywood
	animation
	trailers
howto	bodybuilding
	cooking
	beauty_tips
music	arabic
	rap
	rock
news	elections
	documentary
	politics
sports	soccer
	basketball
	water_sports

Figure 1: Sample categories

approach two classifiers are trained to recognize whether a tweet is about an event and whether this event is controversial.

3. RESEARCH QUESTIONS

We consider the task of assigning categories to unrestricted web videos found on YouTube. Unlike the related research, our category set is large, comprising 75 two-level categories. Figure 1 gives an idea of what kind of categories we use in our study. The goal is to find a single most relevant two-level category for the video.

The research questions which have not been answered in previous studies and which we tackle here are as follows:

1. Given that it is prohibitively difficult to obtain a sufficient amount of human-labeled videos for a large set of classes and with a large, sparse feature space, can we learn from imperfect predictions of a classifier trained on a complementary feature set? Is the accuracy of the resulting classifier comparable with the accuracy of the original one? Can a combination of the two classifiers outperform either of the two taken on its own?

2. Which of the categories from our set can be predicted more reliably by the video resp. text-based classifier? Do they indeed capture different semantics of the data?

3. How useful is user-provided textual metadata (i.e., title, description, tags) for video category prediction? Which source of information is particularly helpful?

4. Can categories be reliably predicted from the viewers' comments on the video (comments are known to be extremely noisy)? Furthermore, does the viewers' perspective contribute useful features to improve the performance of the classifier trained on the metadata provided by the uploader? If so, how are the comment features different from those found in the video title, description and tags?

To our knowledge we are the first to use the approach of training a supervised system on noisy predictions of a weakly supervised classifier as previous work has explored co-training and supervised methods. Unlike previous work, our category set is large and includes 75 two-level categories.

4. METHODOLOGY

Our system builds on top of the predictions of a Video2Text classifier introduced by Aradhye et al. [1]. We use Video2Text for three reasons:

- it is weakly supervised and requires no labeled data other than video metadata (title, description) supplied by the user;
- it not only clusters similar videos but also generates a text label for each cluster;
- the resulting label set is comparatively large and is more suited for categorization of unrestricted video content found on YouTube.

Figure 2 describes the training procedure pipeline. Before we proceed with the details of our method, we first describe how Video2Text works.

4.1 Video2Text

Video2Text does not require labeled data to learn which categories to assign to a video. Furthermore, it does not need a predefined set of categories. Instead, it starts from a set of "weak" labels available in the form of video metadata, i.e., in the title, description and uploader-supplied tags. It further creates a vocabulary of concepts which are unigrams and bigrams found in the video metadata. Every concept in the vocabulary is associated with a binary classifier trained from a large set of audio and video signals extracted from the video. The positive instances are the videos which have the concept in their metadata; the negative instances are the videos which do not mention the concept. Initially the vocabulary contains all the possible unigrams and bigrams excluding very frequent and very infrequent ones. The following procedure is repeated until the vocabulary size does not change much or the maximum number of iterations has been reached:

1. A binary classifier is trained for every concept in the vocabulary. The accuracy of the classifier is assessed on a portion of a reserved validation set of videos. Each iteration uses a subset of yet unseen videos from the validation set. The classifier, and hence the concept, is retained only if both precision and recall are above a predefined threshold which is set to 0.7 in the paper.

2. The remaining classifiers are used to update the feature vectors of all videos – the scores assigned to those videos are appended as new features.

The idea behind this iterative procedure is that finer-grained concepts are being learned from concepts added at a previous iteration. E.g., the *video* category is soon discarded as uninformative; *anime* is learned from more general concepts like *cartoon* and *Japanese*. See the original paper by Aradhye et al. [1] for more details on the approach.

In our reimplementation of Video2Text we use the same parameter values as in the original paper. We limit the total number of binary classifiers (and therefore labels) to 75 by discarding those with lower precision or recall. We group together labels related to news, film, sports, howto, music, etc. and thus obtain the final set of 75 two-level categories.

4.2 Categorization with Video2Text

We use Video2Text to assign two-level categories to videos: since all the classifiers are binary, for every video we get a vector of 75 categories together with their likelihood. Figure 1 gives a few examples of the categories we consider. In Figure 2 the output of

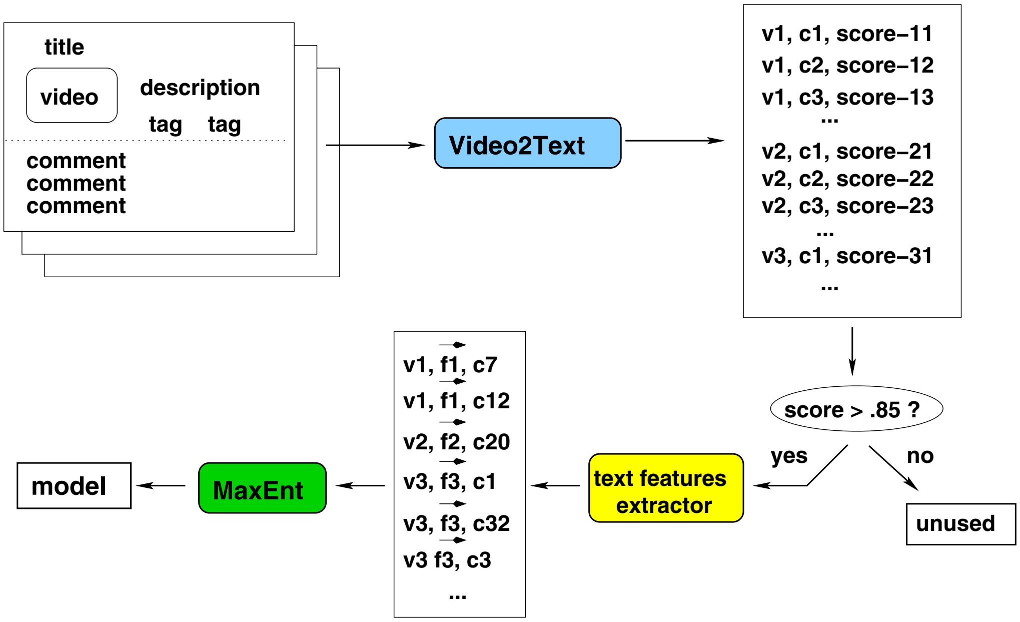


Figure 2: Training procedure pipeline

Video2Text is represented as a list of strings where each string is a sequence of video ID, one of the possible categories and a score which was assigned to this video-category pair – v_i, c_j, s_{ij}. Of course, there is no guarantee that every video would have a single high-scoring prediction.

To train a classifier from text features, we randomly extract approximately 15M videos which have a high-scoring prediction from the Video2Text classifier; in our experiment the threshold is set to 0.85. Videos which get a score smaller than 0.85 are filtered out. For the remaining videos we generate feature vectors, $\vec{f_i}$, from text data associated with them (i.e., title, description, tags, comments; see Sec. 5 for details). The resulting feature vectors and high-scoring categories $v_i, \vec{f_i}, c_j$ are used to train a model with a maximum entropy classifier.

The average number of categories assigned to a video is about 1.3. Based on a small evaluation with human raters, we found that the prediction precision varies across different categories. For example, the threshold of 0.85 results in 100% precision for animation but only 70% for documentary news. Unfortunately, the small size of the evaluation set does not allow us to tune the threshold for every category. However, this noisy setting makes it particularly interesting to see whether a useful model of a comparable quality can be trained from predictions which are known not to be very accurate.

4.3 Distributed MaxEnt

We explore very sparse models for classification where each example covers a very small fraction of the feature space. When combined with the large number of classes (75), the sparsity of a multi-class classification model requires large amounts of training data. The approach presented in this paper allows us to automatically generate training examples for the category classifier. We use a conditional maximum entropy (MaxEnt) optimization criteria to train the classifiers, resulting in a conditional probability model over the classes given the YouTube video:

$$\begin{aligned} c_i^* &= \arg\max_{c \in C} p_\theta(c|v_i) \\ &= \arg\max_{c \in C} p_\theta(c|f_i) \end{aligned}$$

where $c \in C$ is one of the 75 automatically discovered categories. We train the classifier to minimize an L1-regularized loss over the training data $D = ((v_1, c_1), \ldots, (v_m, c_m))$ (the automatically generated, high-precision predictions from the Video2Text classifier), where the loss is the negative log-likelihood of the training data: $L(D) = -\sum_{(f_i, c_i) \in D} \log p(c_i|f_i)$. This results in the following optimization objective function:

$$\theta^* = \arg\min_\theta ||\theta||_1 + \sum_i \log p_\theta(c_i|x_i)$$

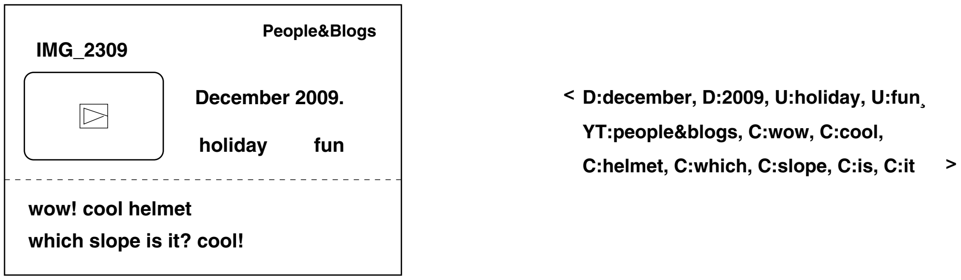
In order to accommodate a large training set, we use a distributed training algorithm known as the Iterative Parameters Mixture update for stochastic gradient descent [13, 9]. This algorithms performs online gradient descent in parallel on subsets of the data, averaging the trained models after each iteration of training and then broadcasting the averaged models back to the parallel training processes. In the current experiments, we train from only 15 million YouTube videos; however, by using a distributed training approach, we are able to train on far larger datasets and can iterate over experimental conditions rapidly.

5. DATA AND MODELS

In this section we present the text-based feature sets and describe system configurations we use. We also introduce a model which predicts video category from the scored set of categories output by Video2Text and a text-based classifier.

5.1 Text Features

The text models we consider differ regarding the text sources from which features are extracted: title, description and user tags


IMG_2309
People&Blogs
December 2009.
holiday
fun
wow! cool helmet
which slope is it? cool!
< D:december, D:2009, U:holiday, U:fun¸
YT:people&blogs, C:wow, C:cool,
C:helmet, C:which, C:slope, C:is, C:it >

(a) Video with its metadata and comments (b) Text-based features

Figure 3: An example video and its feature vector

are the metadata sources provided by the uploader (see the input box in Fig. 2). The video comments provide a viewers' perspective on the video. Finally, as an additional feature we add the YouTube category (one out of 15) which the uploader selected as a relevant one. Note that we are assigning a much more fine-grained classification of 75 categories and the YouTube category is only an indicator of which subset of categories might be relevant.

The features we use are all token-based. We do not use any linguistic preprocessing (e.g., stemming, part-of-speech tagging) other than lowercasing because the data is very noisy and we expect input in any language. To reduce the feature space, we filter out extremely infrequent tokens. Token frequencies are calculated over a set of 150K videos, 10K videos from each of the 15 YouTube categories. Similar to computing document frequency, we count every unique token only once per video. The threshold we use is set to 10 meaning that tokens which appeared in less than 10 videos from the 150K set are discarded.

To distinguish between tokens appearing in different text fields, every token is prefixed with the first letter of where it was extracted from (e.g., *T:xbox, D:xbox, U:xbox, C:xbox* – for the occurrences of *xbox* in the title, description, user tags and comments, respectively). Figure 3 gives an example video and the features extracted from its metadata and comments. The title of the video (*IMG_2309*) is the name of the corresponding file and is not sufficiently frequent to be included in the feature vector. The word *cool* occurs twice in the comment but is included only once as a feature.

Since the number of user comments varies considerably and can be as high as hundreds of thousands, we extract comment features from the 200 most recent comments for each video. We found that using token frequency results in lower accuracy: words from comments on videos with many comments may get extremely high value. However, using the inverted video frequency count as the value for comment features turns our to have a positive effect.

To address the question of which source of textual information is particularly useful, we try several system configurations. In what follows, we consider the following models which differ with respect to the feature sources they use. We train a classifier for each of the metadata sources – title (TITLE), description (DESCR), user tags (U-TAGS). We also look at the combination of all three (TDU) and at a model which is also informed about the YouTube category chosen by the video uploader (TDU+YT). Finally, we consider a model trained from the user comments on the video as the only feature source (COMM) and a model which profited from all five text sources (TDU+YT+C).

5.2 Combined Classifier

To see if the combination of the two views, video and text based, is beneficial, we consider a simple "meta" classifier which ranks video categories based on the predictions made by the two classifiers. The ranking is obtained by converting the video-based predictions into a probability distribution and multiplying it with the probability distribution returned by MaxEnt. Note that any of the text-based classifiers (e.g, TDU or C) can be used in the combination. This combination approach proved to be effective in our experiments, but may not be the optimal method. We do not explore optimizing the combination approach in this work as our goal is simply to support the hypothesis that the two models provide complimentary predictions. Figure 4 schematically represents how combining works for a single video (the normalization to a probability distribution step is omitted as it is not relevant).

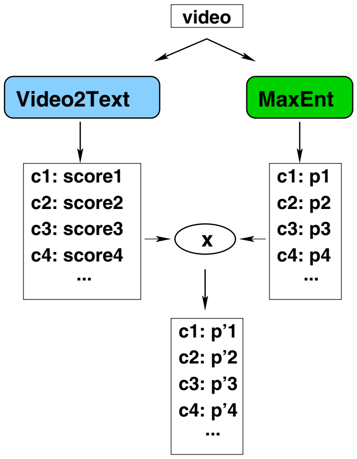


Figure 4: The combination procedure for a single video

The idea behind the multiplied combination is that each of the classifiers has veto power and can thus help rule out wrong categories. The final prediction of the combined classifier is the one with the highest product score. For example, the text-based classifier (MaxEnt) may distribute the total probability mass between two categories – *sports/winter_sports* and *sports/bikes* – because words related to both categories, like *helmet*, appear in the metadata. The video-based classifier (Video2Text) may assign a very low score to the former category (e.g., there is no snow in the video but fields and woods), a somewhat high score to *car/racing* and a high score to *animals/wild_animals*. The product of the two will then change the ranking so that only the category which is likely according to both models, *sports/bikes* gets the highest score.

Of course, the utility of the combined model presumes that the two models have complementary views on the data. In the experiments section we are going to show that this is indeed the case.

6. EXPERIMENTS

To test our hypotheses, we employ human raters and collect their judgments about the relevance of predicted categories. The three classes of models we want to compare are: (1) the Video2Text model, (2) the models trained on text features, and (3) the combination models, described in Section 5.2. Unfortunately, given the size of our category set and all the classes of text features we experiment with, it is prohibitively difficult to collect enough human judgments to compare all the text and the combined models: we would need to evaluate seven text-based models, Video2Text and at least one combined model over 75 categories. Therefore, we use a portion of automatically labeled data to get an idea as to which of the text models is likely to be the most accurate one. This “winner” model is further compared with the video classifier in an experiment with human raters. It is also the model used in combination with Video2Text. In this section, we first describe the results of the preliminary evaluation on the development set and then proceed with the results of the experiments with human raters.

6.1 Preliminary Evaluation of Text Models

As with out training data selection, we randomly collect about 100K videos which get a high-scoring prediction from the video model. A prediction is considered correct if it has a score of at least 0.85 from Video2Text. The average number of “correct” categories assigned to a video is 1.2, and for a video to count as correctly classified the text-based prediction has to be in the set of video-assigned categories. For evaluation we first consider videos which have at least one comment (1+), although we do not filter out cases when no sufficiently frequency feature can be extracted from the video. This restriction is applied to allow for a fair comparison of the models which use comment features. It also helps filter out videos with very little textual information because such videos are often uploaded for sharing with very few people and are not expected to be seen by many. To check whether more accurate predictions are made for videos with many comments, we measure accuracy on a subset of videos with at least ten comments (10+). Although less than 10% videos have ten comments or more, this is an important class of videos as they get considerably more views than those with very few comments. As with the training phase, the maximum number of comments we retrieve is 200.

Of course, the results on the development set (Table 1) should be taken with a grain of salt as predictions of Video2Text are not perfect and some videos may not have a correct label. However, we believe that significant differences are indicative of a better model.

Videos with at least one comment.

Concerning single text feature sources for videos with at least one comment (Accuracy 1+ in Table 1), user tags turned out to be particularly helpful, while comments and descriptions are the least useful sources. This is not surprising given that descriptions, unlike titles, can be left empty, and that a single comment like *lol* or *first* does not give any clue to the video category. Predictions made from user tags, which may be left blank, are more accurate because tags themselves constitute an open category set which the uploader selects as most relevant. However, a combination of all the features significantly improves the performance (TDU); providing YouTube category further refines the predictions (TDU+YT). The higher accuracy is largely due to the fact that in cases where user tags are missing, the title and possibly the description help predict the right category.

Videos with at least ten comments.

Interestingly, the accuracy numbers are much higher for all the models on videos with at least ten comments, even for models which do not use comments as a feature source. The reason is that those videos are of a better quality and have more informative titles and descriptions, which we believe is why they are viewed and commented on by more users. The boost in accuracy is particularly striking for the models using comments. Out of all the single-source models, the comments (COMM) model is the one which substantially outperforms all other models, including the combinations of all models (TDU and TDU+YT). This comes as a surprising result because it means that, given enough comment volume, the viewers can provide us with more helpful signals about video content than the uploaders themselves. The question of whether the uploader is often unable to give an adequate description to his video or whether he is not interested in it is left for future research.

For both evaluation sets (1+ and 10+), the best model is TDU+YT+C which combines all the text sources. It is this model whose performance we assess with human raters and which we use to combine with the video model.

6.2 Experiments with Human Raters

From the 100K videos in the development set, we extract the total of 750 videos equally from the 15 YouTube categories, 50 each. This way we aim at getting a balanced sample of the data with videos from all the possible categories. With the mentioned restriction on the maximum number of retrieved comments (200), the average number of comments and standard deviation in the evaluation set (1+) are 9.6 and 26, respectively. Thus, most videos have only very few comments.

A video and a category were presented to a human rater who was asked to rate it as either *fully correct (3), partially correct (2), somewhat related (1)*, or *off-topic (0)*. The instructions made clear that *partially correct* should be selected when only one of the two levels of the category is applicable. The raters did not see any meta information about the video or the comments. For every video-category pair we obtained exactly three ratings. 12% of the videos were excluded because they were no longer available for watching during the time of the experiment. The three ratings of a video were then converted into a single binary rating by summing the numeric values associated with them, normalizing the sum (i.e., dividing it over nine) and rounding the resulting score. A score of at least 0.5 makes a category count as a correct one. The accuracy we report in Table 2 is computed as $\frac{|correct|}{|total|}$. Note that our evaluation is very strict: for a category to count as correct, two raters must find it at least *partially correct* and the third rating cannot be *off-topic*. A category rated as *somewhat related* does not count as a correct one. A less strict evaluation would obviously result in higher accuracy numbers.

	VIDEO2TEXT	TDU+YT+C	COMBINED
Accuracy 1+	34.6%	38.5%	40.5%
Accuracy 10+	38.9%	43.5%	45.7%

Table 2: Accuracy on the set rated by humans

Video2Text vs. TDU+YT+C *vs. combination.*

The most remarkable result is that the text-based model performs not only on par with the video model it was trained from but is actually significantly better. Furthermore, the simple combination method we tried resulted in improved accuracy because, as we hypothesized, each of the models can help pruning wrong predictions. These findings allow us to answer the first group of questions posed

	TITLE	DESCR	U-TAGS	COMM	TDU	TDU+YT	TDU+YT+C
Accuracy 1+	39.4%	35%	42.4%	35.6%	48.1%	51%	53.2%
Accuracy 10+	44.8%	40.7%	49.2%	57.8%	54.7%	56.5%	62.8%

Table 1: Accuracy for different models on a set of videos with at least one or ten comments, respectively

in Section 3 affirmatively. The text-based classifier is not redundant but complementary to the video-based one. The complementarity explains the increased performance of the combination model.

Differences between the video and text based views. Having looked at the predictions made by the video and the text based models, we found that certain abstract categories which are difficult to recognize from video features (e.g., *news, finance* or *comedy*) were much more accurately predicted by the text model (more than 50% in precision). This is encouraging because it means that a competitive text-based classifier can be trained from very noisy predictions. However, certain categories are better predicted from video signals: *howto/dance* or *film/animation*. The combined model outperformed the source models across many categories; however, there are not specific categories in which it is consistently better. Thus, to answer the second group of questions from Section 3, the two views do capture different semantics of the data, the text classifier being more useful for abstract categories which do not have obvious visual features.

Performance on videos with many comments. We looked at the subset of videos in the rated set which had at least ten comments, 110 out of 660 videos satisfied this condition (10+ in Table 2). As in the preliminary evaluation on the development set, we found that the performance of all the models improves with the number of comments. However, the differences between the models' performance remain the same, the combined model being 6-7% better than the video-based one.

The improved performance on the 10+ subset is consistent with what we observed in our evaluation on the development set: that prediction accuracy increases with the number of comments. Apparently, a reasonable explanation is that more comment-based features can be generated for such videos, and this would allow us to see the comments stream as a valuable source of information about the video content. However, as we have seen in the previous subsection, on videos with many comments more accurate predictions are made even by the models which do not extract any features from the comments (e.g., TDU+YT). Therefore, an increase in accuracy can be expected also for comment-free models. To investigate the question of whether comments indeed provide helpful clues and why this is so we did one more experiment with human raters.

6.3 Impact of Comments

To address the final question in Section 3 and get an insight into why the model with comment features outperforms (at least on the development set) the one that lacks this feature source, we collected human ratings for videos which got different predictions from TDU+YT and TDU+YT+C. In our analysis we focused on three unrelated categories – *film/bollywood*, *music/arabic* and *howto/ bodybuilding* – and selected videos with at least ten comments to make the data set more informative. Thus, for each category c and each model m we extracted ten videos $v_1, ...v_{10}$ where m ranked c the highest and m' assigned some other c' the highest score. Pairs v_i, c and v_i, c' were rated by human experts.

Using the same instructions and approach as in the previous evaluation, we collected three ratings for the total of 120 ($3 \times 10 \times (2 + 2)$) video-category pairs. The average accuracy for the three selected categories as well as the overall accuracy on all the rated videos are presented in Table 3.

	TDU+YT	TDU+YT+C
bodybuilding	0%	22.2%
bollywood	14.3%	77.8%
arabic music	37.5%	100%
overall	26.1%	59.2%

Table 3: Results for the models with and without comment features

The difference between the two models is remarkable and indicates that comments do significantly improve prediction accuracy, provided that there are enough of them (e.g., at least ten). For almost all the videos where the two models got different ratings from the humans it was the comment-informed model whose prediction was correct. Having looked at the rated examples, we identified the following reasons for why viewers' comments were so helpful:

1. In some cases, there were no tags, and title and description were very short and provided little information about the content of the video. Some descriptions and titles contained words indicative of the correct category but they were misspelled, e.g., *boliwook* instead of *bollywood*. Here, comments became a **substitute** for a proper title / description.

2. Some videos had descriptions which could be applicable to many categories, e.g., *"Always playing this song, sorry for poor quality"*, *"This video has no meaning of life"* or *"From my favorite episode"*. The comments, on the other hand, referred to a particular scene and mentioned names of actors, players, characters or brands which made it clear that the video was from a (Bollywood) movie, a soccer game, a cartoon, or about a device. Here comments helped to further **disambiguate** the category.

3. Words from some titles and descriptions were misleading on their own and suggested a category unrelated to the video. E.g., *EZ, curls, pounds, girl* in the *People & Blogs* category made TDU+YT rank *howto/beauty_tips* as the most likely category. Comments like the following helped to **correct** wrong predictions: *"pancake chest, monstrous face, massive waist with mini gut, and weird, pencilish arms."* made it clear that the most relevant category is *howto/bodybuilding*.

Thus, our analysis on a subset of three unrelated categories demonstrates that the video comment stream should not be neglected as a feature source. It is noisy but the signals it provides us with are in many cases indispensable for video classification because the video title, description and tags are not always indicative of the most relevant category.

7. CONCLUSIONS

In this paper we presented a text-based approach to the task of assigning relevant categories to videos. We showed that a competitive classifier can be trained on high-scoring predictions made by a weakly supervised classifier learned from video features. We also showed that the two provide complementary views on the data and that a simple product model which combines two sets of predictions outperforms each of them taken on their own. The prediction rate of 41% is quite high given the unprecedentedly large size of our category set (75) and that we did not use any human-labeled data. Furthermore, for popular videos with at least ten comments we achieved an even higher accuracy of 46%.

We found that all of the text sources – title, description, tags and comments – are helpful for category predictions. A more significant result is that accurate predictions can be made from the users' comments, provided that there are enough of them. We analysed a set of video-category pairs rated by humans and suggested three reasons for why a model which also looks at the viewers' comments outperforms the one which lacks this information source.

Motivated by our findings as well as by recent research in the weblogs domain, in the future we are planning to investigate the usefulness of user comments for other tasks.

8. REFERENCES

[1] H. Aradhye, G. Toderici, and J. Yagnik. Video2Text: Learning to annotate video content. In *Proceedings of the 1st International Workshop on Internet Multimedia Mining,* Miami, Florida, December 6, 2009.

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

[3] A. Blum and T. Mitchell. Combining labeled and unlabeled data with Co-Training. In *Proceedings of the 11th Annual Conference on Learning Theory,* Madison, Wisc., 24–26 July, 1998, pages 92–100, 1998.

[4] D. Cai, X. He, Z. Li, W.-Y. Ma, and J.-R. Wen. Hierarchical clustering of WWW image search results using visual, textual and link information. In *Proceedings of the 6th SIGMM ACM International Workshop on Multimedia Information Retrieval,* New-York, USA, 15-16 October, pages 952–959, 2004.

[5] B. Cui, C. Zhang, and G. Cong. Content-enriched classifier for web video classification. In *Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* Geneva, Switzerland, 19–23 July 2010, pages 619–626, 2010.

[6] E. Erosheva, S. Fienberg, and J. Lafferty. Mixed membership models of scientific publications. In *Proceedings of the National Academy of Sciences*, pages 5220–5227, 2004.

[7] H. Feng, R. Shi, and T.-S. Chua. A bootstrapping framework for annotating and retrieving WWW images. In *Proceedings of the 6th SIGMM ACM International Workshop on Multimedia Information Retrieval,* New-York, USA, 15-16 October, pages 55–62, 2004.

[8] S. Fischer, R. Lienhart, and W. Effelsberg. Automatic recognition of film genres. In *Proceedings of the 3rd ACM International Conference on Multimedia,* San Francisco, CA, 5-9 November, pages 295–304, 1995.

[9] K. Hall, S. Gilpin, and G. Mann. MapReduce/Bigtable for distributed optimization. In *Advances in Neural Information Processing Systems Workshop on Learning on Cores, Clusters and Clouds*, 2010.

[10] S. C. Herring, L. A. Scheidt, S. Bonus, and E. Wright. Bridging the gap: A genre analysis of weblogs. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences,* Hawaii, 5–8 January, 2004, 2004.

[11] C. Huang, T. Fu, and H. Chen. Text-based video content classification for online video-sharing sites. *Journal of the American Society for Information Science and Technology*, 61(5):891–906, 2010.

[12] W.-H. Lin and A. Hauptmann. News video classification using SVM-based multimodal classifiers and combination strategies. In *Proceedings of the ACM International Conference on Multimedia,* Huan-les-Pins, France, 1-6 December, pages 323–326, 2002.

[13] R. McDonald, K. Hall, and G. Mann. Distributed training strategies for the structured perceptron. In *Proceedings of Human Language Technologies Conference of the North American Chapter of the Association for Computational Linguistics,* Los Angeles, CA, 1–6 June 2010, pages 456–464, 2010.

[14] G. Mishne and N. Glance. Leave a reply: An analysis of weblog comments. In *Proceedings of the 3rd Annual Workshop on the Weblogging Ecosystem,* Edinburgh, Scotland, 23 May, 2006, 2006.

[15] A.-M. Popescu and M. Pennacchiotti. Detecting controversial events from Twitter. In *Proceedings of the ACM 19th Conference on Information and Knowledge Management (CIKM 2010),* Toronto, Canada, 26–30 October, 2010, pages 1873–1876, 2010.

[16] G. Toderici, H. Aradhye, M. Paşca, L. Sbaiz, and J. Yagnik. Finding meaning on YouTube: Tag recommendation and category discovery. In *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition,* San Francisco, CA, 13–18 June, 2010.

[17] L. Yang, J. Liu, X. Yang, and X.-S. Hua. Multi-modality web image categorization. In *Proceedings of the 8th ACM SIGMM International Workshop on Multimedia Information Retrieval,* Augsburg, Germany, 28-29 September, pages 265–274, 2007.

[18] T. Yano, W. W. Cohen, and N. A. Smith. Predicting response to political blog posts with topic models. In *Proceedings of Human Language Technologies Conference of the North American Chapter of the Association for Computational Linguistics,* Boulder, Colorado, 30 May – 5 June 2009, pages 477–485, 2009.

[19] T. Yano and N. A. Smith. What's worthy of comment? Content and comment volume in political blogs. In *Proceedings of the 4th International Conference on Weblogs and Social Media,* Washington, DC, USA, 23 – 26 May 2010, pages 359–362, 2010.

[20] R. Zhang, R. Sarukkai, J.-H. Chow, W. Dai, and Z. Zhang. Joint categorization of queries and clips for web-based video search. In *Proceedings of the 8th ACM SIGMM International Workshop on Multimedia Information Retrieval,* Santa-Barbara, CA, 26-27 October, pages 193–202, 2006.