EURECOM at TRECVid AVS 2019

Danny Francis[†], Phuong Anh Nguyen[‡], Benoit Huet[†], Chong-Wah Ngo[‡]
[†]EURECOM, Sophia-Antipolis, France [‡]City University of Hong-Kong, Kowloon, Hong Kong
francis@eurecom.fr, panguyen2-c@my.cityu.edu.hk, huet@eurecom.fr, cscwngo@cityu.edu.hk

Abstract

This notebook reports the model and results of the EURECOM runs at TRECVID AVS 2019.

1. Introduction

In our runs of TRECVid AVS 2019, we propose using a fusion of two multimodal modules trained on different datasets. Our runs are based on the work we introduced in [4].

The remaining sections are organized as follows. Section 2 presents related works in AVS. Section 4 introduces the cross-modal learning employed for training two different modules, Section 4 describes the followed fusion method, and Section 5 reports our results at TRECVid AVS 2019 [2].

2. Related Works

From AVS 2018, the general approaches from the participants can be summarized as follows: linguistic analysis for query understanding combining different techniques for concept selection and fusion; or learning joint embedding space of textual queries and images; or the integration of two mentioned approaches. From the results of ten participants, we conclude that the approach of learning the embedding space is the key of success for AVS task. Following up this direction, we propose to learn two embedding spaces including objects counting and semantic concepts separately, and a fusion method to incorporate these models.

3. Cross-Modal Learning

In this section we will describe the multimodal models we employed. More precisely we will first define their architecture and then how we trained them.

3.1. Feature Representation

Let Q be a textual query and V an image or a video. We want to build a model so that Q and V can be compared. More precisely, we want to be able to assign a score to any (Q, V) to describe the relevance of V with respect to Q. For that purpose, we use a similar model to [3].

For processing textual queries, we represent any query Q of length L as a sequence $(w_1, ..., w_L)$ of one-hot vectors of dimension N, where N is the size of our vocabulary. These one-hot vectors are then embedded in a vector space of dimension D. More formally, we obtain a sequence of word embeddings $(x_1, ..., x_L)$ where $x_k = w_k W_e$ for each k in $\{1, ..., L\}$. The weights of the embedding matrix $W_e \in \mathbb{R}^{D \times N}$ are trainable.

The obtained sequence of word embeddings is then processed by a GRU, whose last hidden state $h_L = \text{GRU}(h_{L-1}, x_L)$ is kept and input to a Fully-Connected layer to get a sentence embedding v_s.

Regarding visual objects, the generic process we employ is to extract a vector representation $\varphi(V)$ of a visual object V where φ corresponds to any relevant concepts or features extractor. Then, we input $\varphi(V)$ to a Fully-Connected layer to obtain a visual embedding v_v.

Our goal is to train these models to be able to compare v_s and v_v. We will explain how these models are trained in Section 3.2.

3.2. Model Training

The objective is to learn a mapping such that the relevancy of a pair of a query and a video (Q, V) can be evaluated. As explained in Section 3.1, our model derives a query representation v_s from Q and a video representation v_v from V. Triplet loss is used as the loss function for model training. Mathematically, if we consider a query representation v_s, a positive video representation v_v (corresponding to v_s) and a negative video representation $\bar{v}_v$ (that does not correspond to v_s), the triplet loss $\mathcal{L}$ for $(v_s, v_v, \bar{v}_v)$ to minimize is defined as follows:

$$\mathcal{L}(v_s, v_v, \bar{v}_v) = \max(0, \alpha - \cos(v_s, v_v) + \cos(v_s, \bar{v}_v)) \quad (1)$$

where α is a margin hyperparameter that we set to 0.2. We chose to employ the hard-margin loss presented in [3], where $\bar{v}_v$ is chosen to be the representation of the negative

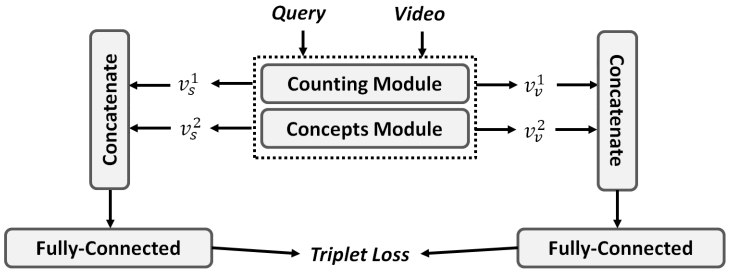

Figure 1. Proposed model derived from [4]. We extract embeddings from two modules: a counting module and a concepts module. These embeddings are then concatenated and input to Fully-Connected layers to obtain new embeddings. That model is also trained using a triplet loss.

video with the highest similarity with the query representation v_s among all videos in the current training mini-batch.

4. Fusion Strategy

In this section we will describe the two multimodal modules we used and how we fused them.

4.1. Multimodal Modules

Our model relies on two multimodal modules: a counting module and a concepts module (see Figure 1). Each of them has the architecture we described in Section 3.1 and has been trained according to the optimization scheme we defined in Section 3.2.

The counting module is based on a Faster-RCNN [10] trained on the OpenImagesv4 dataset [7]. It takes images as inputs. For each input, it detects objects belonging to the 600 classes of OpenImagesv4 and counts them to obtain a vector of dimension 600, where the value at index i corresponds to the number of detected objects of class i. Embeddings are then derived from that vector.

The concepts module takes as input concepts detections coming from four different concept detectors. These concept detectors are ResNet [5] models trained on ImageNet1k, Places-365 [16], TRECVID SIN [15] and HAVIC [12]. Following the same process as for other two modules, we generate embeddings from the concatenation of the concept detections coming from these four detectors.

4.2. Fusion Model

Instead of simply averaging similarity scores to compare videos and queries, we chose to train a model to draw finer similarities between them. For that purpose, we derived embeddings from our modules for videos and queries, and passed them through Fully-Connected layers to obtain new embeddings. More formally, if v_v^1 and v_v^2 are video embeddings respectively generated by the counting module and the concepts module, we derived the new video embedding v_v by inputting the concatenation of v_v^1 and v_v^2 to a fully-connected layer. We obtained the new sentence embedding v_s similarly, based on v_s^1 and v_s^2 (sentence embeddings generated by the counting and the concepts modules, respectively).

We trained our fusion models using the same triplet loss as we did for multimodal modules, as decribed in Section 3.2.

5. Results of runs

In this section, we report the results we obtained at TRECVid 2019.

5.1. Datasets

We trained our models based on the MSCOCO [9] dataset the TGIF [8] dataset and the train and test splits of the MSR-VTT [14] dataset. Validation has been performed on the validation split of MSR-VTT.

5.2. Implementation details

We implemented our models using the Tensorflow [1] framework for Python. Each of them has been trained for 150k iterations with mini-batches of size 64. We used the RMSProp [13] algorithm, with gradients capped to values between -5 and 5 and a learning rate of 10^{-4}. Hidden dimensions of GRUs are always 1024, and embeddings output by multimodal modules and fusion models are of dimension 512. The size of vocabularies has been set to 20k. We applied dropout [11] with rate 0.3 to all outputs of Fully-Connected layers, and batch normalization [6] to the inputs of our models. In triplet losses, the α parameter has been set to 0.2.

MSR-VTT videos have been processed as follows: we extracted uniformly one frame every fifteen frames, applied the extractor on each frame (Faster-RCNN for the counting module or concepts extractors for the concepts module) and averaged obtained vectors.

5.3. Results of Runs

The runs we submitted were the following:

Run	MAP
Run 1	0.014
Run 2	0.014
Run 3	0.020

Table 1. Results of our runs

- Run 1: Fusion of Concepts and Counting modules;
- Run 2: Concepts module alone;
- Run 3: If Q is a query, V a video, $S_1(Q,V)$ the score of the pair (Q,V) computed in run 1 and $S_2(Q,V)$ the score in run 2, the score in run 3 is $S_1(Q,V) + S_2(Q,V)$.

The scores we obtained with these three runs are reported in Table 1.

Results of all automatic runs are reported in Figure 2. Detailed results of Run 1, Run 2 and Run 3 are reported in Figure 3, Figure 4 and Figure 5, respectively.

6. Conclusion

EURECOM runs performed badly with respect to other runs. However, results got better when ensembling run 1 and run 2 into run 3. For future work, we think we should investigate how other methods than multimodal embeddings perform. Moreover, we think that a finer sentence processing method than using a single GRU should be found, for instance putting emphasis on visual concepts.

Acknowledgments

This work was supported by ANR (the French National Research Agency) via the ANTRACT project, the European H2020 research and innovation programme via the project MeMAD (Reference Np.: GA780069), a grant from the Research Grants Council of the Hong Kong SAR, China (Reference No.: CityU 11250716), and a grant from the PROCORE-France/Hong Kong Joint Research Scheme sponsored by the Research Grants Council of Hong Kong and the Consulate General of France in Hong Kong (Reference No.: F-CityU104/17).

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[2] G. Awad, A. Butt, K. Curtis, Y. Lee, J. Fiscus, A. Godil, A. Delgado, A. F. Smeaton, Y. Graham, W. Kraaij, and G. Quénot. Trecvid 2019: An evaluation campaign to benchmark video activity detection, video captioning and matching, and video search & retrieval. In *Proceedings of TRECVID 2019*. NIST, USA, 2019.

[3] F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler. Vse++: Improving visual-semantic embeddings with hard negatives.

[4] D. Francis, P. A. Nguyen, B. Huet, and C.-W. Ngo. Fusion of multimodal embeddings for ad-hoc video search. In *ViRaL 2019, 1st International Workshop on Video Retrieval Methods and Their Limits, co-located with ICCV 2019, 28 October 2019, Seoul, Korea*, 10 2019.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[7] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018.

[8] Y. Li, Y. Song, L. Cao, J. Tetreault, L. Goldberg, A. Jaimes, and J. Luo. Tgif: A new dataset and benchmark on animated gif description. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4641–4650. IEEE, 2016.

[9] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[10] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[11] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[12] S. M. Strassel, A. Morris, J. G. Fiscus, C. Caruso, H. Lee, P. Over, J. Fiumara, B. Shaw, B. Antonishek, and M. Michel. Creating havic: Heterogeneous audio visual internet collection. Citeseer.

[13] T. Tieleman and G. Hinton. Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[14] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[15] W. Zhang, H. Zhang, T. Yao, Y. Lu, J. Chen, and C. Ngo. Vireo@ trecvid 2014: instance search and semantic indexing. In *NIST TRECVID Workshop*, 2014.

[16] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

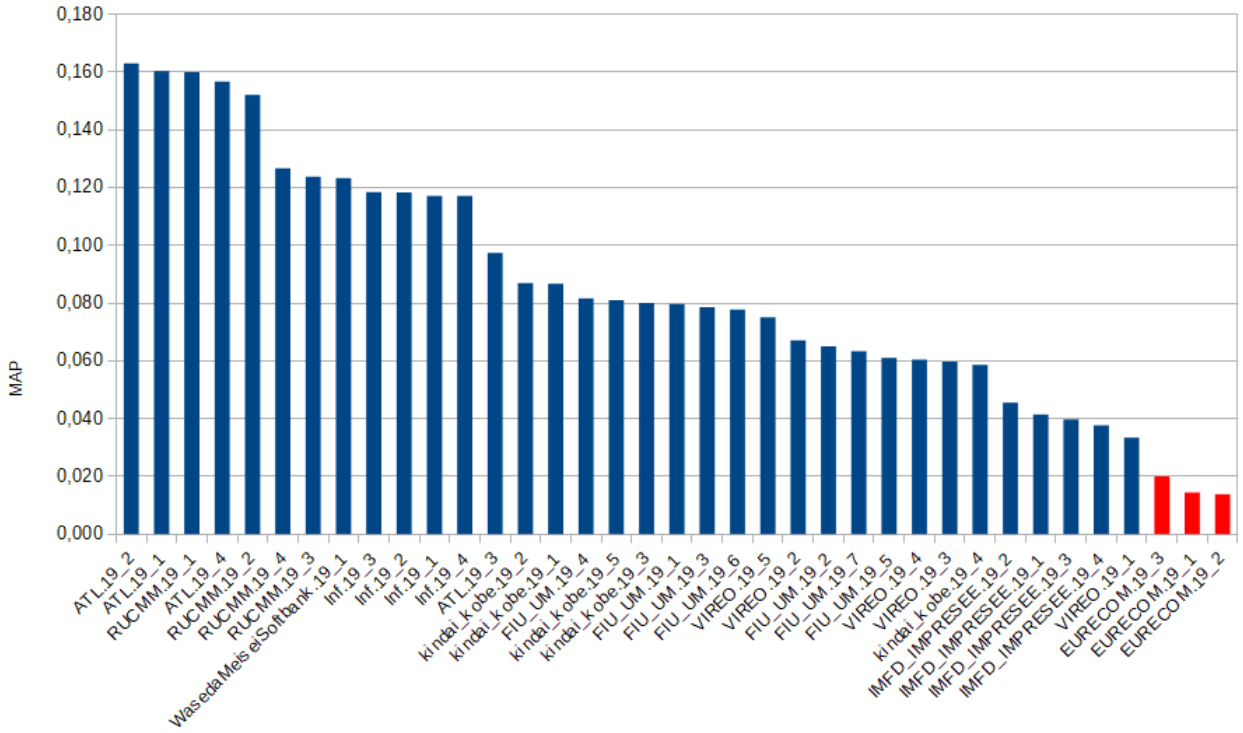

Figure 2. AVS Results (Fully Automated runs only)

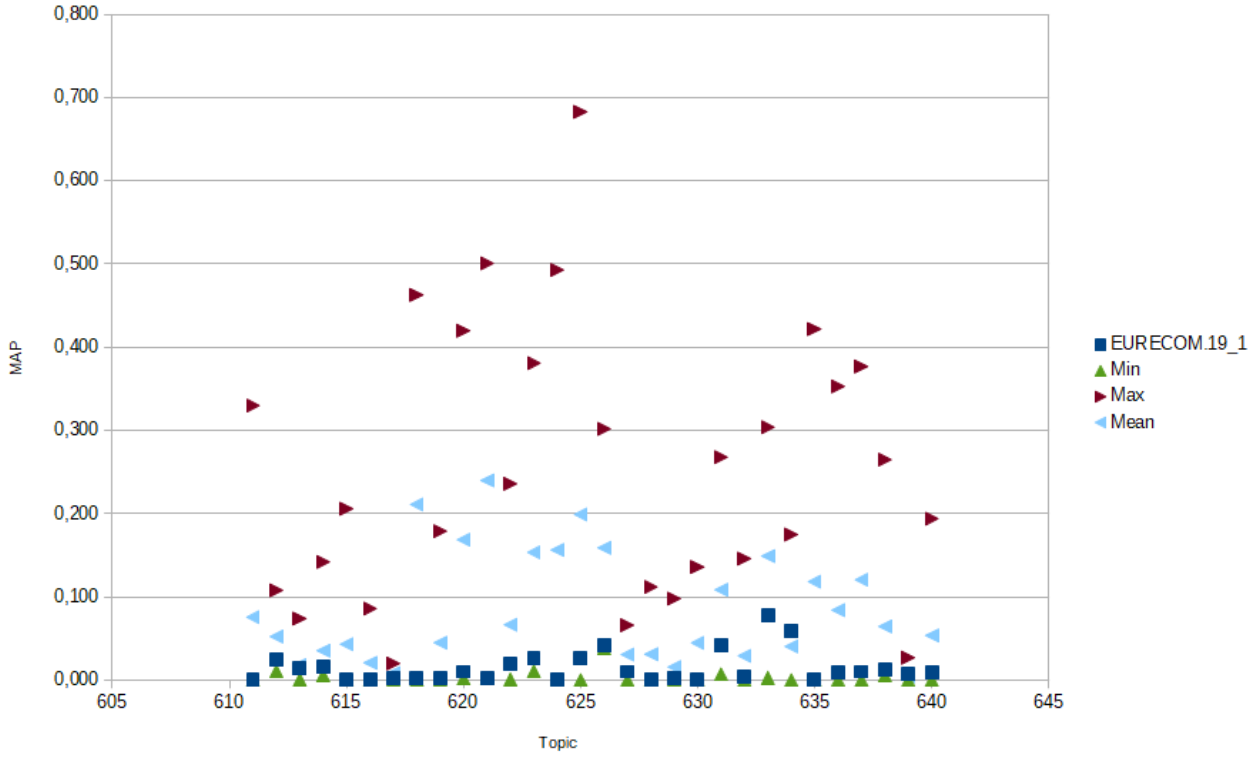

Figure 3. Detailed results of EURECOM run 1

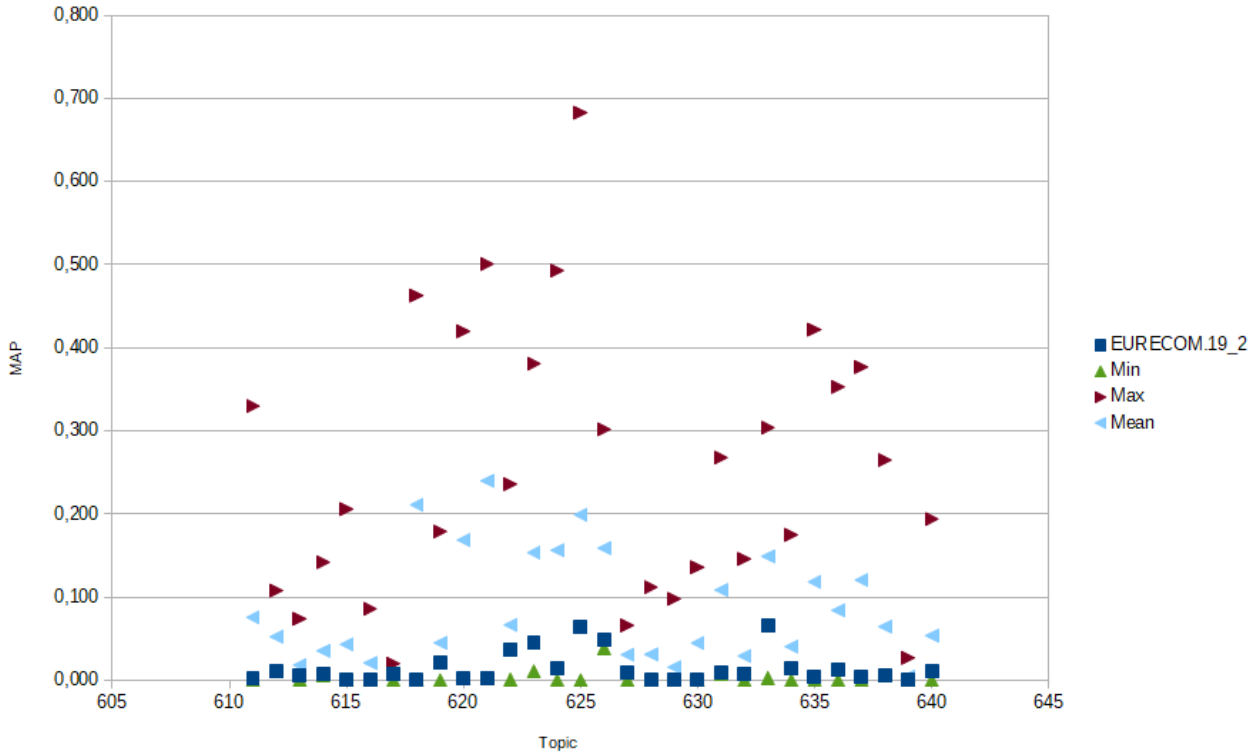

Figure 4. Detailed results of EURECOM run 2

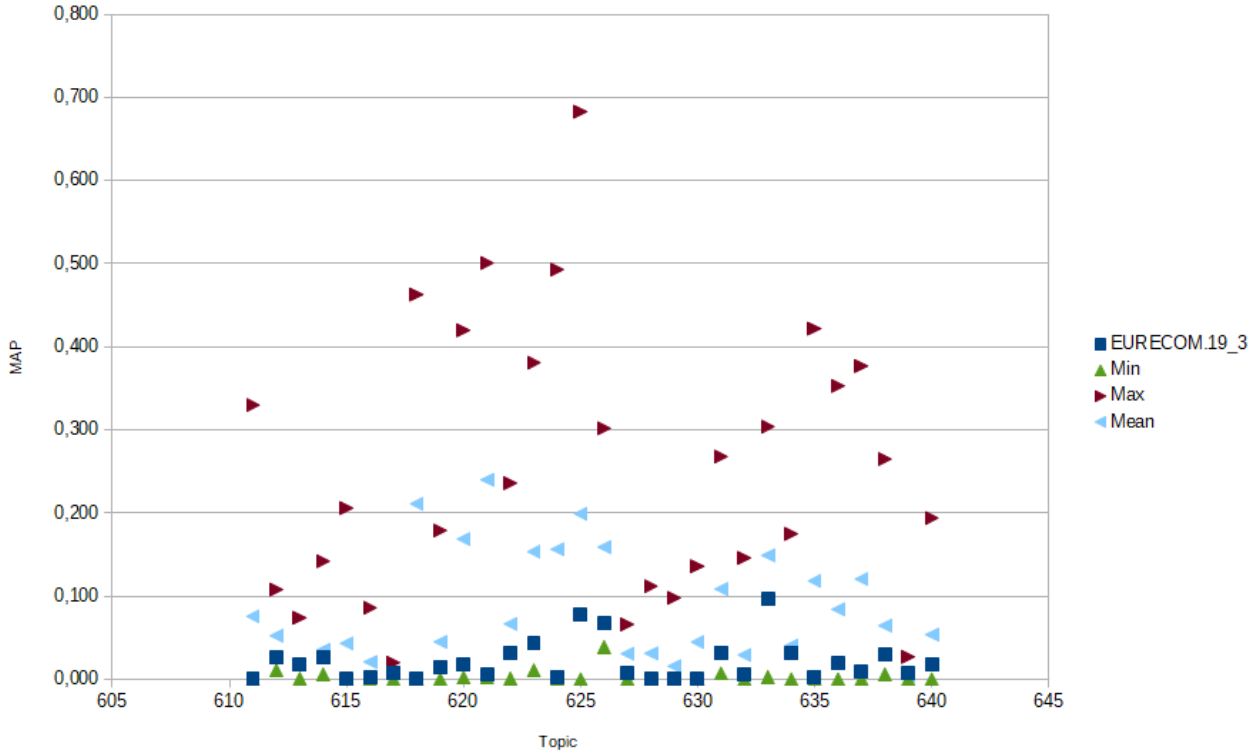

Figure 5. Detailed results of EURECOM run 3