Query and Keyframe Representations for Ad-hoc Video Search

Foteini Markatopoulou
Information Technologies Institute (ITI), CERTH
Thermi, Greece, 57001
Queen Mary University of London
markatopoulou@iti.gr

Damianos Galanopoulos
Information Technologies Institute (ITI), CERTH
Thermi, Greece, 57001
dgalanop@iti.gr

Vasileios Mezaris
Information Technologies Institute (ITI), CERTH
Thermi, Greece, 57001
bmezaris@iti.gr

Ioannis Patras
Queen Mary University of London
Mile end Campus, UK, E14NS
i.patras@qmul.ac.uk

ABSTRACT

This paper presents a fully-automatic method that combines video concept detection and textual query analysis in order to solve the problem of ad-hoc video search. We present a set of NLP steps that cleverly analyse different parts of the query in order to convert it to related semantic concepts, we propose a new method for transforming concept-based keyframe and query representations into a common semantic embedding space, and we show that our proposed combination of concept-based representations with their corresponding semantic embeddings results to improved video search accuracy. Our experiments in the TRECVID AVS 2016 and the Video Search 2008 datasets show the effectiveness of the proposed method compared to other similar approaches.

CCS CONCEPTS

• **Information systems → Query representation**; **Video search**;

KEYWORDS

Video search; zero-shot learning; visual analysis

ACM Reference format:
Foteini Markatopoulou, Damianos Galanopoulos, Vasileios Mezaris, and Ioannis Patras. 2017. Query and Keyframe Representations for Ad-hoc Video Search. In *Proceedings of ICMR '17, Bucharest, Romania, June 06-09, 2017,* 5 pages.
https://doi.org/10.1145/3078971.3079041

1 INTRODUCTION

Ad-hoc video search (AVS) [1] is the problem of retrieving, from a large video collection, video fragments (e.g., video shots) that are related to a given query. A query refers to an ad-hoc textual description, e.g. "Find shots of a woman wearing glasses". This problem is closely related to the simpler problem of concept-based video search, where a set of video shots is retrieved given a specific keyword (a.k.a. concept). In the latter case supervised learning (e.g., deep convolutional neural networks (DCNNs)) can be used to annotate the video shots with concepts. However, AVS is more complicated because an input query could be any complex or also abstract textual description for which annotated data does not exist; as a result, unsupervised learning and natural language processing (NLP) need to be employed for generating a common representation of queries and videos.

In this work we present a fully-automatic AVS method that uses solely a natural-language textual query to retrieve related video shots from a video collection. The novelty of our method is: a) An efficient algorithm that performs a number of NLP and semantic analysis steps to translate a query into a set of predefined concepts. b) A new approach that projects the concept-based video shot and query representations into a common semantic embedding space. And c) the combination of two different measures for the distance between the video shots and the target query, calculated on the concept-based and the semantic embedding representations respectively. Our AVS method was evaluated on the TRECVID AVS 2016 [1] and Video Search 2008 [5] datasets. The results show that it outperforms other state-of-the-art approaches.

2 RELATED WORK

Fully-automatic AVS is a very challenging problem, where the complete video search is performed without any user intervention. Typically, the query is broken down to a set of concepts using NLP. Each video shot from the test video collection is annotated with the same set of concepts, e.g., using DCNNs, and a distance measure is applied in order to retrieve those video shots that are closer to the concept-based query representation [3, 4, 12, 13, 15, 17, 20, 23]. Building the concept-based query representation starts by using simple NLP rules, e.g., removing stop-words, extracting nouns, verbs etc. or simply space-separating the whole query, which results to a set of terms for the query. Then, the semantic relation between each of the terms and the concepts is calculated, and the most semantically similar concepts to these terms are selected. The novelty of [20] is that they also enrich each concept with additional information captured by Google or Wikipedia, while in [4] an inverted index structure is used in order to associate the query with the concepts. A semi-automatic system is presented in [21], where the user is asked to choose keywords given a test query. All the above methods

ICMR '17, June 06-09, 2017, Bucharest, Romania

ACM ISBN 978-1-4503-4701-3/17/06…$15.00
https://doi.org/10.1145/3078971.3079041

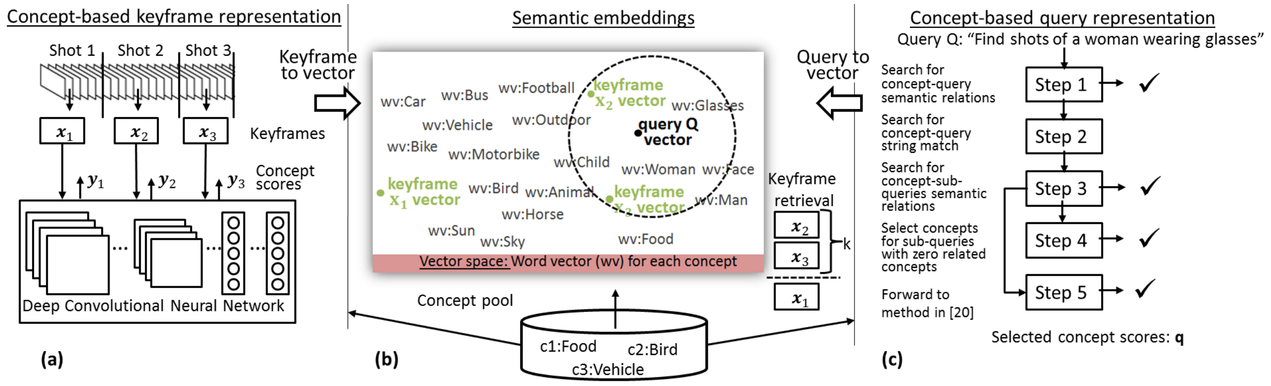

Figure 1: Overview of the proposed ad-hoc video search method.

treat the query as a set of simple terms. However, detecting the most useful parts of it, e.g., subsequences that contain the main content that the user asks for retrieval, could further improve the video search accuracy. Such a method is proposed in this paper. In contrast to the above methods, in [6], query models are trained with videos retrieved from websites, which is significantly slower compared to all the other methods discussed here.

Some recent methods for concept-based search, for example word2vec [10, 11, 14], train semantic embedding spaces of words or sentences from a large corpus using simple architectures of neural networks. After the embedding space is established, both video shots and concepts can be projected to it in order to directly measure their distance [15], [7], [19]. For example, in [15] images are mapped into a semantic embedding space by combining the class label embeddings with the concept detection results. All the methods in this paragraph aim to retrieve images for a single unknown concept label, which is a simpler problem compared to the one that we investigate, i.e., retrieving video shots given a complex textual query. In addition, combining the distances of the video shots from the target query calculated with respect to both concept-based and semantic embedding representations has not been investigated before.

3 THE PROPOSED METHOD

3.1 Overview

Assume that a text query Q and a set of keyframes $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ are given, where one keyframe $\mathbf{x}_i \in \mathbb{R}^d$ has been extracted from each shot of the videos in a collection. Our goal is to retrieve for query Q the k keyframes from $\mathbf{X}$ that are most closely related to it. The overview of our method is presented in Fig. 1. Given a predefined concept pool $C = \{c_j\}_{j=1}^{T}$, our method represents both the keyframes (Fig. 1 (a)) and the query (Fig. 1 (c)) as vectors of related concepts. Then, these concept-based representations are projected into a common semantic embedding space (Fig. 1 (b)). Finally, the k-nearest keyframe representations to the query representation are retrieved using a distance measure.

3.2 Concept-based Keyframe Representation

Our method initially applies a DCNN $D : \mathbb{R}^d \Rightarrow [0, 1]^T$, that has been trained on the T concepts, in every keyframe $\mathbf{x}_i$ in order to calculate concept-based representations $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{N}$. The DCNN's output for an input keyframe $\mathbf{x}$, $D(\mathbf{x}) = \mathbf{y}$, is a vector $\mathbf{y} \in [0, 1]^T$ that indicates the model's belief that each of the concepts in C appears in the input keyframe.

3.3 Concept-based Query Representation

A set of NLP steps is applied in order to translate the query in a set of related concepts chosen from the concept set C. Let $C^Q = \{c_1^Q, c_2^Q, ..., c_{T'}^Q\} \subseteq C$ be the set of concepts selected for the query Q, where $T' \leq T$, and $\mathbf{q} = [q_1, q_2, ..., q_{T'}] \in [0, 1]^{T'}$ a vector that indicates the degree to which each of the selected concepts in C^Q is related to Q. The following steps focus on analysing different parts of the query, instead of treating it as a set of single terms (words), which results to more distinctive retrieved concepts. Starting from the empty set $C^Q = \emptyset$ we calculate C^Q and $\mathbf{q}$ as follows:

Step one: The complete textual description of Q is compared with each concept in C for "semantic relatedness" in terms of the Explicit Semantic Analysis (ESA) measure (which returns a real number in the [0,1] interval) [8]. Those concepts that are semantically close to the query, i.e., the concepts that have ESA value higher than a threshold θ, are added in the set C^Q. If at least one concept is selected in this way, we assume that the entire query is very well described by these concepts and the query processing stops; otherwise, we continue with step two.

Step two: This step searches if any of the concepts in our concept pool appears in any part of the test query by string matching. Some (complex) concepts may describe part of the query quite well, but appear in a way that is difficult to detect them due to the subsequent linguistic analysis, e.g., breaking down the query to sub-queries. Any concept that appears in the query is added in the set C^Q and the query processing continues in step three.

Step three: Queries are complex sentences, but this step automatically transforms them into elementary *sub-queries*; i.e., meaningful smaller phrases or terms that are included in the original query. For example, the query "Find shots of one or more people at train station platform" is split into the following four sub-queries:

"people", "train station platform", "persons" and "train station". Then, each of the *sub-queries* is translated to a concept vector. To identify the sub-queries, part-of-speech tagging and stop-word removal are used together with a task-specific set of NLP rules. For example, extracting "Noun - Verb - Noun" sequences and considering them as sub-queries. The motivation is that such a triad is much more characteristic of the original query than any of the single terms alone. Then, the ESA measure is calculated between each sub-query and each of the concepts in the pool. Concepts that exceed the threshold θ are added in the set C^Q. In the case that for all of the sub-queries at least one concept has been selected, the query processing stops. If for a subset of the sub-queries no concepts have been selected then these sub-queries are propagated to step four. Finally, if for all the sub-queries no concepts have been selected then the test query and all of the sub-queries are propagated to step five.

Step four: For a subset of the sub-queries no concepts were selected. For each of these sub-queries, the concept with the highest value of ESA measure is selected in this step (i.e., threshold θ is ignored), and then the query processing stops.

Step five: For some queries, the processing step up to step three did not select any concepts. In this case, the query and the sub-queries are served as input to the zero-example event detection system of [20], which returns a ranked list of the most relevant concepts in accordance with relatedness scores, based on the ESA measure. In [20] the concepts are enriched with additional information captured by Google or Wikipedia, which virtually augments the concept pool. Then, the query processing has been completed.

Finally, the query's concept vector $\mathbf{q} \in [0,1]^{T'}$ is formed by the corresponding scores of the selected concepts. If a concept has been selected in steps 1, 3, 4 or 5 then the corresponding vector's element is assigned with the relatedness score (calculated using the ESA measure); if it has been selected in step 2, it is set equal to 1. A complete example of applying the above steps in a query is presented in Table 1.

3.4 Semantic Embeddings for Concept-based Query and Keyframe Representations

Given a semantic embedding space $S \subseteq \mathbb{R}^m$, we project both the concept-based keyframe (Section 3.2) and the query (Section 3.3) representations into S, in order to directly measure their distance. Initially, we calculate the set $S^C = \{s(c_1), s(c_2), ..., s(c_T)\}$ of the semantic embedding vectors $s(c_j) \in S$ associated with each concept in C, by applying a pre-trained word2vec model [14].

Then, similarly to [15], our method calculates a keyframe semantic embedding vector $f(\mathbf{x}) \in \mathbb{R}^m$, as the combination of the semantic embeddings of the R-top retrieved concepts for $\mathbf{x}$, according to the concept-based keyframe representation $\mathbf{y} \in \mathbf{Y}$, weighted by their corresponding concept detection scores:

$$f(\mathbf{x}; \mathbf{y}, S^C) = \frac{1}{Z} \sum_{r=1}^{R} y_{g(\mathbf{x},r)} \cdot s(c_{g(\mathbf{x},r)}), \tag{1}$$

where $g(\mathbf{x}, r)$ denotes the r-th most likely concept label for the input keyframe $\mathbf{x}$ according to $\mathbf{y}$, $Z = \sum_{r=1}^{R} y_{g(\mathbf{x},r)}$ the normalization term and R a parameter that considers the maximum number of embeddings that will be combined.

Table 1: Concept-based query representation example.

Query: ***Find shots of three people or more walking or bicycling on a bridge during daytime***			
	Sub-queries	C^Q ($\theta = 0.8$)	**q**
Step 1	*Find shots of...daytime*	$\emptyset$	-
Step 2	***three people or more*** *walking or bicycling on a bridge during daytime*	three or more people	1.0
Step 3	people walking	walking	1.0
	bicycling	bicycle-built-for-two	1.0
		bicycles	0.85
		bicycling	0.84
	bridge	suspension bridge	1.0
		bridges	0.84
	Sub-query *daytime* also found but without corresponding concepts with ESA distance > θ		
Step 4	daytime	daytime outdoor	0.74

Subsequently, we calculate the semantic embedding vector associated with the concept-based query representation by extending the above process as follows. Given the set of concepts C^Q assigned to this query and the corresponding ESA scores $\mathbf{q}$, described in Section 3.3, our method calculates the semantic embedding vector h(Q) for query Q, as the combination of the semantic embeddings of the concepts assigned to this query weighted by their corresponding ESA score:

$$h(Q; \mathbf{q}, S^C) = \frac{1}{Z'} \sum_{l=1}^{T'} q_l \cdot s(c_l^Q), \tag{2}$$

where $Z' = \sum_{l=1}^{T'} q_l$ the normalization term.

After the concept-based keyframe representations have been calculated (Section 3.2), our system measures their distance from the concept-based query representation (Section 3.3), e.g. by calculating the euclidean distance. Similarly, the distance between the semantic embedding keyframe representations and the semantic embedding query representation is calculated and the two distance vectors are combined in terms of arithmetic mean. The k keyframes with the smallest distance are then retrieved.

4 EXPERIMENTAL STUDY

4.1 Dataset and Experimental Setup

Our experiments were performed on the TRECVID AVS 2016 (AVS16) [1] and Video Search 2008 (VS08) [5] datasets that consist of approx. 600 and 100 hours of internet archive videos and are evaluated on 30 and 48 queries, respectively. Ground-truth annotated training data does not exist for these queries. The AVS problem as defined in TRECVID [1] was examined, i.e., given a query, the goal was to retrieve the 1000 video shots that are mostly related with it. We analyze our results in terms of mean extended inferred average precision (MXinfAP), which is an approximation of the mean average precision suitable for the partial ground-truth that accompanies the TRECVID dataset [22].

In order to create the concept-based keyframe representations, each keyframe was automatically annotated with 1000 ImageNet [18] and 346 TRECVID SIN [2] concepts. Regarding the 1000 ImageNet concepts, we applied five pre-trained ImageNet DCNNs on the keyframes and fused their outputs in terms of arithmetic mean to obtain a single score for each of the 1000 concepts. Regarding the 346 SIN concepts, we fine-tuned (FT) two pre-trained ImageNet DCNNs on the 346 concepts using the TRECVID AVS development dataset [1] and the extension strategy proposed in [16], where one extension layer with 4096 neurons was used. We used the last layer of each of these networks to train support vector machine classifiers (SVMs) for each concept. The keyframe score per concept was the average of the probabilities that the two SVM models returned. Each keyframe was finally represented by a 1345-element vector by simply concatenating the score vectors for the ImageNet and the TRECVID SIN concepts. The threshold θ for deciding whether to select a concept or not in our method (Section 3.3) was set to 0.8. The pre-trained Google News Corpus word2vec model [1] was used for calculating the semantic embeddings of the concept-based representations (Section 3.4). In our preliminary experiments, small fluctuations of the overall accuracy were observed for different values of parameter R (Eq. 1), consequently and based on these experiments we set it to 70. The euclidean distance was used for measuring the distance between the keyframe and query representations.

4.2 Experimental Results

Table 2: Experiments (MXInfAP (%)) on the AVS16 dataset to investigate the parameters of the proposed method.

	All	Excluding one step:				
Steps		step 1	step 2	step 3	step 4	step 5
(a) Concept-based representation (Sections 3.2 + 3.3)	5.94	5.92	5.74	3.96	5.95	4.53
(b) Semantic embeddings (Section 3.4)	3.77	3.86	2.98	3.22	3.75	2.80
(c) Combination	6.35	**6.51**	5.77	4.37	6.27	4.99

Table 3: MXInfAP (%) for different compared AVS methods.

Methods	**AVS16**		**VS08**	
(a) Literature methods				
Tzelepis et al. [20]	4.16		8.27	
Ueki et al. [21]	5.65		7.24	
Norouzi et al. [15]	3.14		7.30	
(b) Top-4 TRECVID finalists				
Top-1	Le et al. [4]	5.4	Tang et al.	6.7
Top-2	Markat. et al. [13]	5.1	Snoek al.	5.4
Top-3	Liang et al. [6]	4.0	Ngo et al.	4.2
Top-4	Zhangy et al. [23]	3.8	Mei et al.	4.1
Proposed	**6.35**		**9.11**	

Table 2 presents the results of some intermediate experiments that we performed in order to investigate the performance when: i) the transformation to the semantic embedding space is ignored (Table 2 (a)), ii) the final distance from the query is calculated solely in the semantic embedding space (Table 2 (b)), iii) the complete process is used, i.e., the final distance is the combination of the distances calculated in i) and ii) (Table 2 (c)). For each of the above cases, we also examine the usefulness of each of the steps presented in Section 3.3, i.e., each column shows the corresponding results when one of the steps is excluded. According to Table 2 we conclude as follows: Concept-based representations perform very well on the AVS problem, outperforming the semantic embedding representations. However, combining the two types of representations further improves our method, reaching a MXInfAP of 6.35 %. Almost all of the steps one to five of Section 3.3 contribute to the improved translation of the query into related concepts; excluding one step reduces the performance in most cases. One exception is observed w.r.t. step 1. However, excluding step 1 and evaluating w.r.t. the VS08 dataset slightly reduced the accuracy, which lead us to propose the use of all five steps. Furthermore, step 4 only marginally affects the performance; thus, sub-queries that do not present high semantic relatedness with any of the concepts could be ignored when for at least one sub-query one or more concepts have been selected. Similar conclusions were reached on the VS08 dataset.

Table 3 compares the proposed method with 11 different literature ones. The top part of the table refers to those methods that were re-implemented in order to be adapted for this problem and datasets, whereas in the lower part we introduce the results of the top-four finalists in the AVS16 and the VS08 tasks. Especially for comparing with [21] we used a modified version that does not require the user’s involvement. In this modified version, we automatically split the query into several keywords after removing the stop-words, and for each keyword the concept with the highest ESA value is selected. Overall, as we can see in Table 3, for both datasets our proposed method performs very well compared to the other methods. Specifically, it outperforms all the compared methods, achieving an MXinfAP of 6.35% and 9.11% for AVS16 and VS08, respectively.

5 CONCLUSIONS AND FUTURE WORK

In this work we presented a fully-automatic method that combines video concept detection and query analysis for ad-hoc video search. Extensive experiments reveal the usefulness of the proposed NLP steps for translating a textual query to related predefined concepts and the usefulness of combining different types of keyframe-query representations (e.g., concept-based representations, semantic embeddings). Our proposed method was compared with many state-of-the-art AVS systems and was shown to outperform all fully-automatic entries to the TRECVID AVS 2016 benchmarking activity. In our future work we will investigate the use of different types of semantic embeddings, e.g., [9], and also the influence of the different keyframe-query representations on different types of queries, i.e., in which cases concept-based representations outperform semantic embeddings or the combination of both.

ACKNOWLEDGMENTS

This work was supported by the EU’s Horizon 2020 research and innovation programme under grant agreements H2020-693092 MOVING and H2020-687786 InVID.

[1] https://s3.amazonaws.com/dl4j-distribution/GoogleNews-vectors-negative300.bin.gz

REFERENCES

[1] G. Awad, J. Fiscus, and M. Michel et al. 2016. TRECVID 2016: Evaluating Video Search, Video Event Detection, Localization, and Hyperlinking. In *TRECVID 2016 Workshop*. NIST, USA.

[2] G. Awad, C. G. M. Snoek, and A. F. Smeaton et al. 2016. TRECVid Semantic Indexing of Video: A 6-Year Retrospective. *ITE Transactions on Media Technology and Applications* 4, 3 (2016), 187–208.

[3] M. Elhoseiny, B. Saleh, and A. Elgammal. 2013. Write a Classifier: Zero Shot Learning Using Purely Textual Descriptions. In *Int. Conf. on Computer Vision*.

[4] D.-D. Le et al. 2016. NII-HITACHI-UIT at TRECVID 2016. In *TRECVID 2016 Workshop*. Gaithersburg, MD, USA.

[5] G. Awad et al. 2008. TRECVID 2008–Goals, Tasks, Data, Evaluation Mechanisms and Metrics. In *TRECVID 2008 Workshop*. NIST, USA.

[6] J. Liang et al. 2016. Informedia @ Trecvid 2016. In *TRECVID 2016 Workshop*. Gaithersburg, MD, USA.

[7] Z. Fu, T. Xiang, and E. Kodirov et al. 2015. Zero-Shot Object Recognition by Semantic Manifold Distance. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[8] E. Gabrilovich and S. Markovitch. 2007. Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis.. In *IJCAI*, Vol. 7. 1606–1611.

[9] A. Habibian, T. Mensink, and C. G.M. Snoek. 2014. VideoStory: A New Multimedia Embedding for Few-Example Recognition and Translation of Events. In *Proceedings of the 22nd ACM International Conference on Multimedia (MM '14)*. ACM, NY, USA, 17–26.

[10] T. Kenter, A. Borisov, and M. de Rijke. 2016. Siamese CBOW: Optimizing Word Embeddings for Sentence Representations. *CoRR* abs/1606.04640 (2016).

[11] R. Kiros, Y. Zhu, and R. Salakhutdinov et al. 2015. Skip-Thought Vectors. In *Advances in Neural Information Processing Systems*. Curran Associates, 3294–3302.

[12] Y.-J. Lu, H. Zhang, and M. de Boer et al. 2016. Event Detection with Zero Example: Select the Right and Suppress the Wrong Concepts. In *ACM Int. Conf. on Multimedia Retrieval (ICMR) (ICMR '16)*. ACM, NY, USA, 127–134.

[13] F. Markatopoulou and A. Moumtzidou et al. 2016. ITI-CERTH participation to TRECVID 2016. In *TRECVID 2016 Workshop*. Gaithersburg, MD, USA.

[14] T. Mikolov, I. Sutskever, and K. Chen et al. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *26th Int. Conf. on Neural Information Processing Systems (NIPS'13)*. Curran Associates, USA, 3111–3119.

[15] M. Norouzi, T. Mikolov, and S. Bengio et al. 2013. Zero-Shot Learning by Convex Combination of Semantic Embeddings. *CoRR* abs/1312.5650 (2013).

[16] N. Pittaras, F. Markatopoulou, and V. Mezaris et al. 2017. *Comparison of Fine-Tuning and Extension Strategies for Deep Convolutional Neural Networks*. Springer, Cham, 102–114.

[17] B. Romera-Paredes and P. H.S. Torr. 2015. An embarrassingly simple approach to zero-shot learning. *32nd Int. Conf. on Machine Learning (ICML)* (2015).

[18] O. Russakovsky, J. Deng, and H. Su et al. 2015. ImageNet Large Scale Visual Recognition Challenge. *Int. Journal of Computer Vision (IJCV)* 115, 3 (2015), 211–252.

[19] R. Socher, M. Ganjoo, and C. D Manning et al. 2013. Zero-Shot Learning Through Cross-Modal Transfer. In *Advances in Neural Information Processing Systems*. Curran Associates, 935–943.

[20] C. Tzelepis, D. Galanopoulos, and V. Mezaris et al. 2016. Learning to Detect Video Events from Zero or Very Few Video Examples. *Image Vision Computing* 53 (Sept. 2016), 35–44.

[21] K. Ueki, K. Kikuchi, and T. Kobayashi. 2016. Waseda at TRECVID 2016: Ad-hoc Video Search. In *TRECVID 2016 Workshop*. Gaithersburg, MD, USA.

[22] E. Yilmaz, E. Kanoulas, and J. A. Aslam. 2008. A Simple and Efficient Sampling Method for Estimating AP and NDCG. In *31st ACM SIGIR Int. Conf. on Research and Development in Information Retrieval*. ACM, USA, 603–610.

[23] H. Zhang, L. Pang, Y. Lu, and C. Ngo. 2016. VIREO@TRECVID 2016: Multimedia Event Detection, Ad-hoc Video Search, Video to Text Description. In *TRECVID 2016 Workshop*. Gaithersburg, MD, USA.