National Institute of Informatics, Japan at TRECVID 2012

Duy-Dinh Le [1], Cai-Zhi Zhu [1], Sebastien Poullot [1], Vu Q. Lam [2]
Vu H. Nguyen [3], Nhan C. Duong [2], Thanh D. Ngo [1]
Duc A. Duong [3], Shin'ichi Satoh [1]

[1] *National Institute of Informatics*
2-1-2 Hitotsubashi, Chiyoda-ku, Japan 101-8430

[2] *Faculty of Information Technology*
University of Science, VNU
227 Nguyen Van Cu, Dst 5, Ho Chi Minh City, Vietnam

[3] *University of Information Technology, VNU*
KM 20 Hanoi Highway, Thu Duc Dist, Ho Chi Minh City, Vietnam

{ledduy,cai-zhizhu,poullot.sebastien,satoh,ndthanh}@nii.ac.jp,
{lqvu,dcnhan}@fit.hcmus.edu.vn
{ducda,vunh}@uit.edu.vn

***Abstract*—This paper reports our experiments for four TRECVID 2012 tasks: instance search, semantic indexing, multimedia event detection, and known-item search. For the instance search task, we use the same approach as the last year's system with some modifications in quantization and fusion of query representations. The experiments show improved performance of this year's system compared to the last year's system. For the semantic indexing task and the multimedia event detection task, we report the experiments using NII-KAORI-SECODE framework. Our best run for the semantic indexing task using local features achieve 20.7% (MAP), ranked 6/15 groups. Especially, the run using only one local feature achieves 18.9% (MAP). As for the multimedia event detection task, we only use global features to serve as a baseline method for comparison with other complicated systems using local features and multi-modal features.**

I. Instance Search

This year the algorithm submitted by NII for TRECVID instance search share most similar points with the one submitted last year [1]. To summarize, a large vocabulary (up to 1 million visual words) quantization based Bag-of-Words framework [2] is adopted; inspired by the work [3], we take each video in the database and each query topic as unit for ranking, here each video/query topic is represented by a pool of local features (e.g. color sift feature [4] in our submitted runs) extracted from all the sampled frames/images composing itself, as shown in Figure 1 and Figure 2; finally an inverted indexing efficiently return the ranking list in seconds(see Figure 2). The main differences lies in following two aspects: (1) Instead of clustering by hierarchical k-means algorithm last year, this year approximate k-means [5] is used for getting more accurate clustering centers. With this step, we improved performance from 51% mAP to 55% on the instance search dataset last year. (2) Based on comparison in paper [6], dense sifts extracted from the object region do not boost performance much, but will increase the computational burden greatly, so we skip it this year. Instead, for each query topic, we separately inquired the dataset with whole query images composing it, i.e. including both background and object regions, and also object regions only, then we fuse these two ranking lists. The final performance table of our best run is shown in Figure 3. We specially submitted the run by using last year's algorithm for comparison, and the performance was improved from last year's 8.8% mAP to 16.8% this year.

II. The NII-KAORI-SECODE Framework

In our framework, features are extracted from the input keyframes representing for shots. We extracted 10 keyframes per shot that are spaced out equally within the provided shot boundary. In the training stage, we use these features to learn SVM classifiers. These classifiers are then used to compute the raw output scores for the test image in the testing stage. These output scores can be further fused by taking the average for computing the final output score. In order to return K shots most relevant for one concept query that then are evaluated and compared in TRECVID benchmark, all normalized final output scores of shots are sorted in descending order and top K shots are returned. In the case of a shot consisting of several sub-shots, only the maximum score among subshots' scores is used for that shot.

As for feature extraction, dense sampling is used for finding keypoints from which SIFT and COLORSIFT descriptors are

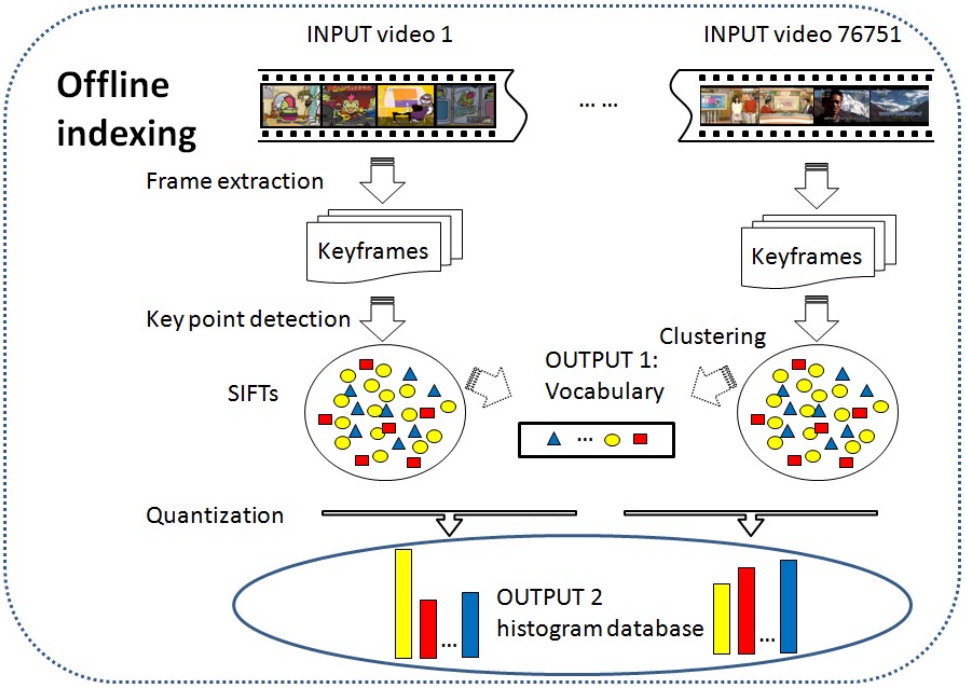


Fig. 1. Framework of offline indexing.

extracted. We used GreedyRSC+KMeans to find 500 clusters for vector quantization. Then a standard bag-of-words with soft-weighting was used to form the feature vector.

III. Semantic Indexing

The performance of runs with different configurations is available online at: http://satoh-lab.ex.nii.ac.jp/users/ledduy/Demo-KAORI-SECODE/.

We submitted 2 full-type runs, 1 light-type run, and 1 no-annotation-type run. For the no annotation run, for each concept, its name is used to craw max 300 images from Google Image Search Engine (medium image size, photo image only). These images are considered as positive samples, and used for training one concept detector. This concept detector is used to apply to the initial 300 image list, and then the scores are used to re-rank. Finally, top 100 images are kept for training The performance of these runs is shown in Table I:

IV. Multimedia Event Detection

We extracted one keyframe for every 4 seconds. Total [keyframes - clips] for MED12TEST (Test samples): [3,300,006 - 98,118]. Total [keyframes - clips] for EVENTS (Positive samples): [161,569 - 4,392]. Total [keyframes - clips] for BACKGROUND (Negative samples): [289,439 - 10,671].

For each keyframe, we extract global features such as color histogram (Luv, HSV), local binary patterns (LBP), edge orientation histogram (EOH). For each clip, using max pooling to aggregate features of the keyframes extracted from that clip.

All clip EVENT i are used as POSITIVE training samples. All clip of BACKGROUND are used as NEGATIVE training samples. LibSVM with RBF-chi-square kernel is used for learning models to predict probability of an input image belonging to one EVENT i.

Total processing time is around 48 hours using 300 cores, 2.26 - 2.5 GHz Intel Xeon, 2GB-4GB RAM per core, 30TB Data Server.

The scores of $NDC, PFa, PMiss$ are 0.8760, 0.0048, 0.8163 respectively.

V. Known-Item Search

For the known-item search task, we submitted two automatic runs with original and translated metadata and two corresponding interactive runs

A. Automatic runs

In the automatic runs, we only used metadata provided along with the IACC video data. Initially, the text data from metadata are preprocessed. Next, they are indexed for the future retrieval.

The preprocessing phase is quite simple. We extracted six fields for indexing, namely *title, description, subject, keywords, comments, notes, shotlist, segments*. However, there are metadata files which are not in English, while queries are always in English. As a result, we need to translate the non English metadata into English automatically. Google Translate was used for automatic translation. Before translation, some preprocessing steps are executed on the text such as *eliminating punctuations, and special characters.*

Lucene is then used as the engine for text indexing and retrieval. In the retrieval phase, visual cues and original queries are combined into single queries. The advantages of the combination step are *(i). keywords in visual cues are duplicated*

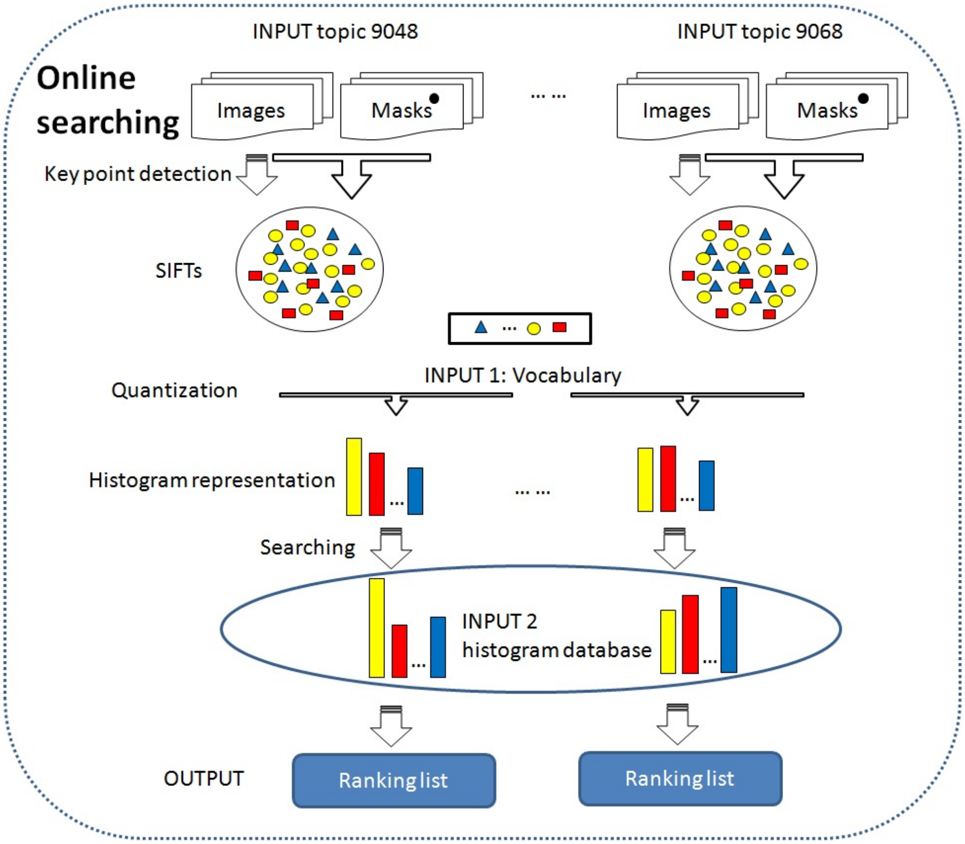


Fig. 2. Framework of online searching.

TABLE I
PERFORMANCE OF OUR SUBMITTED RUNS FOR SIN TASK.

RunID	Run Type	Description	MAP (%)
F_A_nii.Kitty-AF1_1	Full	Fusion of features such as dense6mul.rgbsift.norm3x1, dense6mul.oppsift.nom3x1, dense4.sift.norm3x1, dense6.sift.norm3x1	20.7
F_A_nii.Kitty-AF1_2	Full	Single local feature, dense6mul.rgbsift.norm3x1, dense sampling, step size of 6 pixels, rgbSIFT descriptor, grid 3x1	18.9
L_A_nii.Kitty-AL3_3	Light	Fusion of features such as dense6mul.rgbsift.norm3x1, dense6mul.oppsift.nom3x1, dense4.sift.norm3x1, dense6.sift.norm3x1	26.6
L_E_nii.Kitty-EL4_4	Light, No annotation	Single local feature, dense6mul.rgbsift.norm3x1	4.4

to gain a bigger weight for retrieval and *(ii). no information in original queries is missed.*

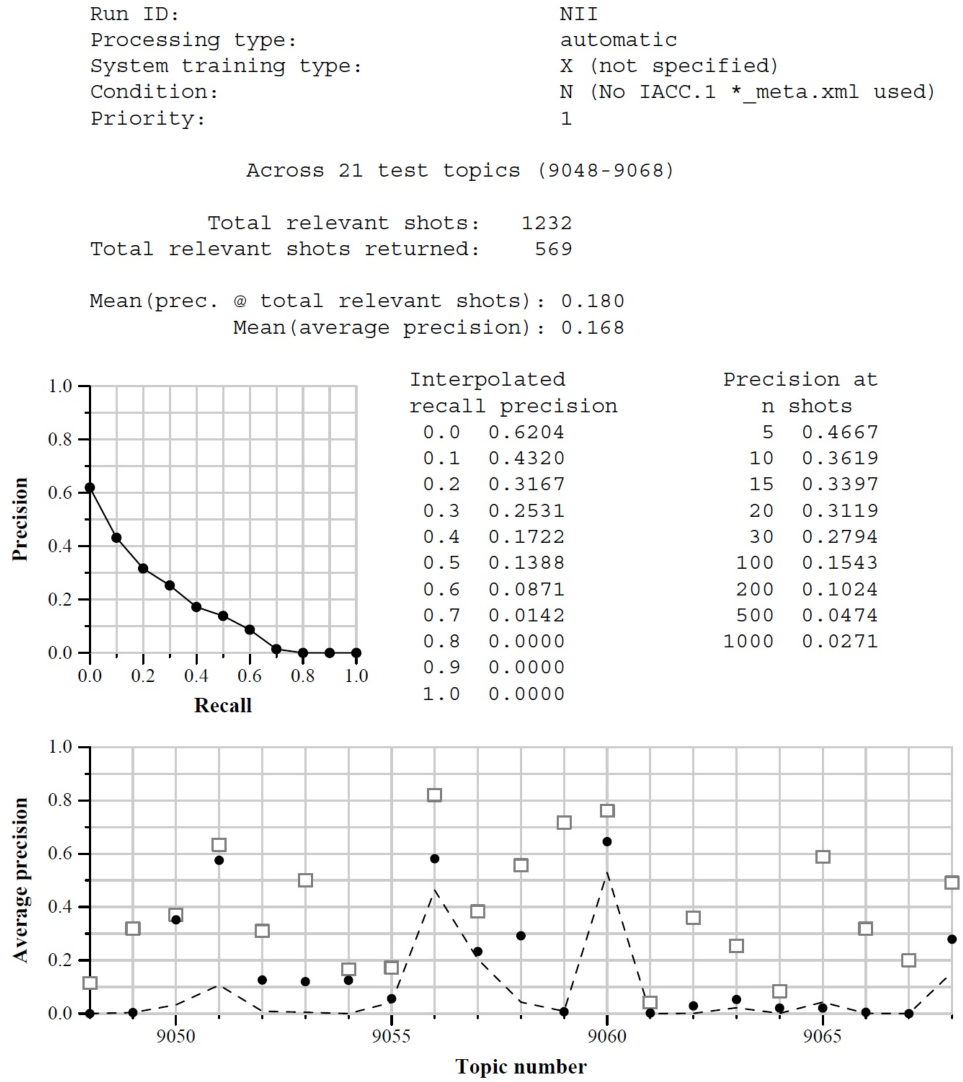
B. Interactive runs

We implemented a graphic user interface used for interactive runs, so that users can easily interact with the system for faster retrieval process. The system is composed of three main components

- ***Ranked list:*** With a similar procedure for retrieval as the automatic runs, a ranked list of videos is produced by Lucene. Each video is displayed by five representative key frames vertically. And the videos are presented page by page. Users can browse over the list of ranked videos to look for the target video. Once they see any video that is likely the target video, they can add it to the candidate segment for a more detail view
- ***Candidate segment:*** In this segment, videos are shown in a more detail view. Each video is represented by 15 key frames horizontally. With a finer view, it is easier for user to decide which one has high probability of being the target video. And if they are relatively sure about any video, they could add it to the oracle queue
- ***Oracle queue:*** contains a queue of videos waiting for being sent to the web-based oracle for verification. If some video is verified to be the correct one, the searching process is terminated

C. Experimental Results

For automatic task, we submitted two runs. The first run, NII1, using original metadata, and the second run, NII3, using auto-translated metadata. The comparison result is shown in Figure 6. It is not expected when seeing the performance



```
Run ID:                             NII
Processing type:                    automatic
System training type:               X (not specified)
Condition:                          N (No IACC.1 *_meta.xml used)
Priority:                           1

            Across 21 test topics (9048-9068)

         Total relevant shots:   1232
Total relevant shots returned:    569

Mean(prec. @ total relevant shots): 0.180
           Mean(average precision): 0.168
```

```
Interpolated              Precision at
recall precision            n shots
 0.0  0.6204                5  0.4667
 0.1  0.4320               10  0.3619
 0.2  0.3167               15  0.3397
 0.3  0.2531               20  0.3119
 0.4  0.1722               30  0.2794
 0.5  0.1388              100  0.1543
 0.6  0.0871              200  0.1024
 0.7  0.0142              500  0.0474
 0.8  0.0000             1000  0.0271
 0.9  0.0000
 1.0  0.0000
```

Fig. 3. Best run with this algorithm.

of NII1 is higher than that of NII3. This may due to some information of metadata skipped during the preprocessing step before translation.

There are nearly half of topics found (176 over 361) by NII1. The detail result is shown in Figure 6 which shows the number of topics that have ranks in ranges 1 to 10, 11 to 20, etc...

- ***Original metadata***: 12 over 24 topics found
- ***Auto-translated metadata***: 15 over 24 topics found

The number of topics found by NII3 is lower than NII1 (165 over 361). The detail result of NII3 is shown in Figure 7. For interactive task, we submitted two runs: NII2 (original metadata) and NII4 (auto-translated metadata). For each topic, a ranked list of 5,000 videos is produced by Lucene search engine. The comparison is shown in Figure 8. We can see that it is comparable to other teams's performance. Figure 9 and Figure 10 represent detailed information for NII2 and NII4 runs respectively.

REFERENCES

[1] D.-D. Le, C.-Z. Zhu, S. Poullot, V. Lam, D. Duong, and S. Satoh, "National institute of informatics, japan at trecvid 2011," in *TRECVID Workshop*, 2011.

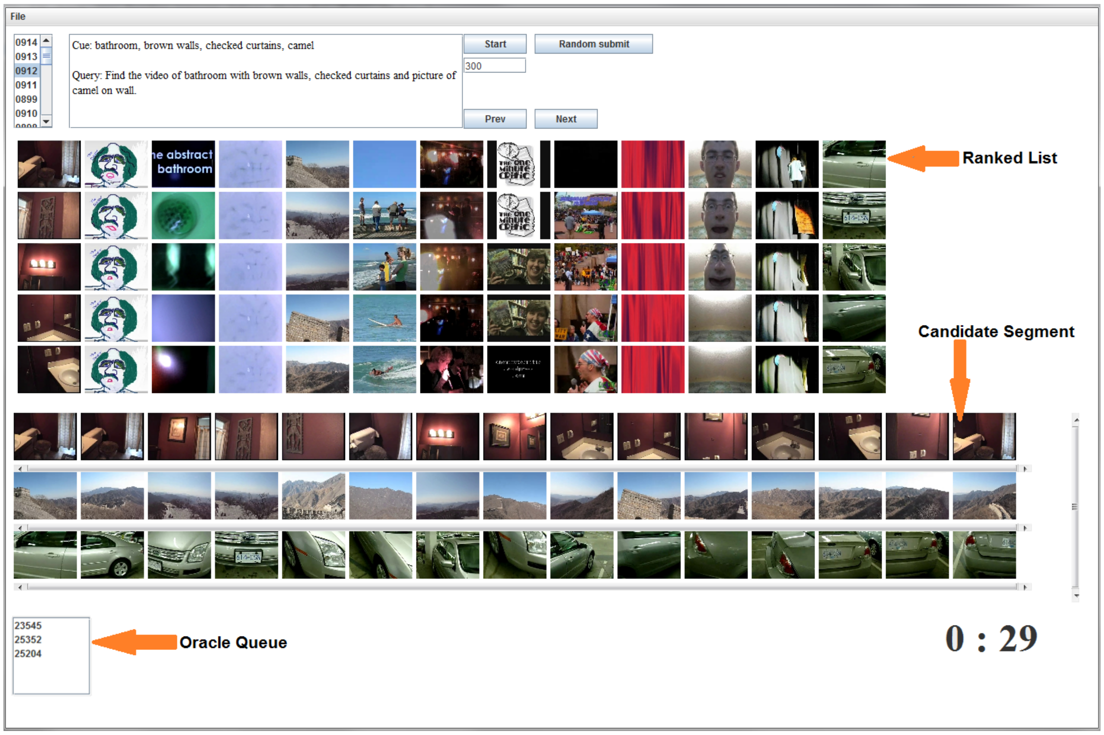
[2] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV*, 2003.

[3] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *CVPR*, 2008.

[4] "Featurespace," http://www.featurespace.org/.

[5] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[6] C.-Z. Zhu and S. Satoh, "Large vocabulary quantization for searching instances from videos," in *ACM International Conference on Multimedia Retrieval*, 2012.



Cue: bathroom, brown walls, checked curtains, camel

Query: Find the video of bathroom with brown walls, checked curtains and picture of camel on wall.

23545
25352
25204

0 : 29

Fig. 4. Interactive System

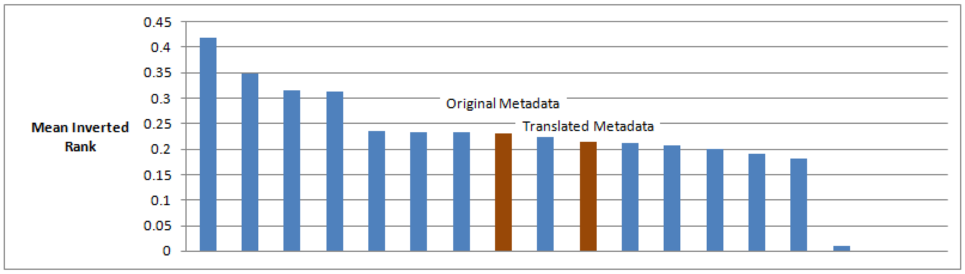


Fig. 5. Automatic result

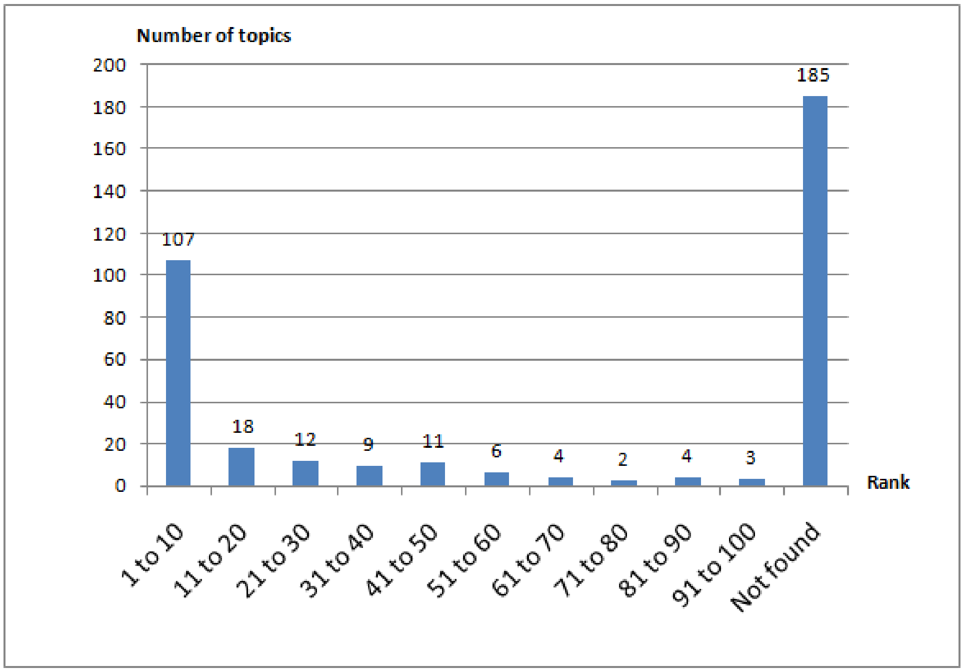


Fig. 6. NII1 run

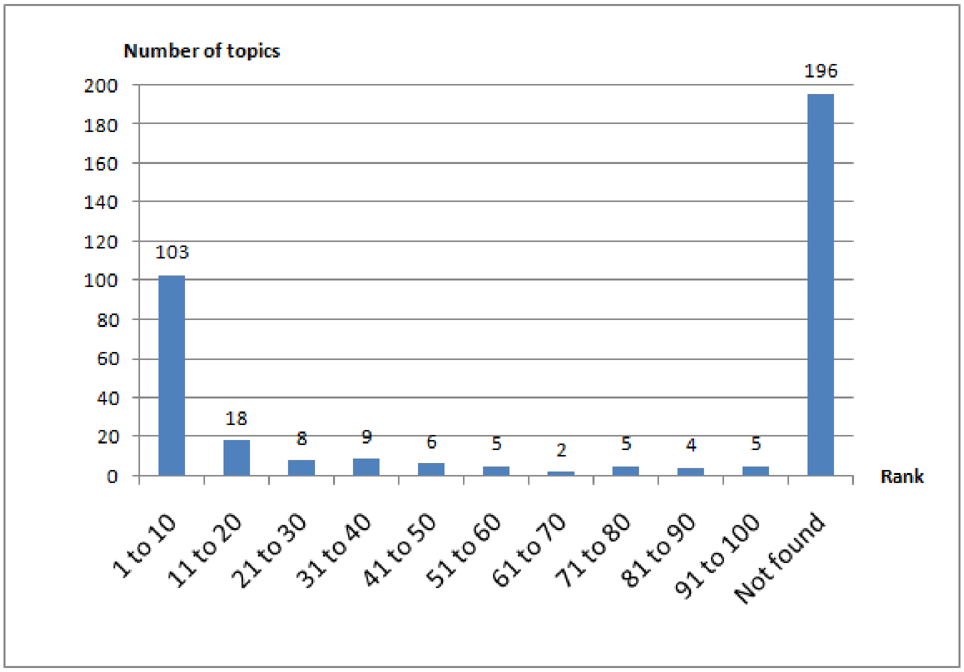


Fig. 7. NII3 run

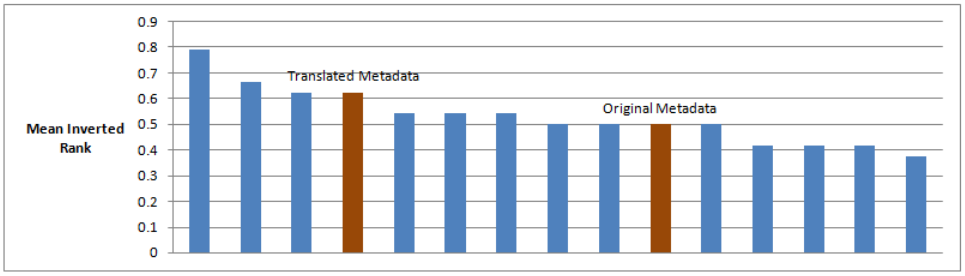


Fig. 8. Interactive result

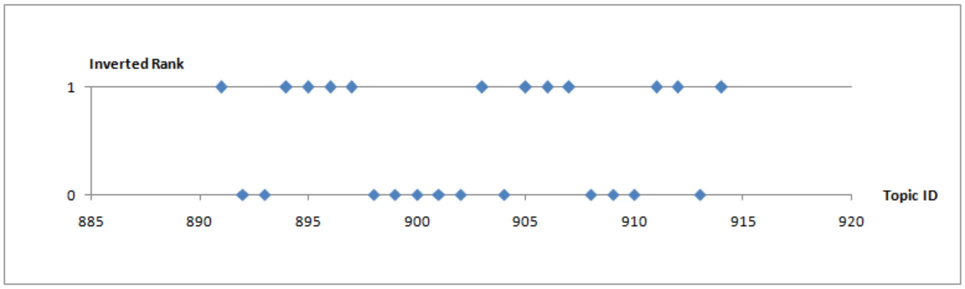


Fig. 9. NII2 run

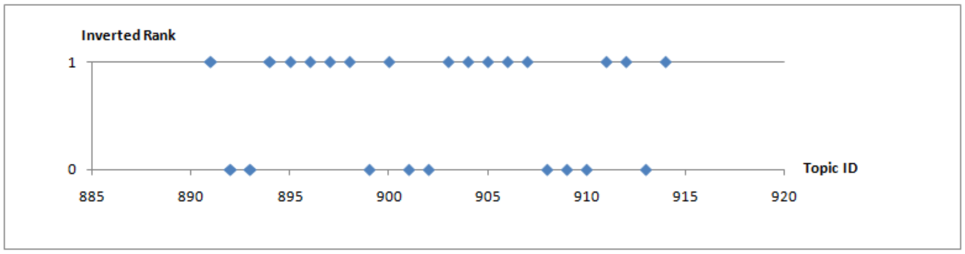


Fig. 10. NII4 run