Consensus Ranking for Increasing Mean Average Precision in Keyword Spotting

Anders Hast
Uppsala, Sweden.
anders.hast@it.uu.se

Department of information technology,
division of visual information and interaction.
Uppsala University,

Abstract

Word spotting use a query word image to find any instances of that word among document images. The obtained list of words is ranked according to similarity to the query word. Ideally, any false positives should only occur in the end of that list. However, in reality they often occur higher up, which decreases the so called mean average precision. It is shown how creating new ranked lists by re-scoring using the top n occurrences in the original list, and then fusing the scores, can increase the mean average precision.

1 Introduction

Transcription of large collections of handwritten material is a tedious and costly task. Nevertheless, images of documents can still be made searchable by a technique called keyword spotting, or simply word spotting [GSGN17, LRF$^+$12]. Word spotting can be considered as a special case of image retrieval, where the goal is to find all instances of the query word in the document image collection at question. When searching in several document images, using a query word in the form of an example image, there are often some false positives in the retrieved list. Therefore ranking is performed so that the words most similar to the query word will appear in the top and the most dissimilar, but still similar enough to be presumed to be correct, will appear in the bottom of the list.

In this paper, methods for improving the ranked list will be evaluated and discussed. The main idea is to take the top n occurrences in the ranked list and then rank it again against each one of them, thus creating n rankings, which then are fused to create a better overall ranking.

2 Background

Generally, in both information retrieval and object-image retrieval the retrieved result is ranked according to its relevance. Similarly, the retrieved regions supposedly containing the searched word of all the pages are collected and ranked according to their similarity score. This procedure involves pairwise comparison, in which the query word image is compared to all retrieved word images using some scoring function (i.e. computing the distance

In: A. Amelio, G. Borgefors, A. Hast (eds.): Proceedings of the 2^{nd} International Workshop on Visual Pattern Extraction and Recognition for Cultural Heritage Understanding, Bari, Italy, 29-Jan-2020, published at http://ceur-ws.org

between feature vectors obtained from the word images) that will give a score, also known as confidence value, for each comparison, i.e. each word image.

Re-ranking usually refers to the procedure of ranking all over again using a more exact method for computing the score. However, in this work, only the situation where the same scoring function is used will be considered, even though there is nothing that prohibits the use of different scoring functions. The use of machine learning techniques to train the ranking model is often referred to as "learning to rank" [LLZ09], which includes methods like LambdaMART and LambdaRank, which is based on RankNet [Bur10]. Relevance feedback [Har92] could also be used with the proposed method by letting the user choose the relevant words for ranking and fusion.

Normally, some score normalisation procedure is necessary in order to make scores, which are obtained using different scoring functions, e.g. from different information retrieval systems, comparable to each other [WCB06]. This is not necessary here, since the proposed algorithm use the same scoring function.

In information retrieval, data *fusion* [Wu12] is a technique for combining the ranked lists from several retrieval methods [SF95]. It has been noted that the fusion together of document rankings can yield a higher level of relevance than either of the individual methods [Sme98, MS05]. It has also been noted that in general it is better to fuse scores rather than ranks [BP09]. One common technique for score based fusion in information retrieval is CombSUM [BKFS95], where the scores from several lists are simply added together to obtain a new scores. By multiplying this score by the number of lists, i.e. frequency, where the document occurs, combMNZ is obtained [BC17, WH18]. Herein, the average of scores for each instance of the word will be used, since the new score will be used for purging those that fall under a certain threshold τ

The relevance of documents in an information retrieval system can vary quite a lot and can therefore be considered being fuzzy, while the word found in word spotting is either the correct word or not, and can therefore being considered binary. Of course, there are some exceptions like differences in spelling or the same word with different semantics etc. Nevertheless, it is possible to utilise this fact by taking the top n occurrences in the first ranking and rank the whole list against each one of them creating n rankings, which then are fused to create a better ranking.

2.1 Fusion for keyword spotting

Rusiñol et al. [RnL14] propose three different techniques for data fusion techniques, where variability in writing style is handled by preforming multiple queries and then combining the results. Louloudis et al. [LKG12] evaluate three techniques based on the rank position. A systematic evaluation of different fusion techniques was presented by Peña et al. [PnSMVG10]. Wei et al. [WGS15] discuss how *query expansion* (QE) [AZ12] that produce one ranked list per expanded search, can be subsequently merged to obtain the final ranked list. Later on, this method will be compared to the proposed method.

3 Consensus Ranking and Query Expansion

A learning and segmentation free word spotting engine [HF16, HCV18] was used that search for an example image, referred to as a query word, in one or several document images [HCVA19]. Tentative words that fall below a certain threshold τ will be discarded as garbage (i.e. a so called *negative*). It is obviously important to set this threshold so that the negatives are mostly true negatives and the positives are mostly true positives. However, due to the nature of the problem there will often be some false positives as well as false negatives.

Moreover, the matching of the query word and the found tentative words cannot be too precise as some words would be missed, i.e. the *recall* will be lower. At the same time it cannot be too relaxed so that many false positives will be found, i.e. the *precision* will be low (these terms are explained under section 4). In the tests presented a rather tight threshold was used to find mostly true positives, and then perform QE on these, where new searches are performed, using each found word as a query on the same page as the word is found, as well as on the previous and following page. This will ensure that the QE will have enough instances of the word to query on, but not too many as it will increase search time and also increase the risk of finding false positives. Hence, even if this approach uses the same underlying architecture as a previously published approach [VHF19], it is still different as that one only used words found on the same page. Nevertheless, QE will increase the recall substantially while still keeping the precision high.

The ranked list will be ranked depending on the similarity to the original query word only. Therefore, the score value can be lower for certain words in the end of the ranked list, than the threshold τ used in each individual search. This fact can be utilised to purge the end of the list for all words having a score value below τ or some other threshold depending on the quality of the found words.

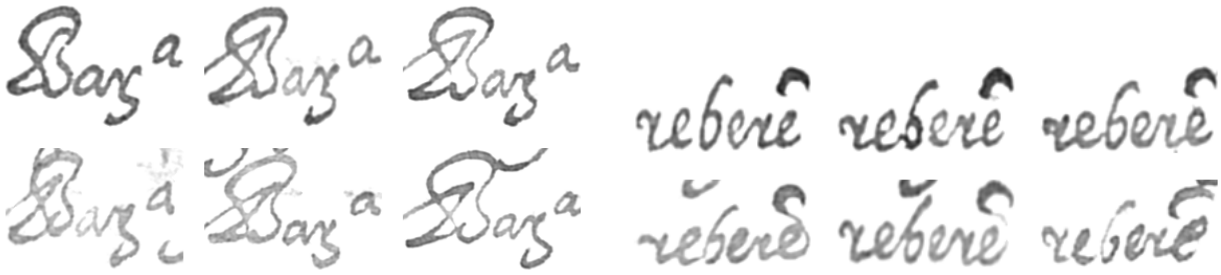
The idea behind QE is that the found words contain some variation that will help finding even more words that the single query word could not find, i.e. the score will be quite different depending on the variation. Similarly, the presented *consensus ranking* utilise this fact and takes the top n words from the resulting list from the QE search and score each of them to the ranked list to create n new ranked lists. Hence, each list is a bit different and by fusing them a better ranked list is obtained.

The above approach will therefore be different from the one of Wei et al. [WGS15] where the the QE produce several lists, i.e. one ranked list per search. These lists are ranked on the similarity of each query word, that are subsequently *merged* to obtain the final ranked list. Each QE will generally not find exactly the same words and therefore each ranked list will be a bit different. When merged, the list will be as complete as possible, but the fusion will depend on how many expanded query searches finds each word. In general, the more queries that end up finding one word the better, since the final score will be computed from more lists. However, by taking the top n words in the final list resulting from QE and making n new lists by re-scoring, this problem will not occur. This approach will be referred to as *consensus ranking* (CR). Nonetheless, one possible drawback with it is that the QE list will be ordered depending on the similarity to the original query word. If this word is not for some reason very representative for the words in the document, then the approach by Wei et al. might be better. Therefore, it will be investigated herein the effect of utilising their approach to merge the lists resulting from QE and then use the top n words in that list to perform the proposed CR. This approach is later referred to as *ML-CR*.

4 Method

First as a proof of concept for the proposed algorithm, the ranking of digits in the MNIST data set [LCB10] is performed, which showed good performance even when the top n digit images contains some false positives (see section 5).

Then tests were performed both on the Esposalles data set [RFS+13] and the Bentham data set from the Transcribe Bentham project [MTW11]. Four different words were chosen as query words and the initial search was performed on 10 pages for each data set. For each word 6 instances were chosen in order to capture some variance in the handwriting. By comparing all words in the ground truth with every other word, a confusion matrix was obtained containing scores (similarity). For both data sets, the 3 words having highest mean value in each column of the matrix, i.e. being as equal to the others as possible, were therefore considered being representative query words. The 3 words having lowest mean were also used to see how the algorithms involved cope with such situation when the query word is less representative, i.e. more difficult, as it might contain strokes from adjacent lines, being written differently or having some kind of degradation. Examples of both variants are shown in Figure 1.

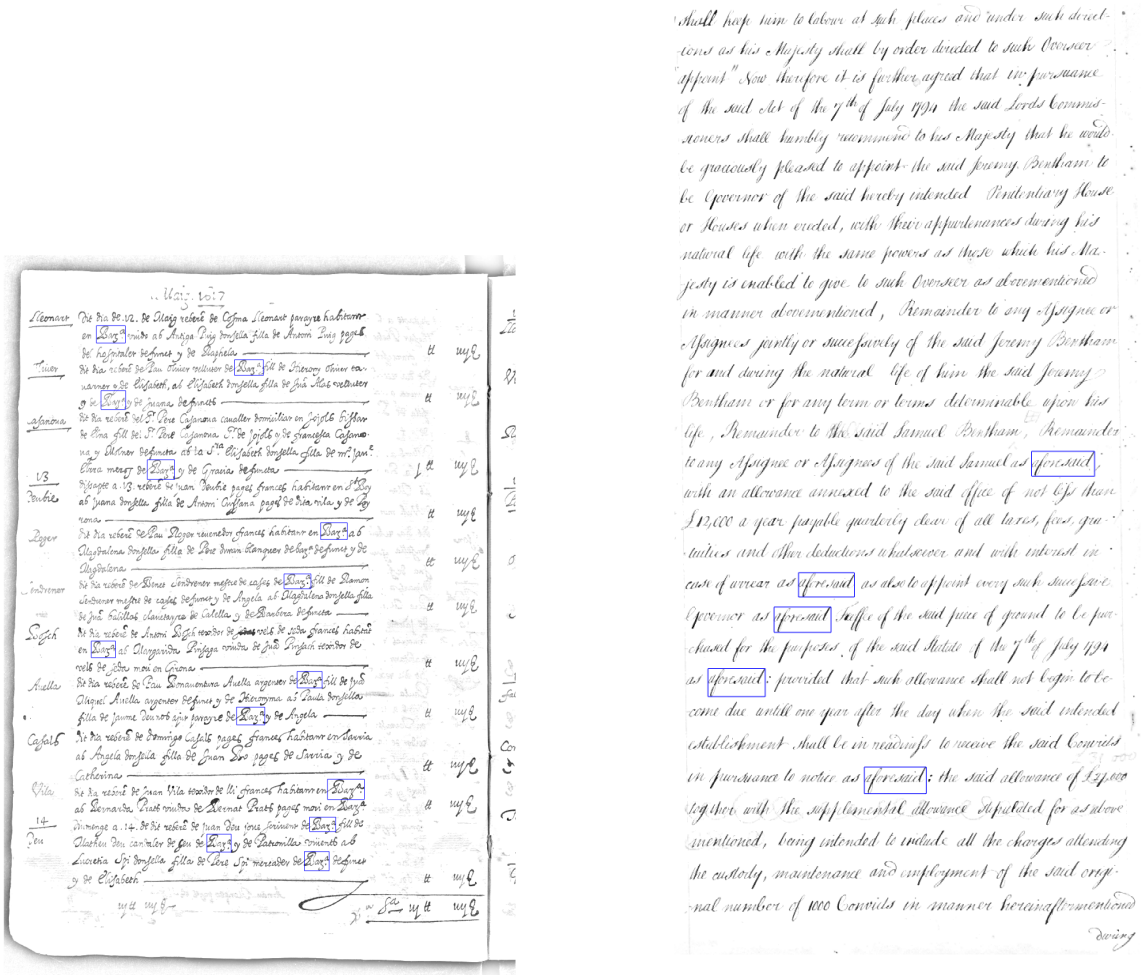


(a) The query images Barca

(b) The query images "reber".

Figure 1: Example of two query words and their images used in the tests. The top row shows word images that give higher mAP for word spotting and the bottom row shows words that give lower mAP, since they contain strokes from surrounding lines or are written differently. (**a**) "Barca" (**b**) "reber".

Figure 2 shows two examples of word spotting of the word "Barca, which is the abbreviation for "Barcelona" in the the Esposalles data set 2a and the word "aforesaid" in the Bentham data set 2b. Here a background removal technique has been used so that the strokes appear with less disturbing background noise [VHS17]. One can note how the bounding boxes captures the word well even if they vary quite a lot in size and appearance [VH17].



(a) The word Barca has been found on one page in the Esposalles data set.

(b) The word aforesaid has been found on one page of the Bentham data set.

Figure 2: Example of how the word spotting system finds words in the (**a**) Esposalles data set, and (**b**) the Bentham data set. (Figure best viewed in colour)

4.1 Mean Average Precision

The word spotting algorithm can be evaluated by computing the Precision-Recall curve. Precision is the number of relevant (true positive) objects found divided by the number of all retrieved objects, and Recall is the number of relevant objects found divided by the number of all relevant objects in the set [Pow07]. They are computed as:

$$Precision = \frac{|retrieved \cap relevant|}{|retrieved|} \tag{1}$$

$$Recall = \frac{|retrieved \cap relevant|}{|relevant|} \tag{2}$$

Many prefer using a single value for easily comparing different approaches, such as the mean Average Precision (mAP), which corresponds to the area below the Precision-Recall curve, which is computed using the following equation:

$$\text{mAP} = \frac{\sum_{n=1}^{|\text{retrieved}|} P@n \times r(n)}{|\text{relevant}|} \tag{3}$$

where $P@n$ is the precision at the n top-most returned results, and $r(n)$ is a binary function indicating whether the n-th item in the returned ranked list is a relevant object (true positive).

5 Results

The results from both the proof of concept and applying QE and CR on the Esposalles and the Bentham data sets are reported in this section.

5.1 Proof of Concept

The proposed CR algorithm was used for ranking of digits in the MNIST data set [LCB10], as a proof of concept. Feature vectors were computed (as in [HLV19]) for 2000 images containing 200 images for each digit. The set was divided into two disjoint sets with 100 number of occurrences of each digit in each set. The idea is to rank the images in the second set according to how similar they are to each digit in the first set. This was done by computing the cosine distance (i.e a score value) and therefore the list contains the nearest neighbours (NN) in a descending scale. The results are shown in Table 1. Here it was chosen to use only a maximum of 12 of the nearest neighbours having a distance equal to or above 0.8. However, it should be noted that as for MNIST this did not prohibit from performing CR using some false positives among the top n images. Actually, it was quite common, with an average of 17.14% for the reported tests, which the low mAP also confirms. Despite this, the CR still increases the mAP with an average of 12.30%, which proves the efficiency and the stability of the proposed algorithm.

Table 1: Comparison of mAP for the digits in MNIST between nearest neighbour search (NN) and applying CR on that result. Increase in percentage is reported.

Digit	**NN**	**CR**	**%**
0	0.5915	0.6873	16.20%
1	0.5983	0.6540	9.31%
2	0.4514	0.5195	15.10%
3	0.4416	0.5164	16.94%
4	0.3993	0.4008	0.38%
5	0.3430	0.3809	11.07%
6	0.5646	0.6395	13.27%
7	0.4208	0.4443	5.59%
8	0.5400	0.6421	18.92%
9	0.4369	0.5076	16.20%
mean	0.4787	0.5393	12.30%

5.2 Performance on real data sets

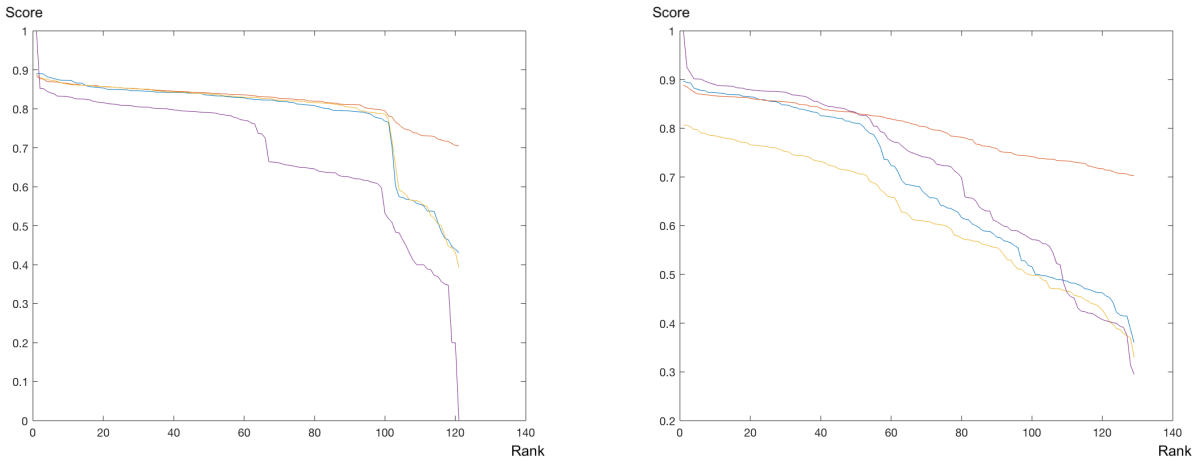
Three different approaches were used that are reported in this section, which are producing a new ranking after fusion of the results obtained from a word spotting algorithm (subsequently referred in the tables as WS). Query Expansion is performed on the result obtained from WS and scored against the original query word. This approach will be referred to as QE in both images and tables. The fusion approaches are as follows:

- CR: Consensus ranking: re-score the QE list using the top n words and perform fusion.
- ML: Merge all lists from QE and perform Fusion.
- ML-CR: Perform CR on the result from ML.

In the tests n was chosen to be 12, with the condition that the score for each and one word was higher than a specified threshold, in this case 0.8. Hence n might be lower than 12 for some cases. This threshold was set

a bit higher than the threshold $\tau = 0.7$ for regarding a word as a correct word, in order to be more on the safe side when re-scoring, Both conditions are assuring that the words that are being used for re-scoring are true positives to a high probability. Note, that if the QE do a bad job in the first place this will not hold. However, as shown in the proof of concept, the CR can usually still improve the ranking, even if some false positives are found among the top n candidates for re-scoring.

The plot in 3 shows the confidence score (or similarity value), where the ranked lists are from QE (magenta), CR (blue), ML (red), CR-ML (yellow). The $y - axis$ shows the score and the $x - axis$ the position in the list. One can note that the query word used was taken from the same page images since the highest score is 1.0 (which means it finds itself) and then decaying for QE. One can also note that the score for ML is generally higher at the end of the ranked list since it by definition cannot go below the threshold τ used both for WS and QE. Nevertheless, the score for CR can be lower since the re-scoring between words can result in low scores even if both are true positives, as they might be rather dissimilar.



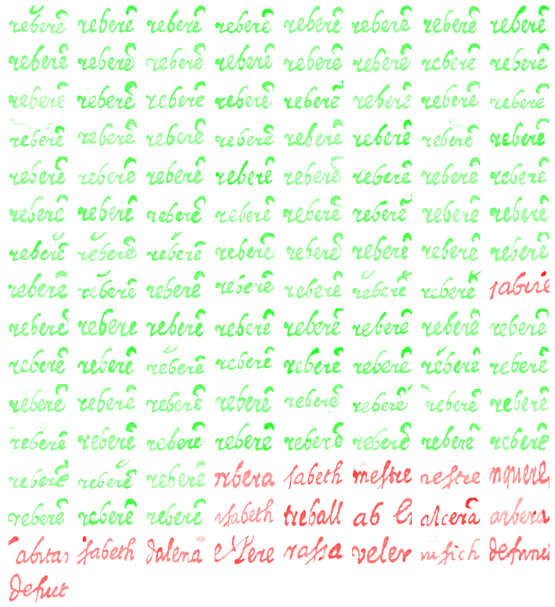
(a) "reber" (b) "Barca"

Figure 3: Comparison of the ranked lists from QE (magenta), CR (blue), ML (red), CR-ML (yellow). The $y - axis$ show the score and the $x - axis$ the position in the list (rank). (Figure best viewed in colour)

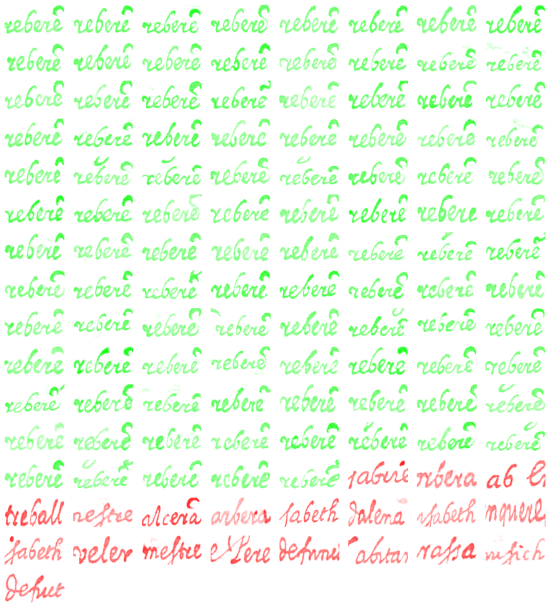
One advantage with the CR and CR-ML is therefore that it is possible to purge the list by removing those under a certain threshold. This will make it easier for a user to investigate the result if obvious false positives are removed. Therefore, the user of a search tool or a transcription tool using word spotting [HV18], only needs to concentrate on a few items (true positives) in the end of the purged list that might be miss ranked and end up scattered among the false positives.

A comparison is shown in Figure 4 and 5 of the output from the word spotting system (WS) 4a and the result of applying fusion through consensus ranking (CR) 4b. The true positives are depicted in green and the false positives in red. Once again the background removal technique has been used. One can note how the miss ranked words are all pushed to the bottom of the list. The ranked list obtained by merging lists (ML) 4c and performing a CR on ML 4c is also shown. One can note how ML generally performs worse than CR. This is due to the fact that τ was set rather tight in order to avoid false positives. This causes the algorithm to find fewer correct words in both the WS and in each single search in the QE. However, in general this method leads to both high recall and precision. By lowering the threshold τ, ML will perform better, with the drawback of having much more false positives in the finals lists.

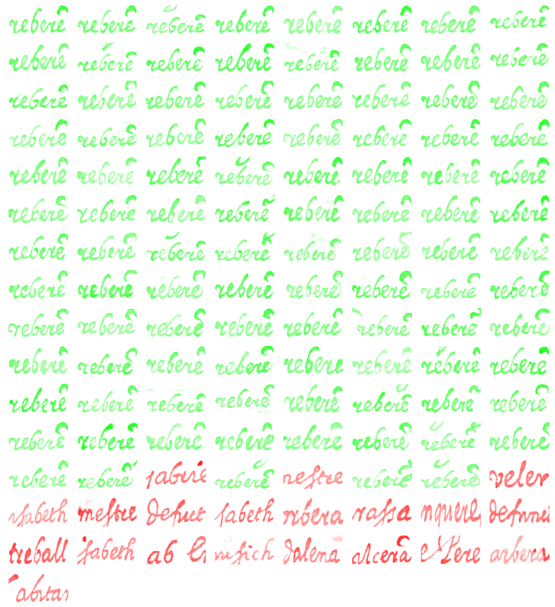
One can note how CR improves the position of miss ranked words quite a lot, even if the mAP improvement is modest. The results are not always perfect, since the same score function is used both for QE and CR, which cannot be too precise in order to still capture the variation in handwriting.



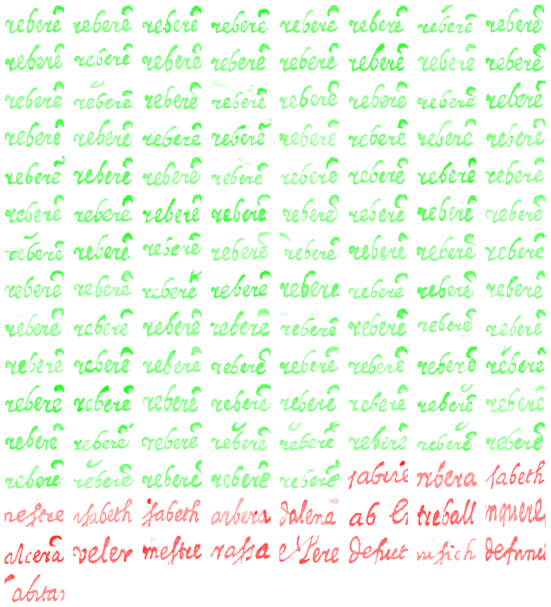
(a) List ranked using the original query word.



(b) List after consensus ranking.

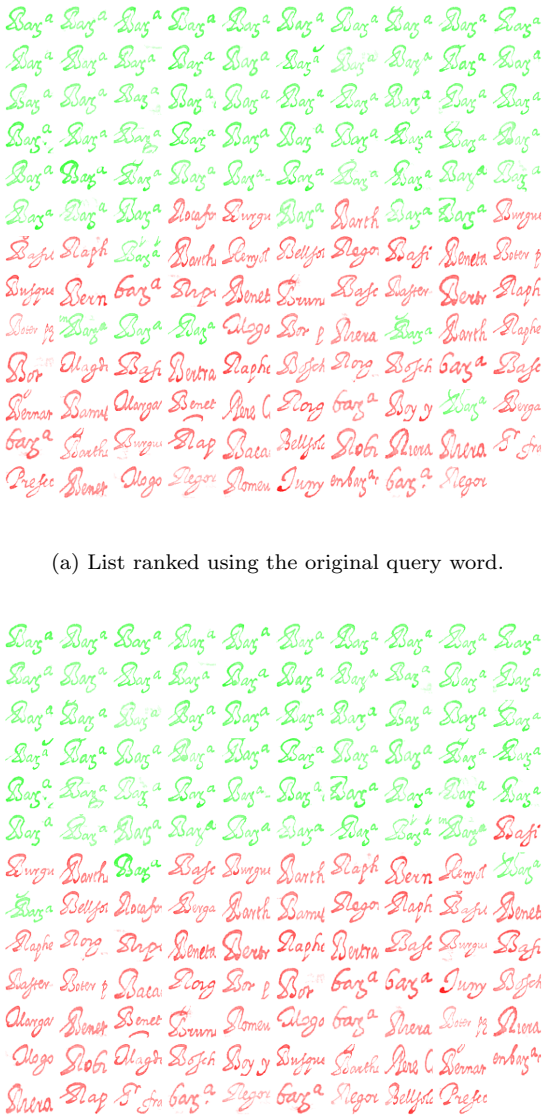


(c) List ranked after merging. list

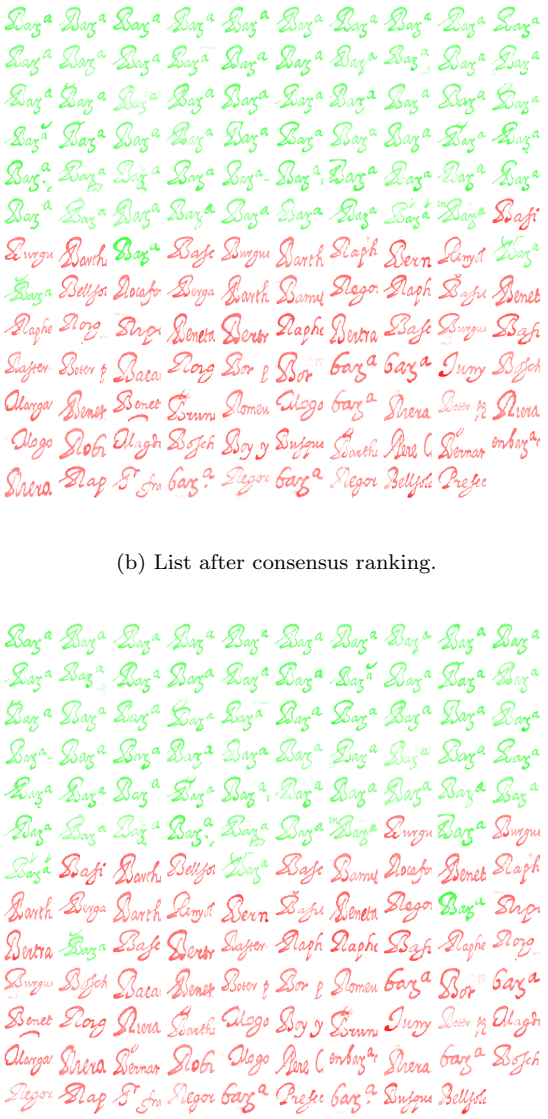


(d) List after merging list and performing consensus ranking.

Figure 4: Example of a comparison of the output from the word spotting system using query expansion searching in 10 pages, where the true positives are depicted in green and the false positives in red. (**a**) the original ranked list. (**b**) the result of applying fusion using the proposed consensus ranking scheme. (**c**) fusion by merging lists from each query expansion. (**d**) Consensus ranking performed on the merged list. (Figure best viewed in colour)



(a) List ranked using the original query word.



(b) List after consensus ranking.

(c) List ranked after merging.

(d) List after merging list and performing consensus ranking.

Figure 5: Example of a comparison of the output from the word spotting system using query expansion searching in 10 pages, where the true positives are depicted in green and the false positives in red. (**a**) the original ranked list. (**b**) the result of applying fusion using the proposed consensus ranking scheme. (**c**) fusion by merging lists from each query expansion. (**d**) Consensus ranking performed on the merged list. (Figure best viewed in colour)

Table 2 show the result (in mAP) of running the different algorithms on several different words, as explained earlier for the Esposalles data set. Each row shows the average of mAP from using three different query words. One can note how QE improves over WS, but also how CR pushes up the mAP. The increase might seem small but in reality the effect is that the ranked list will contain a longer sequence of true positives and the false positives will be aggregated in the end of the list, instead of being scattered in the middle of the list.

Table 3 shows the effect of choosing words that give worse results as explained earlier. One can note how the WS performs worse, but that QE still deals with that problem in an elegant way. Generally, CR pushes the mAP up higher. Furthermore, table 4 and Table 5 shows some results for the Bentham data set.

Table 2: Comparison between different methods, where the average of three representative occurrences for four different words from the Esposalles data set is reported. The increase compared QE qith WS as well as CR, ML and ML-CR with QE.

Word	**WS**	**QE**	**CR**	**ML**	**ML-CR**
Bara	0,844631	0,960884	0,982386	0,869193	**0,982389**
donsella	0,933588	0,971495	**0,980789**	0,978913	0,980245
pages	0,952578	0,995402	0,998977	0,999088	**0,999105**
rebere	0,990066	0,999624	**1,000000**	0,999674	**1,000000**
Mean	0,930216	0,981851	**0,990538**	0,961717	0,990435
Increase		$5,55\%$	**0,88%**	$-2,05\%$	$0,87\%$

Table 3: Comparison between different methods, where the average of three difficult occurrences for four different words from the Esposalles data set is reported. The increase compared QE qith WS as well as CR, ML and ML-CR with QE.

Word	**WS**	**QE**	**CR**	**ML**	**ML-CR**
Bara	0,116402	0,639233	0,649848	**0,656740**	0,655781
donsella	0,401411	0,906396	**0,911237**	0,887814	0,895160
pages	0,293333	0,969244	0,981201	**0,982882**	0,982378
rebere	0,389439	0,965221	0,970122	0,967734	**0,970262**
mean	0,300146	0,870024	**0,878102**	0,873792	0,875895
Increase		189,87%	**0,93%**	0,43%	0,67%

Table 4: Comparison between different methods, where the average of three representative occurrences for four different words from the Bentham data set is reported. The increase compared QE with WS as well as CR, ML and ML-CR with QE.

Word	**WS**	**QE**	**CR**	**ML**	**ML-CR**
aforesaid	0,900281	0,942199	**0,956681**	0,453054	0,841440
amount	0,750419	0,843464	**0,935143**	0,644368	0,898775
intended	0,833333	**0,900000**	**0,900000**	0,836233	**0,900000**
Jeremy	0,865481	0,957196	**0,971777**	0,801528	0,941335
Mean	0,837378	0,910715	**0,940900**	0,683796	0,895388
Increase		8,76%	**3,31%**	-24,92%	-1,68%

Table 5: Comparison between different methods, where the average of three difficult occurrences for four different words from the Bentham data set is reported. The increase compared QE qith WS as well as CR, ML and ML-CR with QE.

Word	**WS**	**QE**	**CR**	**ML**	**ML-CR**
aforesaid	0,643265	0,960291	**0,969982**	0,520125	0,571602
amount	0,387363	0,654970	**0,716165**	0,518282	0,713422
intended	0,596667	**0,633333**	**0,633333**	0,574102	**0,633333**
Jeremy	0,333333	**0,760000**	**0,760000**	**0,760000**	**0,760000**
mean	0,490157	0,752148	**0,769870**	0,593127	0,669589
Increase		53,45%	**2,36%**	-21,14%	-10,98%

6 Discussion

The proof of concept showed that the idea generally works very well (the mAP generally improved by 12.30%) even for heavily contaminated sets, such as in the case of the experiment with MNIST where only 10% in the ranked list are true positives. Furthermore, the experiment showed that the proposed consensus ranking still works rather well even when some of the items used for the CR are false positives (in this case, on average 17.14%). Hence, a few false positives does not, in general, make the algorithm fail.

The increases presented in the tables for the Esposalles and the Bentham data sets in section 5, using CR after QE might seem rather low. However, they are important as they reorder the rank so that it will be easier for a human to quickly distinguish true positives fro false positives by a visual inspection, as shown in the presented images in section 5. This will be a useful feature both for semi-automatic transcription [HCV18] as well as for searching non transcribed document images [HCVA19].

One advantage with CR is that it helps to purge what must be false positives to a high degree of certainty. By inspecting Figure 3 and the ranked list in Figure 4 and Figure 5 one can see how CR (the blue curve) suddenly drops rapidly and in the end of that drop there are no true positives anymore. Just a few false positives are found in the drop or above. This drop can be quite obvious as in the case for "reber" or more subtle as for "Barca". Therefore, it is probably better to set a fixed threshold for purging than trying to locate the "drop". It seems like the threshold generally could be set equal to τ or a bit below. Being used in a transcription or a search tool, the threshold could be a parameter set using a slider by the user.

CR will not substantially increase the mAP when there are very few true positives. This is generally not a big problem, since the retrieved ranked list will consequently be very short anyway. For instance "intended" has only 10 occurrences in the Bentham data set that was used herein (the other words have between 12 and 25 occurrences.) and the mAP do not improve as seen in Table 4. In fact, it is already good and could not be improved further since QE found 9 out of 10 occurrences and ranked them on top.

Overall, the time it takes to perform QE is many times longer the time it takes to perform CR, due to the word spotting search per page, and the QE that performs such search on several words per page. Nevertheless, the time it takes to perform CR is not negligible, especially not for when the final list produced by QE is long, i.e. several hundred words. QE always ends by performing one ranking using the original query word to produce a final ranked list. CR then performs up to n such rankings and thus adds n times more execution. The fusion itself is almost negligible since it is just averaging scores, compared to the work it takes to match images to compute the scores in the first place.

One disappointment is that the ML-CR generally did not perform better than CR for the difficult word images. Of course the tests performed here are quite limited and running on larger data sets and also using different number for the top n words might turn out to give different results. Nevertheless, ML generally performs much worse than CR because of the tight set thresholds in QE that will lower the number of false positives and hence decrease the visual load for the user.

7 Conclusions

The ranking of words found by word spotting is important for tasks like fast transcription or simple searches. The user do not want to be bothered with the many false positives that often are retrieved. Therefore, ranking is very important, pushing the false positives to the end of the list. Moreover, purging the list automatically from the obvious false positives also helps the user. The proposed Consensus Ranking helps the user with both of these aspects. The idea is simply to take the top n words in the list ranked by matching to the query word, and re-score the list using each of these words to re-score again, giving n new lists that can be fused together to obtain a better ranked list. By averaging the scores, a majority of the false positives will generally obtain lower scores than the threshold used for the word spotting itself, and can therefore be purged.

References

[AZ12] Relja Arandjelović and Andrew Zisserman. Multiple queries for large scale specific object retrieval. In *British Machine Vision Conference*, 2012.

[BC17] Rodger Benham and J. Shane Culpepper. Risk-reward trade-offs in rank fusion. In *Proceedings of the 22Nd Australasian Document Computing Symposium*, ADCS 2017, pages 1:1–1:8, New York, NY, USA, 2017. ACM.

[BKFS95] N. J. Belkin, P. Kantor, E. A. Fox, and J. A. Shaw. Combining the evidence of multiple query representations for information retrieval. *Inf. Process. Manage.*, 31(3):431–448, May 1995.

[BP09] Narayan L. Bhamidipati and Sankar K. Pal. Comparing scores intended for ranking. *IEEE Trans. on Knowl. and Data Eng.*, 21(1):21–34, jan 2009.

[Bur10] Chris J.C. Burges. From ranknet to lambdarank to lambdamart: An overview. Technical Report MSR-TR-2010-82, Microsoft Research, June 2010.

[GSGN17] Angelos P Giotis, Giorgos Sfikas, Basilis Gatos, and Christophoros Nikou. A survey of document image word spotting techniques. *Pattern Recognition*, 68:310–332, 2017.

[Har92] Donna Harman. Relevance feedback and other query modification techniques. In William B. Frakes and Ricardo Baeza-Yates, editors, *Information Retrieval*, pages 241–263. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

[HCV18] Anders Hast, Per Cullhed, and Ekta Vats. \text - text extractor tool for handwritten document transcription and annotation. In *Digital Libraries and Multimedia Archives - 14th Italian Research Conference on Digital Libraries, IRCDL 2018, Udine, Italy, January 25-26, 2018, Proceedings*, pages 81–92, 2018.

[HCVA19] Anders Hast, Per Cullhed, Ekta Vats, and Matteo Abrate. Making large collections of handwritten material easily accessible and searchable. In *Digital Libraries: Supporting Open Science - 15th Italian Research Conference on Digital Libraries, IRCDL 2019, Pisa, Italy, January 31 - February 1, 2019, Proceedings*, pages 18–28, 2019.

[HF16] Anders Hast and Alicia Fornés. A segmentation-free handwritten word spotting approach by relaxed feature matching. In *Document Analysis Systems (DAS), 2016 12th IAPR Workshop on*, pages 150–155. IEEE, 2016.

[HLV19] Anders Hast, Mats Lind, and Ekta Vats. Embedded prototype subspace classification : A subspace learning framework. In Percannella G. Vento M., editor, *Computer Analysis of Images and Patterns: CAIP2019*, Lecture Notes in Computer Science, 2019.

[HV18] Anders Hast and Ekta Vats. An intelligent user interface for efficient semi-automatic transcription of historical handwritten documents. In *Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion*, IUI '18 Companion, pages 48:1–48:2, New York, NY, USA, 2018. ACM.

[LCB10] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010.

[LKG12] G. Louloudis, A. L. Kesidis, and B. Gatos. Efficient word retrieval using a multiple ranking combination scheme. In *Proceedings of the 2012 10th IAPR International Workshop on Document Analysis Systems*, DAS '12, pages 379–383, Washington, DC, USA, 2012. IEEE Computer Society.

[LLZ09] Hang Li, Tie-Yan Liu, and ChengXiang Zhai. Learning to rank for information retrieval (lr4ir 2009). *SIGIR Forum*, 43(2):41–45, dec 2009.

[LRF+12] J. Llados, M. Rusinol, A. Fornes, D. Fernandez, and A. Dutta. On the influence of word representations for handwritten word spotting in historical documents. *International Journal of Pattern Recognition and Artificial Intelligence*, 26(05), 2012.

[MS05] Kieran McDonald and Alan F. Smeaton. A comparison of score, rank and probability-based fusion methods for video shot retrieval. In *Proceedings of the 4th International Conference on Image and Video Retrieval*, CIVR'05, pages 61–70, Berlin, Heidelberg, 2005. Springer-Verlag.

[MTW11] Martin Moyle, Justin Tonra, and Valerie Wallace. Manuscript transcription by crowdsourcing: Transcribe bentham. *Liber Quarterly*, 20(3-4), 2011.

[PnSMVG10] Sebastián Peña Saldarriaga, Emmanuel Morin, and Christian Viard-Gaudin. Ranking fusion methods applied to on-line handwriting information retrieval. In *Proceedings of the 32Nd European Conference on Advances in Information Retrieval*, ECIR'2010, pages 253–264, Berlin, Heidelberg, 2010. Springer-Verlag.

[Pow07] David Powers. Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation. Technical Report Technical Report SIE-07-001, School of Informatics and Engineering Flinders University, Adelaide, Australia, December 2007.

[RFS+13] VeróNica Romero, Alicia Fornés, NicoláS Serrano, Joan Andreu Sánchez, Alejandro H Toselli, Volkmar Frinken, Enrique Vidal, and Josep Lladós. The ESPOSALLES database: An ancient marriage license corpus for off-line handwriting recognition. *Pattern Recognition*, 46(6):1658–1669, 2013.

[RnL14] Marçal Rusiñol and Josep Lladós. Boosting the handwritten word spotting experience by including the user in the loop. *Pattern Recogn.*, 47(3):1063–1072, mar 2014.

[SF95] Joseph A. Shaw and Edward A. Fox. Combination of multiple searches. In Donna K. Harman, editor, *Overview of the third Text REtreival Conference (TREC-3)*, pages 105–108, 1995.

[Sme98] Alan F. Smeaton. Independence of contributing retrieval strategies in data fusion for effective information retrieval. In *Proceedings of the 20th Annual BCS-IRSG Conference on Information Retrieval Research*, IRSG'98, pages 12–12, Swindon, UK, 1998. BCS Learning & Development Ltd.

[VH17] Ekta Vats and Anders Hast. On-the-fly historical handwritten text annotation. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, volume 8, pages 10–14. IEEE, 2017.

[VHF19] Ekta Vats, Anders Hast, and Alicia Fornearch. Training-free and segmentation-free word spotting using feature matching and query expansion. In *Proc. 15th International Conference on Document Analysis and Recognition*, 2019.

[VHS17] Ekta Vats, Anders Hast, and Prashant Singh. Automatic document image binarization using bayesian optimization. In *Proceedings of the 4th International Workshop on Historical Document Imaging and Processing*, pages 89–94. ACM, 2017.

[WCB06] Shengli Wu, Fabio Crestani, and Yaxin Bi. Evaluating score normalization methods in data fusion. In Hwee Tou Ng, Mun-Kew Leong, Min-Yen Kan, and Donghong Ji, editors, *Information Retrieval Technology*, pages 642–648, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[WGS15] H. Wei, G. Gao, and X. Su. A multiple instances approach to improving keyword spotting on historical mongolian document images. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 121–125, Aug 2015.

[WH18] Markus Wegmann and Andreas Henrich. Search for an appropriate journal - in depth evaluation of data fusion techniques. In Rainer Gemulla, Simone Paolo Ponzetto, Christian Bizer, Margret Keuper, and Heiner Stuckenschmidt, editors, *Proceedings of the Conference "Lernen, Wissen, Daten, Analysen", LWDA 2018, Mannheim, Germany, August 22-24, 2018.*, volume 2191 of *CEUR Workshop Proceedings*, pages 343–354. CEUR-WS.org, 2018.

[Wu12] Shengli Wu. *Ranking-Based Fusion*, pages 135–147. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.