CRMActive: An Active Learning Based Approach for Effective Video Annotation and Retrieval

Moitreya Chatterjee
USC Institute for Creative Technologies
Playa Vista, CA, USA
metro.smiles@gmail.com

Anton Leuski
USC Institute for Creative Technologies
Playa Vista, CA, USA
leuski@ict.usc.edu

ABSTRACT

Conventional multimedia annotation/retrieval systems such as Normalized Continuous Relevance Model (NormCRM) [7] require a fully labeled training data for a good performance. Active Learning, by determining an order for labeling the training data, allows for a good performance even before the training data is fully annotated. In this work we propose an active learning algorithm, which combines a novel measure of sample uncertainty with a novel clustering-based approach for determining sample density and diversity and integrate it with NormCRM. The clusters are also iteratively refined to ensure both feature and label-level agreement among samples. We show that our approach outperforms multiple baselines both on a new, open dataset and on the popular TRECVID corpus at both the tasks of annotation and text-based retrieval of videos.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering, Retrieval Models; H.5.1 [**Multimedia Information Systems**]: Video (e.g., tape, disk, DVI)

Keywords

Active Learning; Clustering; Uncertainty; Informativeness

1. INTRODUCTION

The ubiquity of multimedia content in our daily lives requires effective tools for multimedia annotation and retrieval. Multimedia annotation tools automatically annotate image or video content (samples) with text labels specifying different objects, events, etc. called *concepts*. A typical multimedia retrieval system, on the other hand, ranks the multimedia samples based on their relevance to the user's text query. Generally, the retrieval is done by comparing the query to the sample concept labels. Thus an exhaustive annotation of the sample is often a pre-requisite for such retrieval systems.

Normalized Continuous Relevance Model (NormCRM) [7] is an example of a technique that allows for a direct retrieval of samples without having to annotate them. However training this model (like many others), requires fully annotated data. The human-effort costs of concept annotation is significant and this raises an interesting research question: is there a way to achieve a decent annotation/retrieval performance without requiring a fully annotated training dataset?

The community has taken to *Active Learning* to address this issue [5]. Active Learning, is a machine learning technique that interactively selects unlabeled samples and queries an oracle to provide labels for the samples. Such a system outputs an order of labeling the samples such that a decent annotation/retrieval performance is achieved before all unlabeled data is queried. A typical active learning system consists of a learning engine, which does the annotation/retrieval and a sample selection engine, responsible for determining the labeling order of the unlabeled samples.

In this work, we use NormCRM as the learning engine and propose a novel sample selection algorithm. We call the system CRMActive and apply it for video annotation and retrieval tasks. The algorithm uses a measure of *informativeness* for ranking unlabeled samples during active learning. The informativeness combines a new measure of sample uncertainty with a novel cluster-refinement approach for determining sample density and diversity. Our experiments show CRMActive outperforms a state-of-the-art baseline.

2. PROPOSED APPROACH

Normalized Continuous Relevance Model (NormCRM) is a generative annotation/retrieval technique [7]. Let's consider a video sample I defined by a M-dimensional feature vector $\mathbf{r}$ and $\mathcal{V}$ be the vocabulary of all concept labels (each concept 1 word long). NormCRM defines conditional probability for using a label word $w \in \mathcal{V}$ to annotate the video I, as $P(w|\mathbf{r}) = P(w, \mathbf{r})/P(\mathbf{r})$. Lavrenko et al. [7] suggest that for annotation we pick the top-k words with highest $P(w_i|\mathbf{r})$, $i = 1, 2, ..., k$. For the task of retrieval using a query word w, we pick the top-t videos with highest $P(w|\mathbf{r}_i)$, $i = 1, 2, ..., t$. In both cases, the joint-distribution of words and features $P(\mathbf{w}, \mathbf{r})$ is estimated from the training data by

$$P(\mathbf{w}, \mathbf{r}) = \sum_{J \in T} (P(J) \prod_{w \in \mathbf{w}} P(w|J) \prod_{r_i \in \mathbf{r}, i=1}^{M} P(r_i|J)),$$

where T is the set of training video samples and $\mathbf{w}$ is the set of words in question.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ICMR'15, June 23–26, 2015, Shanghai, China.
Copyright © 2015 ACM 978-1-4503-3274-3/15/06 ...$15.00.
http://dx.doi.org/10.1145/2671188.2749342.

However, NormCRM requires a fully annotated data for training. To address this, we propose an Active Learning approach that combines NormCRM with a sample selection engine. The engine selects samples for annotation based on their *informativeness*, which we calculate by combining measures of sample *uncertainty*, *density* and *diversity*.

Sample Uncertainty is a measure of how uncertain the learning engine is about the sample labels. Entropy and distance of the sample from the decision boundary have been explored as sample uncertainty measures [9, 12]. However, these techniques do not capture the ambiguity between the relevant labels and the irrelevant ones for NormCRM-based models. Hence, we define a novel measure of uncertainty of an unlabeled sample (a M-dimensional vector $\mathbf{x}$) as:

$$unct(\mathbf{x}) = \frac{1}{P(w_1|\mathbf{x}) - P(w_{k+1}|\mathbf{x})}, \tag{1}$$

where $w_1, ..., w_k$ (in decreasing order of relevance) are the top-k most relevant labels assigned to $\mathbf{x}$. The denominator in Eq. 1 gives a measure of the gap (distance) between the posterior probabilities of the most relevant label and the first irrelevant one and can thus be used to obtain uncertainty.

Sample Density is a measure of how likely a certain sample is to occur given the underlying distribution that generated the data while a high *Sample Diversity* score ensures that the samples chosen for labeling aren't too similar to each other. To compute sample density and diversity, we start by clustering all samples in the training data $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, consisting of the initial labeled training data $\mathcal{L}$ and the unlabeled training data $\mathcal{U}$ ($\mathcal{X} = \mathcal{L} \cup \mathcal{U}$). We first represent every sample in the visual feature space and perform X-Means clustering. X-Means is a variant of K-Means, which automatically picks the parameter K by comparing the Bayesian Information Criterion (BIC) scores of the clustering system for a range of values of K and picking the one with an optimal score [8]. We then check if every labeled sample shares a concept with at least one other labeled sample in the same cluster. If we find a sample that shares no labels, we create a new cluster for it and redistribute unlabeled samples from the original cluster between the old and the new clusters using 2-Means.

In order to measure the extent of agreement among the labeled samples in a cluster, both in terms of their visual features and their labels, we use *Empirical Entropy* [3]. For a cluster C, it is defined as:

$$h^C = -\frac{1}{n}\sum_{i=1}^{n} \log(\frac{1}{n}\sum_{j=1}^{n} K(\mathbf{x}_i, \mathbf{x}_j)), \tag{2}$$

where there are $n > 1$ labeled samples in the cluster and $K(.,.)$ is a kernel function. A kernel is a mapping : $\chi \times \chi \rightarrow \mathbb{R}$, where χ is the input space. A kernel may be considered as a measure of similarity. For continuous input spaces, such as video features, a Gaussian kernel is often used [13]:

$$K_{Gauss}(\mathbf{x}, \mathbf{x}') = \exp(-||\mathbf{x} - \mathbf{x}'||^2/2\sigma^2),$$

where $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$. For discrete input spaces, such as the space of labels, a Bernoulli product kernel may be used [6]:

$$K_{Bern}(\mathbf{x}, \mathbf{x}') = \prod_{d=1}^{D}[(\gamma_d^{x_d} \times \gamma_d^{x'_d}) \times (1-\gamma_d)^{(1-x_d)} \times (1-\gamma_d)^{(1-x'_d)}],$$

where $\mathbf{x}, \mathbf{x}' \in \{0,1\}^D$, x_d, x'_d shows the presence (1) or absence (0) of the d^{th} concept and γ_d is the probability of the d^{th} concept occurring. In order to capture the notion of sample similarity both from the visual and label perspectives, we define a new kernel as a combination of the two [4]:

$$K(\mathbf{x}, \mathbf{x}') = K_{Bern}(\mathbf{x}, \mathbf{x}') \times K_{Gauss}(\mathbf{x}, \mathbf{x}')$$

Once we clustered the sample videos, we compute the sample density of an unlabeled sample $\mathbf{x}$ in cluster C as

$$den(\mathbf{x}) = \frac{p(\mathbf{x})}{\max\limits_{\mathbf{x}_i \in \mathcal{X}} p(\mathbf{x}_i)},$$

where $p(\mathbf{x})$ is the kernel density estimate:

$$p(\mathbf{x}) = \frac{1}{|C|}\sum_{\mathbf{x}_i \in C} K_{Gauss}(\mathbf{x}, \mathbf{x}_i)$$

and $|C|$ is the total number of samples in cluster C.

Algorithm 1 CRMActive

```
Input: The set $\mathcal{L} = \{\mathbf{l}_1, \mathbf{l}_2, ..., \mathbf{l}_P\}$, their labels $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_P\}$ where $\mathbf{y}_i \in \{0,1\}^D$, the set $\mathcal{U} = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_Q\}$ and $K$ :- nos. of samples to pick in a batch.
Output: The set $\mathcal{L}$, containing the order in which the unlabeled samples are labeled.
Algorithm:
Perform X-Means, using the visual features, on the set of $\mathcal{L} \cup \mathcal{U}$ samples. Say, T be the optimal number of clusters and let $rep(C_i)$ denote the representative sample of cluster $C_i$.
Check if $\forall \mathbf{l}_j, \mathbf{l}_j \in \mathcal{L}, \mathbf{l}_j \in C_k, \mathbf{l}_j$ shares $\geq 1$ concept with at least 1 labeled sample in $C_k$, otherwise call Redistribute($C_k, \mathbf{l}_j$).
$h_{worst}$ := NIL // Initialize $h_{worst}$
while $\mathcal{U} \neq \phi$ do
    Train NormCRM using $\mathcal{L}$, evaluate model on test set.
    Update $h_{worst}$ to max. entropy value among all clusters with at least 2 labeled samples
    Compute $Info(\mathbf{x}_i), \forall \mathbf{x}_i \in \mathcal{U}$
    Pick top-$K$ samples, $\mathbf{Lab} = \{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_K\}$ for labeling.
    $\mathcal{L} := \mathcal{L} \cup \mathbf{Lab}, \mathcal{U} := \mathcal{U} - \mathbf{Lab}$ // Update the lists
    // Now refine the clusters based on newly labeled samples
    for $j = 1, 2, ...,$ K do
        if $h_{worst}$ = NIL then // If $h_{worst}$ is not set
            Check if sample $\mathbf{a}_j, \mathbf{a}_j \in C_k$ shares $\geq 1$ concept with at least 1 labeled sample in $C_k$, otherwise call Redistribute($C_k, \mathbf{a}_j$).
        else// Determine which sample in $C_k$ to knock out
            Compute $h^{C_k}$, where $\mathbf{a}_j \in C_k$ //$C_k > 1$ labeled sample
            if $h^{C_k} > h_{worst}$ then // Exceeds threshold
                for $r = 1, 2, ...,$ # labeled samples in $C_k$ do
                    $C'_k := C_k - r^{th}$ labeled sample in $C_k$
                    if $h^{C'_k} \leq h_{worst}$ then // Meets threshold
                        $\mathbf{W} := r^{th}$ labeled sample
                        Redistribute($C_k, \mathbf{W}$) // Split cluster
                        break
```

Our definition of the sample density, though similar to Zha et al. [13], differs by using clusters, which are refined (see later in this section), to determine the neighboring samples of $\mathbf{x}$ rather than a static set of its k-nearest neighbors.

```
procedure REDISTRIBUTE(Samples in $C_k$, $\mathbf{a}$)
    Input: Set of all samples in cluster $C_k$ & the seed sample $\mathbf{a}$
    Output: Updated set of clusters
    Algorithm:
    Create a new cluster, $C'_k$, with $\mathbf{a}$ as the centroid.
    Perform 2-Means on the unlabeled samples of cluster $C_k$ with $rep(C_k)$ and $\mathbf{a}$ as the two initial cluster centroids.
    Update $rep(C'_k)$ as the representative sample of cluster $C'_k$.
    Determine the centroid of the labeled and the remaining unlabeled samples in $C_k$ and similarly update $rep(C_k)$.
```

To compute the sample diversity, we use the angular distance between features similar to Brinker's technique [2]. However we choose only the representative samples of every cluster (i.e. the sample closest to the cluster centroid),

$rep(C)$, rather than all the samples in $\mathcal{X}$, to gain speed. Diversity of the unlabeled samples is thus, defined as:

$$div(\mathbf{x}) = 1 - \max_{\mathbf{x}_i \in \mathcal{S}} \frac{K_{Gauss}(\mathbf{x}, \mathbf{x}_i)}{\sqrt{K_{Gauss}(\mathbf{x}, \mathbf{x}) \times K_{Gauss}(\mathbf{x}_i, \mathbf{x}_i)}},$$

where $\mathcal{S}$ is the set of all T cluster representatives $\mathcal{S} = \{rep(C_1), rep(C_2), ..., rep(C_T)\}$. We combine these measures to define the *informativeness* of an unlabeled sample $\mathbf{x}$:

$$Info(\mathbf{x}) = \lambda_1 \times unct(\mathbf{x}) + \lambda_2 \times den(\mathbf{x}) + \lambda_3 \times div(\mathbf{x}).$$

We rank the unlabeled samples in the order of decreasing $Info(\mathbf{x})$ score to select a batch of top-K samples for labeling. While Zha et al. use a combination of sample local structure, density, diversity, and relevance to score the samples [13], our approach differs, most notably, in the use of clustering and the novel uncertainty measure.

Equation 2 shows that a cluster with low inter-sample disagreement has a low entropy. As more samples in a cluster C are labeled, the disagreement among its labeled samples increases. This changes the empirical entropy h^C in a monotonically non-decreasing fashion. We refine the clusters by doing the following: After each labeling batch the algorithm finds the cluster with the worst entropy and uses its h^C as the threshold to decide whether to keep or split a cluster during the next batch. This is repeated for successive iterations. If a newly labeled sample increases the cluster entropy beyond the batch threshold, then we use grid search to find the first sample without which the cluster meets the entropy threshold. We create a new cluster for the sample and rearrange the unlabeled samples via 2-Means (see Algorithm. 1).

3. EXPERIMENTS

We conduct two sets of experiments. In each set, the dataset is divided into training and test subsets. For the first set of experiments, the task of an algorithm is to annotate a test video with a subset of concepts from the vocabulary. The algorithm starts with the training data set divided into labeled ($\mathcal{L}$) and unlabeled ($\mathcal{U}$) parts. Initially only a small subset of the training set is considered to be labeled. The algorithm uses this information to annotate the test set with concept labels. For the next step, the algorithm selects a batch of K unlabeled training samples, we reveal the labels for the selected samples, and the algorithm repeats the annotation task. For every iteration, we report the average precision (AP) scores on the test-set for each concept.

In the second set of experiments, an algorithm ranks the test samples by their similarity to a single word query without annotating the test samples. Again, the algorithm starts with the training dataset divided into labeled and unlabeled parts. For each concept label in the vocabulary, the algorithm ranks the test samples by their similarity to the concept. It then selects a batch of K unlabeled training samples, we reveal the labels for the selected samples, and the algorithm repeats the ranking task. For each round, we report the AP scores for the top 5 images/videos.

3.1 Datasets

TRECVID 2007: The TRECVID 2007 video corpus has 110 short video clips [1] . Each frame in every video is annotated with at most 16 concept labels selected from a set of 36 concepts such as “crowd”, “building”, “airplane”, etc. This corpus has been used extensively in video annotation experiments [13]. For every frame we compute a 225-dimensional feature vector (color moment, edge orientation histogram, wavelet PWTTWT texture) as described in the work of Zha et al. [13]. We test our model on the frames from 13 randomly selected videos and we use the rest of the data (frames from 97 videos) for training. We selected 4000 frames from the training data as the initial set of labeled samples $\mathcal{L}$, containing at least 1 positive example of every concept. We set, batch size, K to 2400.

USC SmartBody: SmartBody is an open virtual character animation platform. It ships with a library of 274 animations such as walking, pointing, eye-brow raising, lip corner stretching, etc. [11]. The animations are defined on a 3D skeleton of 119 individual joints and the joints 3D coordinates are available from the SmartBody API. We annotated each animation using at most 6 concept labels from a set of 30 labels such as “Legs”, “Arms”, “Face”, “Left”, “Right”, etc. The X-axis of Figure 1 gives an exhaustive list of all the concepts. We annotated the animations at the video clip level (i.e. the individual frames are not annotated). We handpicked 9 out of 119 joints (neck, left(L)/right(R) shoulders, L/R elbows, L/R hip joints, and L/R knees). For each frame in an animation we calculated the skeleton angles at these joints [10] and encoded the differences between the minimum and the maximum values for the angles during the whole animation sequence as a 9-dimensional feature vector. The dataset will be available at our web site[1]. We randomly selected 24 animations for testing and we use the rest of the data (250 animations) for training. We selected 40 animations from the training data as the initial set of labeled samples $\mathcal{L}$, containing at least one positive example of each concept. We now set, batch size, K to 23.

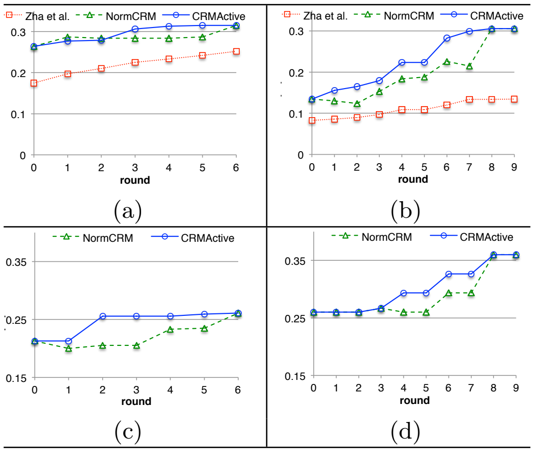
3.2 Baseline Systems

For the annotation task we compare CRMActive with two methods. The first one is an active learning system that uses NormCRM as the learning engine while the samples are selected randomly. The results are averaged over 3 runs with different random seeds. The second baseline is the method proposed by Zha et al. (state-of-the-art) [13]. We determine the two NormCRM smoothing parameters λ and β [7] and the validated parameters of the second baseline using 10-fold cross-validation on the first annotation batch. These values are then fixed for successive rounds. The values of the fixed parameters for the second baseline are reused from the paper [13]. For CRMActive, probability γ_d, is re-estimated from the labeled training data on each annotation batch and the weighting parameters $\lambda_i = \frac{1}{3}, i = 1..3$. Finally, both NormCRM and CRMActive work by ranking annotation concepts, so we assign the top 16 concepts for TRECVID 2007 and the top 6 for SmartBody as relevant. For direct retrieval, CRMActive is compared only with the first baseline discussed above, since no prior work is known.

3.3 Results and Discussion

The results in Table 1 show that the NormCRM-based models, i.e. the first baseline (NormCRM) and CRMActive, generally perform better than the Zha et al. approach for annotation. We believe that this is because NormCRM captures the inter-label correlation, while Zha et al. trains individual classifiers for every concept. Also, the NormCRM-based systems jointly model the labels and features and it

[1] http://nld.ict.usc.edu/group/corpora/smartbody-annot



(a) (b)

(c) (d)

Table 1: AP scores (on Y-axis) for annotation on TRECVID (a), SmartBody (b) and AP scores (on Y-axis) for retrieval of top-5 videos on TRECVID (c), SmartBody (d).

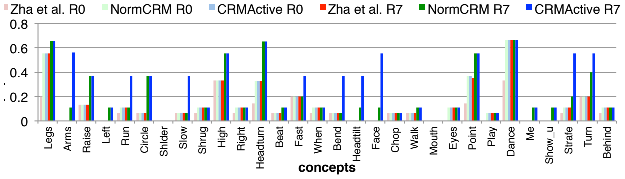


Figure 1: Precision scores for annotation of individual concepts of SmartBody for Round 0 (R0) and Round 7 (R7) of active learning.

allows them to capture the patterns from both these perspectives. CRMActive trains a more robust model early on by selecting the more informative samples; it results in its monotonic non-decreasing AP score for annotation/retrieval. This is in contrast with the occasional dips in the AP scores of the random baseline, which might potentially select some of the relatively noisy training samples early on. Figure 2 shows a sample annotation result on the SmartBody dataset using CRMActive. We see that the model gets all top 3 labels correct at Round 7, before the data is fully annotated.

Figure 1 shows the annotation performance of all the models for the individual concepts of the SmartBody dataset over two rounds (initial and towards the end). The concept scores for the NormCRM are obtained by averaging over the results of 3 runs. We notice a performance gain for all the models across most concepts over the two rounds, indicating that more training data helps. We also notice that CRMActive performs best on all concepts. All models do well on concepts with many positive examples (e.g., Legs) or complex concepts (e.g., Dance).

4. CONCLUSIONS

In this work, we proposed a sample selection algorithm based on active learning by combining a novel measure of sample uncertainty and a novel cluster-refinement approach for determining sample density and diversity. This approach is shown to outperform multiple baselines at both annotation and retrieval tasks. Our experiments also reveal the pros of using a generative approach of jointly modeling both the features and labels. CRMActive is thus shown to be a promising active learning approach to explore.

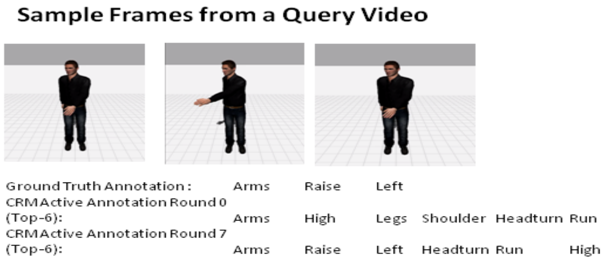

Sample Frames from a Query Video
Ground Truth Annotation: Arms Raise Left
CRMActive Annotation Round 0 (Top-6): Arms High Legs Shoulder Headturn Run
CRMActive Annotation Round 7 (Top-6): Arms Raise Left Headturn Run High

Figure 2: A sample annotation result on SmartBody dataset, showing the top-6 annotated labels by CRMActive after Round 0 and Round 7.

5. REFERENCES

[1] TRECVID 2007: TREC video retrieval evaluation. link: http://www-nlpir.nist.gov/projects/tv2007/tv2007.html.

[2] K. Brinker. Incorporating diversity in active learning with support vector machines. In *Proceedings of ICML*, volume 3, pages 59–66, 2003.

[3] C. K. Dagli et al. Leveraging active learning for relevance feedback using an information theoretic diversity measure. In *Image and Video Retrieval*. 2006.

[4] M. G. Genton. Classes of kernels for machine learning: a statistics perspective. *The Journal of Machine Learning Research*, 2:299–312, 2002.

[5] T. S. Huang et al. Active learning for interactive multimedia retrieval. *Proceedings of the IEEE*, 2008.

[6] T. Jebara, R. Kondor, and A. Howard. Probability product kernels. *The Journal of Machine Learning Research*, 5:819–844, 2004.

[7] V. Lavrenko, S. Feng, and R. Manmatha. Statistical models for automatic video annotation and retrieval. In *Proceedings of ICASSP*, volume 3, 2004.

[8] D. Pelleg, A. W. Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of ICML*, pages 727–734, 2000.

[9] G.-J. Qi et al. Video annotation by active learning and cluster tuning. In *CVPR Workshops*, 2006.

[10] J. Sedmidubsky and J. Valcik. Retrieving similar movements in motion capture data. In *Similarity Search and Applications*, pages 325–330. 2013.

[11] M. Thiebaux, S. Marsella, A. N. Marshall, and M. Kallmann. Smartbody: Behavior realization for embodied conversational agents. In *AAMAS*, 2008.

[12] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Ninth ACM MM*, 2001.

[13] Z.-J. Zha et al. Interactive video indexing with statistical active learning. *Multimedia, IEEE Transactions on*, 14(1), 2012.

ACKNOWLEDGEMENT: This material is based on work supported in part by the DreamWorks Animation SKG. Any opinions, findings, conclusions or recommendations expressed in this material are the authors' and do not reflect those of the sponsors. The project or effort described here has been sponsored in part by the U.S. Army. Statements and opinions expressed do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred. Additionally, the authors would also like to sincerely thank every member of the ICT natural language dialogue group, Dr. Ari Shapiro at ICT, and Prof. Louis-Philippe Morency at CMU for their insightful suggestions, feedback, and co-operation.