PicSOM Experiments in TRECVID 2012

Mats Sjöberg, Satoru Ishikawa, Markus Koskela, Jorma Laaksonen, Erkki Oja
Department of Information and Computer Science
Aalto University School of Science
P.O. Box 15400, FI-00076 Aalto, Finland
firstname.lastname@aalto.fi

Abstract

Our experiments in TRECVID 2012 include participation in semantic indexing, known-item search, and instance search.

In the semantic indexing task we implemented linear and non-linear SVM-based classifiers on six visual features extracted from the main keyframes and also additional frames from longer shots. We used homogeneous kernel map approximations for the linear classifiers, which narrow the performance gap to the non-linear SVMs. We submitted to the full task the following four runs:

- `PicSOM_1`: linear + non-linear classifiers, two-stage fusion
- `PicSOM_2`: linear + non-linear classifiers
- `PicSOM_3`: non-linear classifiers
- `PicSOM_4`: linear classifiers

The run `PicSOM_1` obtained the highest MXIAP score of 0.2263.

In the known-item search task we submitted four automatic runs:

- `PicSOM_1`: metadata only text search
- `PicSOM_2`: metadata + ASR (with aspell) + OCR text search
- `PicSOM_3`: metadata only text search + Google image search
- `PicSOM_4`: metadata + ASR (with aspell) + OCR text search + Google Image Search

Our automatic runs used text search with a single video-level index containing the title, subject, and description from the metadata. We also included text detected by OCR and provided by ASR with spell correction in some runs. Furthermore, we tried to incorporate image content cues by using images retrieved with Google Image Search, but this did not improve the results over our best run which was `PicSOM_2` with a MIR score of 0.235.

In the instance search task, we submitted four automatic runs:

- `PicSOM_1`: large vocabulary BoV (LVBoV)
- `PicSOM_2`: SOM-based content-based retrieval (CBIR)
- `PicSOM_3`: LVBoV + CBIR
- `PicSOM_4`: LVBoV + CBIR + pairwise matching of local descriptors

The fusion of LVBoV + CBIR resulted in better performance than either of the algorithms alone. Reranking based on pairwise matching of local descriptors further improved the results. Our best-scoring run was `PicSOM_4` with a MAP score of 0.100.

I. Introduction

In this notebook paper, we describe our experiments for the TRECVID 2012 evaluation [1]. We participated in the semantic indexing task (Section II), in the automatic known-item search task (Section III), and in the automatic instance search pilot task (Section IV).

II. Semantic indexing

Our system for the semantic indexing (SIN) task is based on fusing several supervised detectors trained for each concept, based on different shot-level image and video features. The basic system architecture is the same as we have used in previous editions of TRECVID [2], [3]. We continue our experiments of last year where we aim to speed up concept prediction by using linear classifiers. The accuracy of linear classifiers is improved by employing explicit kernel maps.

As the concept-wise ground-truth for the supervised detectors we used the annotations gathered by the organised collaborative annotation effort [4]. All our runs were submitted to the full task and are of type A.

A. Low-level features

In addition to the main keyframe provided in the master shot reference, we extracted additional frames from shots longer than two seconds, similarly as last year [3]. We extracted six image features from all extracted frames, four of them BoV-type (*SIFT*, *ColorSIFT*, *SIFTds*, and *ColorSIFTds*) and two others (*Centrist* and *ScalableColor*). See [5], [3] for details.

B. Linear and non-linear SVM classifiers

For the non-linear SVM classifiers we used an adaptation of the C-SVC implementation of LIBSVM [6], extended to support additional kernels. The exponential χ^2 kernel was used for the BoV features and the RBF kernel was used for *Centrist* and *ScalableColor*. The classifier parameters were optimized similarly as last year [3] using line and grid search and 10-fold cross-validation.

TABLE I
AN OVERVIEW OF THE SUBMITTED RUNS IN THE SEMANTIC INDEXING TASK. SEE TEXT FOR DETAILS.

#	run id	classifiers		MXIAP	
		linear	non-linear	full task	light task
1	PicSOM_1	•	•	0.2263	0.2602
2	PicSOM_2	•	•	0.2252	0.2590
3	PicSOM_3		•	0.2224	0.2555
4	PicSOM_4	•		0.1984	0.2292

For the linear BoV classifiers, we utilize the homogeneous kernel maps proposed by Vedaldi and Zisserman [7], [8] to approximate additive non-linear kernels, such as the intersection or the χ^2 kernel. Using a homogeneous kernel map, we can encode a d-dimensional feature vector as a $d(2n+1)$-dimensional vector and use a linear classifier with it to approximate the corresponding non-linear kernel. Similarly as in [7], [8], we have observed in our experiments that the homogeneous kernel map approximations reach the performances of the corresponding non-linear additive kernels. We apply kernel map approximation of the intersection kernel of order $n = 3$, using the implementation available in the VLFeat library [9]. All linear classifiers were trained using the LIBLINEAR [10] library.

C. Submitted runs

This section details our submitted semantic indexing runs. Table I shows an overview. The two columns in the middle refer to the used classifiers: linear and/or non-linear SVMs. All six features are used in all runs. The shot-wise probability estimates are obtained from all extracted keyframes as the maximum over the keyframe-wise probabilities. The two rightmost columns list the corresponding mean extended inferred average precision (MXIAP) [11] values, both for the full and light tasks. Figure 1 illustrates the concept-wise XIAP results of our submitted runs.

The runs `PicSOM_3` and `PicSOM_4` use only the non-linear SVMs and linear classifiers, respectively. The feature-wise classifiers are fused using arithmetic mean. In `PicSOM_2`, all classifiers are used in the fusion.

The run `PicSOM_1` uses a two-stage weighting scheme where the linear and non-linear classifiers are first fused separately using arithmetic mean and these are then fused together using geometric mean.

The MXIAP results in Table I show that, by using explicit kernel maps, linear classifiers are relatively competitive. Using exponential kernels, the fused results are about 11–12% higher than with the linear classifiers. This increase comes, however, with considerable computational costs in both training the classifiers and evaluating them. Using all available classifiers brings also a slight improvement. The two-stage fusion run (`PicSOM_1`) obtained our highest MXIAP score of 0.2263 in the full task.

III. AUTOMATIC KNOWN-ITEM SEARCH

In the known-item search task we submitted four automatic runs. Our baseline approach was the simple text search of the metadata documents by *Lucene*. We used the title, subject, and description from the metadata of each video as one concatenated "meta" field, the provided automatic speech recognition data as the "ASR" field, and output of optical character recognition data as the "OCR" field in the Lucene document model. During the search time, we used the concatenated query sentences and the key visual cues as the Lucene text queries. This year we also tried to improve the search results by incorporating visual similarity information from low-level features extracted in the SIN task and Google Image Search results for the key visual cues of each search task.

A. OCR

We used the *Tesseract* OCR engine to perform optical character recognition on all keyframes (main keyframe + additional frames) in the test set. The results are very noisy, and Tesseract often does not succeed to recognize even relatively clear text, possibly due to the low quality and resolution of the keyframes. Still, in our tests with the TRECVID 2010 and TRECVID 2011 data set, including the OCR gave a small improvement in retrieval performance overall.

B. Text search

For text search we used the Lucene search engine based on the meta field (title, subject, description), the ASR field, and the OCR field data from the videos. We processed both the video-wise textual documents and the query texts with spelling correction. The spelling correction was performed using *GNU Aspell* and adding the first suggestion made by Aspell for misspelled words to the corresponding document fields and query texts. Then, we created Lucene search indices with the *EnglishAnalyzer* for all possible combinations of the video data fields (i.e. meta+ASR+OCR) with and without spelling corrections.

For choosing the two best combinations of the index data fields and the query text, we run experiments with the TRECVID 2011 data set and its KIS queries. As the baseline, we chose to use only the metadata as the input. For it, it turned out that spelling correction was not beneficial and the best performance was obtained by using the concatenated query and key visual cue, also without spelling corrections. When the input data was chosen freely among the available choices, the best combination found was to use the concatenation of the metadata without spelling correction, the spelling corrected automatic speech recognition, and the non-corrected OCR result.

C. Google Image Search

In order to incorporate also visual search cues, we used the Google Image Search system through the Google Custom Search API to find images that match visually with the key visual cues of the search queries. For each key visual cue, such as *geysers* or *bus* or *flags* we retrieved 30 images. On the average, each of the known item search tasks contained 3.7 key visual cues, resulting to the average number of 111 visual examples being used for each search task.

TABLE II
AN OVERVIEW OF THE SUBMITTED RUNS IN THE KNOWN-ITEM SEARCH TASK. SEE TEXT FOR DETAILS.

#	run id	meta	OCR	ASR	Google images	MIR
1	PicSOM_1	•				0.230
2	PicSOM_2	•	•	•		0.235
3	PicSOM_3	•			•	0.215
4	PicSOM_4	•	•	•	•	0.191

We extracted the *Centrist*, *ScalableColor*, *SIFT*, *ColorSIFT*, *SIFTds*, and *ColorSIFTds* features from the example images and placed them in the same Self-Organizing Map (SOM) based image indices [12] with the corresponding features extracted from the keyframe images. The SOM tends to map mutually similar feature vectors in the same or nearby map units. Based on that, we were able to calculate for each video keyframe a matching score that was the larger the closer that keyframe's feature vectors were mapped to the feature vectors of the retrieved example images on the SOM lattice.

The visual matching score for each video was then obtained as the maximum of all the scores for the keyframes extracted from its shots. These video-wise matching scores were then weighted with a fixed coefficient prior to summing the value with the corresponding Lucene scores to form the final score values. The optimal value for the weighting coefficient was chosen based on experiments with TRECVID 2011 KIS data.

D. Submitted runs

Our submitted runs in the known-item search task are summarized in Table II together with their mean inverse rank scores (MIR). PicSOM_1 and PicSOM_2 are of training type A and PicSOM_3 and PicSOM_4 are of training type D as they are using Google Image Search results as additional information.

PicSOM_1 and PicSOM_3 runs use a basic Lucene search on the meta field, and PicSOM_2 and PicSOM_4 runs use a basic Lucene search on the meta, ASR, and OCR fields. For all submitted runs where ASR text was used, it was augmented with spelling correction suggestions from GNU Aspell.

In PicSOM_3 and PicSOM_4, the results of the simple metadata text searches, PicSOM_1 and PicSOM_2, and visual matching between video keyframes and Google Image Search were combined. As can be seen, the use of Google Image Search examples unfortunately reduces the performance as compared to the pure text-based search even though we indeed obtained some advantage with it in experiments with the TRECVID 2011 data. So we obtained our highest MIR result 0.235 with metadata only text search (PicSOM_2).

IV. AUTOMATIC INSTANCE SEARCH

Our submissions to the automatic instance search task were based on the common large vocabulary BoV approach [13], [14]. In addition, we used our SOM-based content-based image retrieval algorithm [12] as a baseline, and experimented with reranking based on pairwise matching of local descriptors [15].

The test clips were sampled one frame per second, which resulted in a database of 685k frames. All the used algorithms operate on the frame level, and the final clip-level results are obtained using the maximum over the frame-wise results.

A. Large vocabulary BoV

A codebook of 1 million SIFT features was generated using hierarchical k-means clustering using a branching factor of 100.

An approximative kd-tree index was used to find the nearest feature when generating the SIFT visual word histograms for each image.

The visual words of the query's images are then mapped to images in the database that contain the same visual words by using an inverted file index. Given a query q, with query images $j = 1 \dots J$, the matching score $s_{i,q}$ for a database image i to the query is then calculated as

$$s_{i,q} = \max_j \sum_{w \in W_i \cap W_j} \text{idf}(w)\ \text{tf}(N_w^i)\ \text{tf}(N_w^j), \qquad (1)$$

where W_i is the set of visual words in the image i, and N_w^i is the number of times the visual word w occurs in image i. In our submitted runs we used

$$\text{idf}(w) = \log\left(\frac{I_{\text{all}}}{I_w}\right), \qquad (2)$$

where I_{all} is the total number of database images, and I_w is the number of database images in which w occurs, and

$$\text{tf}(x) = 1 + \log(x). \qquad (3)$$

In later experiments we also tried other weighting schemes: $\text{idf}(w) = I_{\text{all}}/I_w$, $\text{idf}(w) = 1$, and $\text{tf}(x) = 1$, $\text{tf}(x) = x$.

B. CBIR baseline

We used our SOM-based content-based image retrieval algorithm [12] to form a baseline for the instance search task. The algorithm has been used in previous TRECVIDs for various purposes, including automatic search (e.g. [16]). Tree-structured SOMs with 512×512 units on the bottom-most layer were used for three features: *ScalableColor*, *Centrist*, and *ColorSIFTds*. The topic-wise example images (without masks) were used as positive examples in each query.

C. Pairwise matching

The results of the algorithms described above were reranked using pairwise matching of local descriptors [15]. The matching is performed using randomized kd-trees and verified by estimating a homography between the point correspondences using RANSAC. On this stage, we use the topic-wise example images with the corresponding masks, and match 3000 top results with the masked example images. The found matching frames are then raised to the top of the results list.

TABLE III
AN OVERVIEW OF THE SUBMITTED AND SOME ADDITIONAL RUNS IN THE INSTANCE SEARCH TASK. SEE TEXT FOR DETAILS.

#	run id / add. info	LVBoV	CBIR	matching	MAP
1	PicSOM_1	•			0.074
2	PicSOM_2		•		0.036
3	PicSOM_3	•	•		0.086
4	PicSOM_4	•	•	•	0.100
	$\mathrm{tf}(x) = 1$	•			0.091
		•	•		0.098

D. Submitted and additional runs

This section provides details of our submitted runs, and some additional runs performed after the official evaluation. Table III shows an overview, where the third to fifth columns refer to which ones of the three available methods (large vocabulary BoV (LVBoV), SOM-based content-based image retrieval (CBIR), and pairwise matching of local descriptors) are used for this particular run. The last column lists the corresponding MAP scores. Figure 2 shows the topic-wise results of the runs.

On all runs, the relevance scores are first calculated on the frame level and the video-level scores are obtained by taking the maximum over all corresponding frame-level scores.

The runs PicSOM_1 and PicSOM_2 use only LVBoV and CBIR, respectively. PicSOM_3 fuses the results of both algorithms using weighted arithmetic mean on the frame level. As CBIR scores we use linear rank-based scoring. The weights are estimated using the instance search topics of 2011. The fusion of the two algorithms improved the results, even though the performance of CBIR was clearly lower than with LVBoV.

In PicSOM_4, the results of PicSOM_3 are reranked using pairwise matching of local features. The matching algorithm was able to find positive matches on six topics, but resulted in notable improvement only on two topics (Figure 2). PicSOM_4 obtained our highest MAP score of 0.100.

After the official evaluations we tried different variants for the $\mathrm{idf}(x)$ and $\mathrm{tf}(x)$ functions in Eq. (1) as explained in Section IV-A. The best result was achieved with $\mathrm{idf}(x)$ as in Eq. (2) and $\mathrm{tf}(x) = 1$. This result together with the fusion with the CBIR result, are shown in the last two rows in Table III.

ACKNOWLEDGMENTS

This work has been funded by the grants 255745 and 251170 of the Academy of Finland, TFMC 11863 of EIT ICT Labs, and *Next Media* and *D2I* SHOK projects. The calculations were performed using computer resources within the Aalto University School of Science "Science-IT" project.

REFERENCES

[1] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Greg Sanders, Barbara Shaw, Wessel Kraaij, Alan F. Smeaton, and Georges Queenot. Trecvid 2012 – an overview of the goals, tasks, data, evaluation mechani sms and metrics. In *Proceedings of TRECVID 2012*. NIST, USA, 2012.

[2] Mats Sjöberg, Markus Koskela, Milen Chechev, and Jorma Laaksonen. PicSOM experiments in TRECVID 2010. In *Proceedings of the TRECVID 2010 Workshop*, Gaithersburg, MD, USA, November 2010.

[3] Mats Sjöberg, Satoru Ishikawa, Markus Koskela, Jorma Laaksonen, and Erkki Oja. PicSOM experiments in TRECVID 2011. In *Proceedings of the TRECVID 2011 Workshop*, Gaithersburg, MD, USA, December 2011.

[4] Stéphane Ayache and Georges Quénot. Video corpus annotation using active learning. In *Proceedings of 30th European Conference on Information Retrieval (ECIR'08)*, pages 187–198, Glasgow, UK, March-April 2008.

[5] Markus Koskela, Mats Sjöberg, Ville Viitaniemi, Jorma Laaksonen, and Philip Prentis. PicSOM experiments in TRECVID 2007. In *Proceedings of the TRECVID 2007 Workshop*, Gaithersburg, MD, USA, November 2007.

[6] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[7] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2010)*, 2010.

[8] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):480 –492, march 2012.

[9] A. Vedaldi and B. Fulkerson. VLFeat: A library of computer vision algorithms. *http://www.vlfeat.org/*.

[10] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, and C.J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[11] Emine Yilmaz, Evangelos Kanoulas, and Javed A. Aslam. A simple and efficient sampling method for estimating AP and NDCG. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '08)*, pages 603–610, 2008.

[12] Jorma Laaksonen, Markus Koskela, and Erkki Oja. PicSOM—Self-organizing image retrieval with MPEG-7 content descriptions. *IEEE Transactions on Neural Networks*, 13(4):841–853, July 2002.

[13] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. of ICCV'03*, volume 2, pages 1470–1477, October 2003.

[14] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *Proceedings of IEEE CVPR 2006*, volume 2, pages 2161–2168, 2006.

[15] Xi Chen and Markus Koskela. Mobile visual search from dynamic image databases. In *Proceedings of Scandinavian Conference on Image Analysis (SCIA 2011)*, Ystad, Sweden, May 2011.

[16] Mats Sjöberg, Ville Viitaniemi, Markus Koskela, and Jorma Laaksonen. PicSOM experiments in TRECVID 2009. In *Proceedings of the TRECVID 2009 Workshop*, Gaithersburg, MD, USA, November 2009.

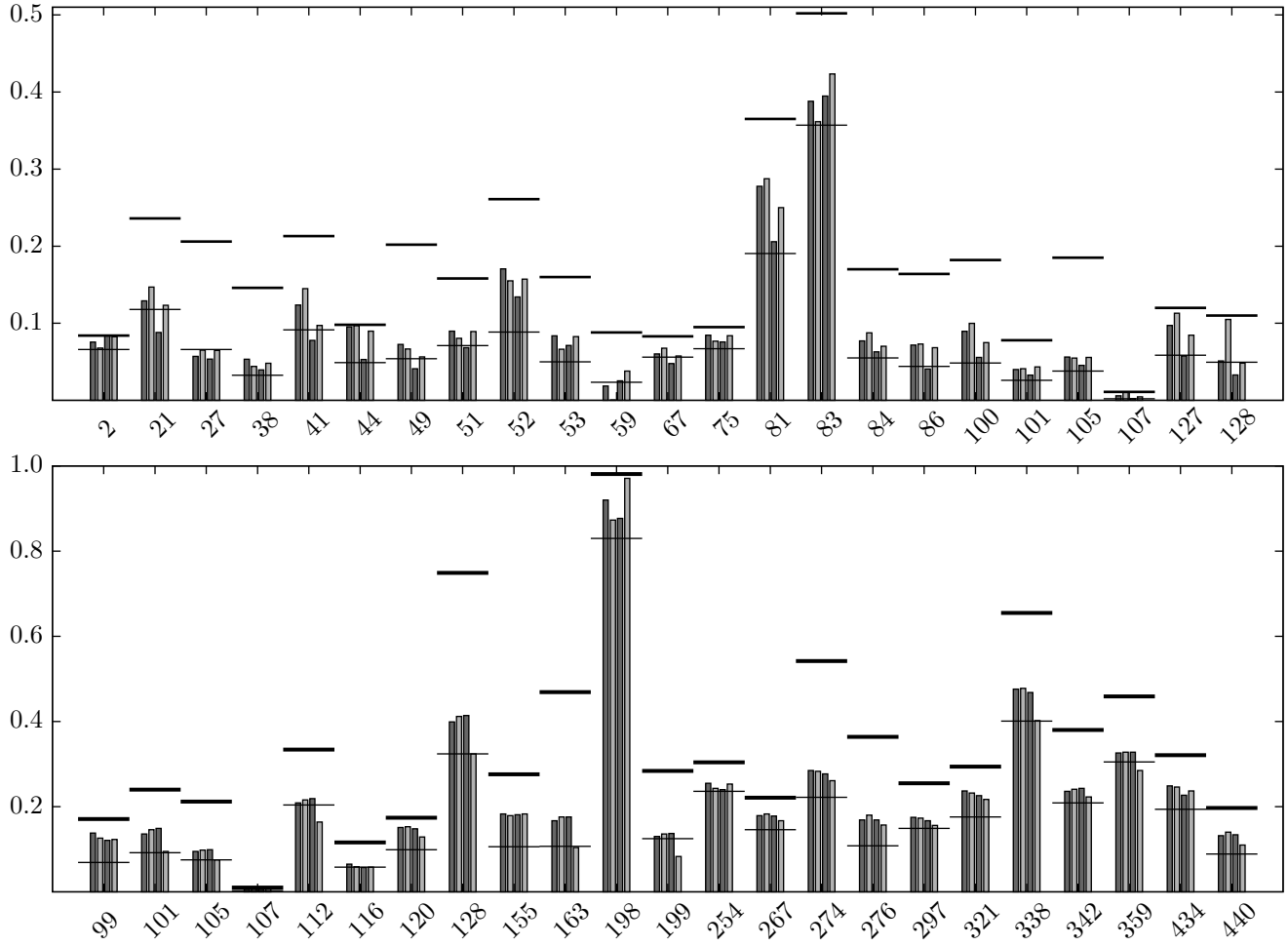

Fig. 1. The concept-wise XIAP results of our submitted runs for each evaluated concept in the full semantic indexing task. The order of the runs is as in Table I (i.e. the leftmost bar corresponds to `PicSOM_1`, etc.). The median and maximum values over all type-A submissions are illustrated as horizontal lines.

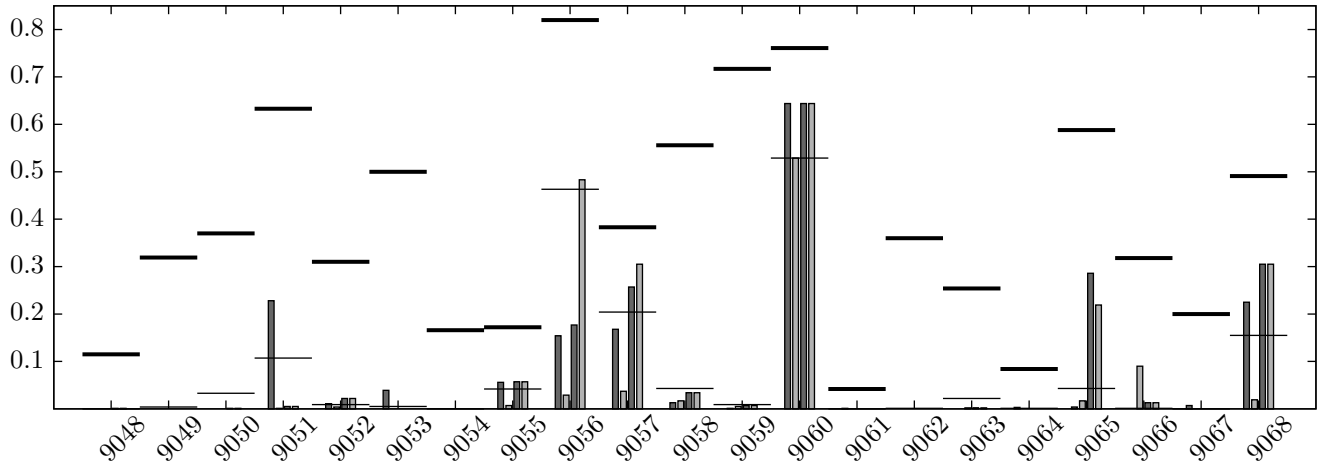

Fig. 2. The topic-wise AP results of our submitted runs for each evaluated topic in the instance search task. The order of the runs is as in Table III (i.e. the leftmost bar corresponds to `PicSOM_1`, etc.). The median and maximum values over all fully automatic submissions are illustrated as horizontal lines.