# Machine learning methods for results merging in patent retrieval

Vasileios Stamatis, Michail Salampasis and Konstantinos Diamantaras
*Department of Information and Electronic Engineering, International Hellenic University, Thessaloniki, Greece*

Received 16 June 2021
Revised 2 February 2022
Accepted 7 March 2022

## Abstract

**Purpose** – In federated search, a query is sent simultaneously to multiple resources and each one of them returns a list of results. These lists are merged into a single list using the results merging process. In this work, the authors apply machine learning methods for results merging in federated patent search. Even though several methods for results merging have been developed, none of them were tested on patent data nor considered several machine learning models. Thus, the authors experiment with state-of-the-art methods using patent data and they propose two new methods for results merging that use machine learning models.
**Design/methodology/approach** – The methods are based on a centralized index containing samples of documents from all the remote resources, and they implement machine learning models to estimate comparable scores for the documents retrieved by different resources. The authors examine the new methods in cooperative and uncooperative settings where document scores from the remote search engines are available and not, respectively. In uncooperative environments, they propose two methods for assigning document scores.
**Findings** – The effectiveness of the new results merging methods was measured against state-of-the-art models and found to be superior to them in many cases with significant improvements. The random forest model achieves the best results in comparison to all other models and presents new insights for the results merging problem.
**Originality/value** – In this article the authors prove that machine learning models can substitute other standard methods and models that used for results merging for many years. Our methods outperformed state-of-the-art estimation methods for results merging, and they proved that they are more effective for federated patent search.

**Keywords** Results merging, Patent retrieval, Machine learning, Federated search, Distributed information retrieval

**Paper type** Technical paper

## 1. Introduction

Patent and other innovation-related documents can be found in patent offices, online datasets and resources that typically must be searched using various patent search systems and other online services such as Espacenet, Google Patents, Bibliographic Search and many more (Salampasis, 2017). Patent search is considered as a subfield of information retrieval (IR), and its goal is to develop systems that effectively and efficiently retrieve the most relevant patent-related documents in response to a query request (Clarke, 2018). From an information task perspective, patent retrieval tasks are quite often recall-oriented (Lupu and Hanbury, 2013);



*Funding:* This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No: 860721 (DoSSIER Project, https://dossier-project.eu/).

Data Technologies and Applications
Emerald Publishing Limited
2514-9288
DOI 10.1108/DTA-06-2021-0156

therefore, the retrieval of all the patent documents related to a patent application is crucially important, otherwise, missing a single patent might have a significant economic impact (Khode and Jambhorkar, 2019; Mahdabi *et al.*, 2013). Thus, in professional search, it is vital to search efficiently and effectively in all the potentially distributed resources containing patents or other patent-related data.

Federated search (FS) aims to provide a solution to this problem. FS systems implement a Distributed IR (DIR) scenario that permits the simultaneous search of multiple resources that may also be physically distributed. For example, a user who wants to implement a typical prior art search to investigate the patentability of an idea must search in multiple sources. Instead of searching every single source one by one, FS makes this process more efficient by searching multiple sources simultaneously. A hands-on example of such a system is PerFedPat (Salampasis and Hanbury, 2014). PerFedPat offers core services and operations for being able to search multiple online patent resources. It provides a unified single-point search access while hiding complexity from the end-user who uses a common query tool for querying all patent datasets at the same time.

The FS procedure is divided into three different sub-processes (Salampasis, 2017). First is the source representation, then the source selection and eventually the results merging process. The source representation process is used to get statistics and create approximations about the contents of the federated resources. The source selection process is used to determine the resources that will be queried in the retrieval process. Finally, the last step is the results merging process in which the results from all the different sources are merged and returned to the user. Our work focuses on the last phase of the DIR procedure. The results merging process is a critical stage in the DIR procedure as has been early shown by relevant research (Callan, 2002; Callan *et al.*, 1995; Craswell *et al.*, 1999; Si and Callan, 2003), in the sense that even if the other sub-processes work satisfactorily if the results merging sub-process does not operate effectively, the overall effectiveness of the system will deteriorate.

IR research has been studying the results merging problem for many years. Several methods have been developed that solve the results merging problem, but they were not explicitly designed for the patent domain, e.g. CORI (Callan *et al.*, 1995), semi-supervised learning (SSL) (Si and Callan, 2003) and sample-agglomerate fitting estimate (SAFE) (Shokouhi and Zobel, 2009). The vital prerequisite is that a single missed prior art can cause significant economic loss, which is why patent search is usually recall-oriented. For example, in SSL and SAFE, the main target was high precision, and therefore they used precision-oriented metrics in their experiments.

In the work presented in this article, the authors extend our initial idea presented in Stamatis and Salampasis (2020), and more specifically, they implement state-of-the-art (SotA) results merging algorithms designed for FS and test them on patent data. Additionally, they propose Machine Learning Models for Results Merging (MLRM), two new methods for results merging that, similarly to the existing methods, they estimate comparable scores for documents from different resources based on a centralized sample index. This centralized sample index is created using samples of documents from all the different resources so the scores in it are directly comparable no matter the initial origin of the documents. Thus, the goal is to use the local collection-specific scores from the different resources as features for Machine learning (ML) models to estimate these comparable scores. To do this prediction, usually, linear regression has been used (Si and Callan, 2003; Paltoglou *et al.*, 2007), but this approach assumes that there is a linear mapping between the scores of the documents in the resources and their respective scores in the centralized index. When SSL and SAFE were proposed, linear regression was more like a statistical process to linearly model the relationship between two or more variables.

Nowadays, linear regression falls into the ML field. So, other ML models could be used instead of linear regression that might fit the data better, especially when the relationship between the variables is not linear. The authors address this need by applying ML models other than linear regression to examine if they can better map the resources' scores to centralized scores. Our contribution can be summarized as follows.

- The authors implement SotA FS results merging algorithms and test them on patent data.
- The authors propose new methods for results merging that uses predictive ML models and conduct experiments to evaluate them.
- The authors examine two architectures and several ML models and compare the results with other well-performing methods. They discuss which model explains better the correlation between the documents' scores in a ranking and their respective centralized scores.
- The authors examine our models in cooperative and uncooperative environments.
- The authors compare two methods that solve the lack of relevancy scores in uncooperative environments.

The rest of this article is organized in the following manner. In the next section, the prior work is reviewed. Following that, the methodology is presented as well as the environment used for the experiments. Next, the results and findings are analyzed and discussed. Finally, the authors conclude the article by presenting ideas for further development.

## 2. Prior work

The results merging problem appeared in research many years ago. The first work about results merging was presented by Voorhees *et al.* (1995). However, it has been studied as a general DIR problem and not in the specific context of the patent domain. While IR research has gained significant achievements in research and development, professional search and, more specifically, the patent industry, is considered a more traditional and complex domain, therefore a more challenging area (Shalaby and Zadrozny, 2019).

Methodologically speaking, results merging can be categorized into estimation methods, download methods and hybrid methods. In estimation methods, methodologies like regression, weighted score merging, etc. are used to calculate the relevance of the documents returned from the heterogenous remote collections (Callan *et al.*, 1995; Si and Callan, 2003; Shokouhi and Zobel, 2009). The download methods, download all the documents returned and re-calculate their relevance to the query locally (Craswell *et al.*, 1999; Hung, 2019). Finally, hybrid methods are a combination of estimation and download methods (Paltoglou *et al.*, 2008). Most of the research in the field is focused on estimation methods because of the high communication cost to download many remote documents. The main problem that results merging solves is that the scores of the returned documents from the different resources are not directly comparable or even unknown. In most cases, the resources return only ranked lists of documents with no scores associated with each document.

One of the oldest, most widely used and robust estimation method is the collection inference retrieval network CORI (Callan *et al.*, 1995). CORI uses a weighted score merging scheme. It uses a linear combination of the score of the document returned by the collection and the source selection score and combines both using a simple heuristic formula. It finally normalizes the collection-specific scores to produce global comparable scores.

Another estimation algorithm is the SSL algorithm (Si and Callan, 2003) which is based on linear regression. The SSL algorithm proposed by Si and Callan (2003) applies linear regression to assign the local collection-specific scores to the global comparable scores. To achieve that, the algorithm functions on the common documents returned every time, between a collection and a centralized index created from samples retrieved from all the available collections.

SAFE was the next significant estimation results merging algorithm proposed by Shokouhi and Zobel (2009). The SAFE algorithm was designed to function in uncooperative environments as it does not rely on the overlapped documents between the collections. SAFE is based on the principle that the results of the sampled documents for each query are a sub-ranking of the original collection, so this sub-ranking can be used to conduct curve fitting in order to predict the original scores.

Based on SAFE, other algorithms and extensions of it appeared in research. He *et al.* (2011) proposed Weighted Curve Fitting (WCF) for results merging. This method is based on SAFE, and they add two more parameters to it. First, they give more value to the documents with the true ranks and lower the value to the documents with the estimated ranks. The second is that they penalize the weights of the lower documents in the ranked lists for regression. They found WCF to perform better than SAFE using the precision metric, but they didn't use other metrics. Hong and Si (2012) identified that existing results merging algorithms do not fully address the heterogeneity of information sources in FS because they use a single centralized retrieval algorithm. Due to that, the different retrieval models, statistics, etc., that remote search engines might have are not considered. Thus they proposed the Mixture of Retrieval Models (MoRM) for results merging, which is actually another extension of the SAFE algorithm. In their framework, they use a centralized index created from sampled documents that contain the centralized scores, and the goal is to map the ranks of the ranked lists to comparable scores. They used multiple retrieval algorithms to query the centralized index and get multiple ranked lists. Then for each ranked list, MoRM will try to map the ranks of the documents from the federated sources to centralized comparable scores. They recorded improvements over SAFE, but they only used precision@k as their evaluation metric.

In terms of download methods, Hung (2019) proposed a technique in which the best documents are downloaded to re-rank and create the final merged list. He used ML and genetic programming to re-rank the final merged results. While download methods seem to perform better than estimation approaches in the context tested by in Craswell *et al.* (1999), they have essential disadvantages such as increased computation, download time and bandwidth overhead during the retrieval process.

Paltoglou *et al.* (2008) proposed a hybrid method that combines download and estimation methods. More specifically it downloads a limited number of documents, and based on them, it trains a linear regression model for calculating the relevance of the rest documents. The results showed that this method achieved a good balance between the effectiveness and efficiency of the download and estimation approaches respectively.

Lee *et al.* (2015) presented an optimization framework for results merging. They extended the λ-merge method presented in Sheldon *et al.* (2011) by adding the extra component of the vertical quality in order to conduct results merging. Vijaya *et al.* (2016) exploited the unique links, and they created a scores merging method for meta-search engines using neural networks.

Besides academic research, patents about results merging processes have been developed. Taylor *et al.* (2016) published a patent about a ML process for conducting results merging. One more patent was published by Mao *et al.* (2004) which uses the scores assigned to the lists and the documents to complete the final merging.

## 3. Methodology

### *3.1 MLRM method*

For merging the results, the authors also apply the method of the Centralized Sample Index (CSI) as used in other algorithms (SSL and SAFE). To explain the method, let's assume that we have *N* different federated resources. The first step is to create a CSI which consists of sampled documents from all the different resources. To get these samples, the authors implement the well-known method of query-based sampling (Callan and Connell, 2001). Query-based sampling is applied to create representations of the federated resources in order to approximate the statistics from these resources when they are not available. According to query-based sampling, the authors send random queries to each resource and download the first four documents downloaded until they reach a limit in the number of sampled documents retrieved. Using the same method, the authors create representations of all the remote resources. Each representation set consists of around 300 documents. When fewer documents exist in a collection, all the documents are used.

In the experimental process, when a query is submitted, it is sent to the M-selected resources and the centralized index. The M-resources will return M-ranked lists of documents, and the centralized index will return a list of documents along with scores as we query the centralized index locally and scores are always available. The common documents between the resources and the centralized index (overlapped documents) are utilized to train ML models that use the local resource-specific scores to predict the scores assigned by the CSI. The scores from the CSI are comparable as the sampled documents from all the remote resources co-exist in it (global scores). In other words, the models are trained to convert the resource-specific scores that are noncomparable to global scores that are comparable. Then, the final merging can be calculated. Figure 1 summarizes the process of the results merging in detail. Based on the above scenario, all the steps for implementing end-to-end FS using MLRM are the following:

(1) Representations sets are created for all the different resources using query-based sampling and all these representations sets are combined to create the CSI.

(2) For every query, the query is routed to the source selection method, to calculate relevancy scores for each resource.

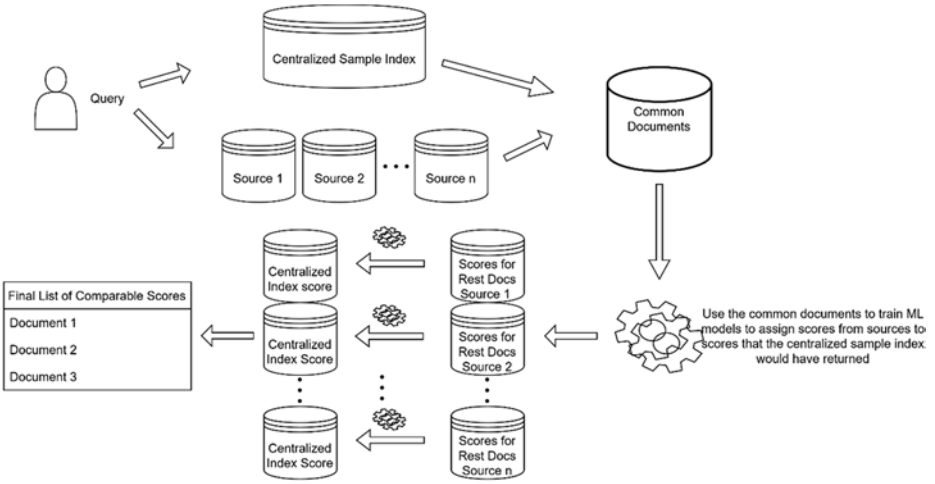


**Figure 1.** Results merging workflow (MLRM architecture)

(3) Each query is then submitted to the top 20 selected resources, and we retrieve 100 results per resource. At the same time, the query is submitted to the CSI, and we retrieve 1,000 results.

(4) The overlapped (common) documents from the resources and the CSI are used to train the ML models for transforming the local resource-specific scores to the comparable scores of the CSI.

(5) The trained ML models are then applied to predict the comparable scores of the CSI for the rest non-common documents to merge the results from the remote resources into a single list of documents.

The authors experiment with two methods that implement ML models for merging the results. The process described above is common for both approaches. The main difference between the two method is how the ML model predicts the global scores. The first method uses the overlapped documents from the resources and the CSI and trains one different model per resource. The trained models are then used to calculate the global relevance scores for the rest non-common documents returned by the respective resource. More specifically, for the training, it uses as single input the local score of the document, and the target is the global score of the CSI. Eventually, the trained model is applied and accepts as inputs the local scores for the rest documents and predicts the global CSI scores. In that way, the predicted scores are comparable, and the authors can conduct the final merging. One of the ML models the authors implement is linear regression, and this approach is actually the SSL algorithm presented in Si and Callan (2003). Furthermore, they use polynomial regression, and more specifically, they apply the $x^2$ and $x^3$ polynomial features. The authors also implement random forest, support vector machine (SVM) and decision tree models. These models are called multiple models (MMs) because one model per resource is created.

The second method uses the same ML model for all the resources for a single query. Thus, the models in this approach are called global models (GMs) and they are trained for every query. These GMs accept as input all the documents' scores as retrieved from all the resources. For instance, suppose that a query will be routed to 10 resources, then there will be 10 inputs for the algorithm which represent the scores of the documents as retrieved from the resources. If a document is not retrieved by some resources, the scores will be zero. The authors implement this architecture because the algorithms take into account the case when the same documents are retrieved by more than one resource. For example, if document D was returned by the first and fifth sources with scores of 300 and 150, respectively, and the centralized index returned the same document with a score of 350, then the input for the training would be as given in Table I:

In that way, the common documents between different resources are also considered which is an additional information. A document retrieved from multiple resources might be more relevant to the respective query. The ML models implemented as GMs are random forest, SVM, decision tree and deep neural network (DNN).

### *3.2 Experimental environment*

The authors conduct the experiments using two different environments. They first use the cooperative environment in which the documents are returned along with scores. Also, collection

**Table I.** Input for the global model

| | 1st | 2nd | 3rd | 4th | 5th | 6th | 7th | 8th | 9th | 10th | Centralized |
|---|---|---|---|---|---|---|---|---|---|---|---|
| D | 300 | 0 | 0 | 0 | 150 | 0 | 0 | 0 | 0 | 0 | 350 |

statistics such as the number of documents, term frequencies, etc., from the different collections are available. While cooperative settings are very rare in the real world, the authors choose to use this environment to investigate the effectiveness of the methods unbiased from assigning artificial local scores. Additionally, the authors conduct the experiments assuming an uncooperative environment which is the most realistic scenario, and the returned results from remote resources are just ranked lists of documents with no scores associated with them. For example, Espacenet (https://worldwide.espacenet.com/patent/), which provides a search interface for patent search, falls into the category of uncooperative environment because we don't have access to its collection statistics like term frequencies, etc., and also the results returned as a ranked list of relevant patent documents and there are no scores associated with them. The uncooperative environment is applied, if not in all, in most search engines nowadays, whether general search engines (Google, Bing, etc.) or domain-specific search engines (Espacenet, WIPO, etc.).

Furthermore, the authors experiment with two theoretical conditions, the optimal and the random scenarios for results merging, each representing respectively the upper and the lower baseline. The optimal method cannot happen in real settings, but they can have important insights when using it. The optimal scenario can be thought of as optimal input for the results merging process and it could be implemented retrospectively only if they know in advance the distribution of relevant documents to remote resources. Instead of performing source selection, the authors route the query only to the search engines containing relevant documents. This method provides results unbiased from the source selection as many times results merging algorithms can perform weak due to poor source selection rather than poor results merging performance. On the other hand, the random scenario can be thought of as the lower limit in the results merging, and the authors performed it again by specifying the relevant sources to be searched but they conduct the final merging randomly. Comparing the results with the random and the optimal merging, we can measure the performance value of the new methods created for results merging relative to these two lower and upper baselines.

For the source selection process in our experiments, the authors use the CORI algorithm. Furthermore, the authors investigated the parameter of the total number of remote collections to submit the query i.e. how many of all the potential resources will be requested to return their results for each query and use them in the merging stage. The authors found 20 to be the optimal number of remote collections. Also, they test retrieving 100, 200 and 500 results per collection. Finally, they run the main experiment with 100 results per collection as this produces optimal results in terms of efficiency and effectiveness.

### *3.3 Dataset*

The experiments reported in this article are based on the CLEF-IP standard test collection (Piroi *et al.*, 2011), which is an extract from the more extensive matrixware research collection (MAREC) collection (MAREC, Online), and it consists of 3,118,088 patent documents pertaining to 1,768,641 patents (i.e. two or more patent documents relate to one patent). The dataset was cleaned, preprocessed and converted to standard TREC format. Also, all the different kinds of patent documents, i.e. A1, B2, etc., were merged into a single document. The fields used were invention title, inventor and applicant's information, abstract, the first 500 words of description and claims. Patent documents are filled in English, French or German languages because of European Patent Office requirements. The authors chose to have all the text in English so, all the non-English text which may exist in patents was translated using Google's translator.

The CLEF-IP standard test collection represents a single source of patent documents. Thus, to create a federated environment, similar to the work done by Salampasis *et al.* (2012), the authors use the IPC codes of the CLEF-IP standard test collection at the subgroup level (level 3), where each IPC code will represent a separate resource containing the patent documents that have been tagged with the specific IPC code. This results in 632 different federated resources. The IPC system represents a language-independent taxonomy that is internationally accepted and used for classifying and organizing patent documents. So, IPC codes are language-independent symbols assigned to patent documents according to the technical area they belong to Giachanou *et al.* (2015). There are about 71,000 different IPC codes in the CLEF-IP standard test collection organized into a five-level + hierarchical system. Note that since a patent can belong to more than one IPC codes, i.e. resources have overlapped documents, something which differentiates our work from many other results merging algorithms that usually assume non-overlapping disjoint collections. The new data consists of 2,438,765 patent documents including the overlapped documents. The authors produce all the indices using Anserini (Yang *et al.*, 2018), a Lucene-based toolkit built with an orientation in IR research.

The queries used in the experiments were produced from the 3,973 topics of the CLEF-IP 2011 evaluation campaign. There are topics in three languages (English, French and German) so to create the queries, the authors only use the first 300 English topics. The authors choose the first 300 English topics to reduce the computing power needed and also to be consistent with the research works that used CLEF-IP collection and the first 300 English topics (Giachanou and Salampasis, 2014; Giachanou *et al.*, 2015). Each query consists of a maximum of 1,000 words produced from the title, abstract, the first 500 words of the description and the claims. The authors use the mean average precision (MAP), RECALL and patent retrieval evaluation score (PRES) scores as the metrics for the experiments. PRES is a metric focusing on the evaluation of recall-oriented IR systems (Walid and Gareth, 2010).

### *3.4 Lack of relevancy scores*

When retrieval takes place in an uncooperative environment, local document scores are not returned therefore, only rankings are available. The authors overcome the lack of relevancy document scores problem using two methods.

The first method is based on assigning artificial scores linearly to the documents according to their rank in the list. The authors score the first document with 0.6 and descending by even steps, they score the last with 0.4. This intuitive scoring method is chosen as it has been shown to work well with CORI in the literature (Avrahami *et al.*, 2006). The authors assign the artificial scores $A(d_i)$ to the documents according to the following equation:

$$A(d_{i,j}) = 0.6 - \text{step} \times (i-1), \quad i = 2, 3, \ldots, n, \; j = 1, 2, \ldots, 632, \tag{1}$$

$$\text{step} = \frac{0.6 - 0.4}{n-1},$$

Where $d_{i,j}$ represents the $i$th document in the initial rank from the collection $j$, i.e. $d_{1,j} = 1$, $d_{2,j} = 2$, etc. for every $j = 1, 2, \ldots, 632$. In total, there are 632 different collections, and this is why $j$ is between 1 and 632. The variable $n$ represents the number of documents retrieved by each collection. For all the collections $j = 1, 2, \ldots, 632$, we have $A(d_{1,j}) = 0.6$ and $A(d_{n,j}) = 0.4$ as these are the first and the last scores, respectively.

The second method is similar to the first, but we use a weighted scheme for assigning local scores. More specifically, the authors assign the same artificial scores to the documents in the same way described in the previous paragraphs, but they also use the source selection score as input to the algorithms. Considering the source selection score in the results merging phase could reduce the bias associated with the rank of the documents in different, more, and less relevant resources. For instance, in a more relevant resource to a query, the 10th document might be more relevant than the 4th document of another less relevant resource. The final score $S(d_{i,j})$ is calculated as follows:

$$S(d_{i,j}) = A(d_{i,j}) \times C_{j,i}, \quad i = 1, 2, \ldots, n, j = 1, 2, \ldots, 632, \tag{2}$$

where $A(d_{i,j})$ is the artificial score calculated using formula (1), and $C_j$ is the score of the resource calculated during the source selection phase.

*3.5 Baselines*

For comparison reasons, the authors implement the state-of-the-art methods CORI, SSL and SAFE, representing the most important phases in the evolution of the results merging algorithms. The results merging algorithms were not specifically targetting patent search. Also, metrics such as recall, which are vital for patent search have not been examined in many experiments and studies. Thus, to investigate and compare the value of our proposed methods, the authors locally implement all the state-of-the-art baselines and run the experiments using the CLEF-IP dataset. Furthermore, the authors conduct statistical significance tests to examine the significance of the results.

*3.5.1 CORI.* CORI is based on a weighted score merging scheme. The scores returned from the resources will be transformed into normalized comparable scores $D'$. CORI uses a linear combination of resources and collection selection scores using the formula below.

$$D' = \frac{D + 0.4 \times D \times C'}{1.4}$$

where $D$ is the document's score returned by the collection, and $C'$ is the normalized collection score calculated as:

$$C' = \frac{C - C_{\min}}{C_{\max} - C_{\min}},$$

where $C$ is the collection selection score.

*3.5.2 SSL.* SSL is based on the CSI approach explained in Section 3.2. For every query, It uses the overlapped documents between the resources and the CSI, and it trains linear regression models to map the local noncomparable scores to the comparable global scores of the centralized index.

*3.5.3 SAFE.* SAFE again uses the CSI and it maps the ranks of the documents in the collection to the CSI scores using linear regression. SAFE uses the overlapped documents between the CSI and the resources to identify the ranks in the original resources. If there are no overlapped documents, it estimates the ranks of documents in the original resources.

*3.5.4 Centralized.* The authors also run the experiments using the centralized approach. For this approach, they concatenate all the resources to a single index and submit the queries directly to it. The centralized approach can be thought of as the other edge of FS and requires neither results merging nor other FS technics. The authors implemented this system to compare the federated and the centralized systems.

*3.5.5 Random merging.* Lastly, the authors conduct random merging to get the lowest baseline and better understand the added value of the methods.

## 4. Results

The base experiment was executed three times. First, the authors set a cooperative environment where document scores are returned. Second, they assume an uncooperative environment where they assign artificial scores using formula (1), and third, they assume an uncooperative environment again but they assign artificial scores using formula (2). In each experiment, the authors compare the GMs and the MMs, using the normal scores as well as the scores from the random and optimal scenarios explained in Section 3.2. Also, the authors compare with the baselines described in Section 3.5. To achieve a more complete comparison, they conduct statistical tests to examine the significance of the results compared to the very robust models CORI, SSL and SAFE. To test statistical significance, they use the Student's *t*-test when the equality of the variance is achieved and Welch's *t*-test if this condition does not meet. In the results, the asterisk (*), the dagger (†) and the double-dagger (‡) represent statistical significance at the 90%, 95% and 99% confidence intervals, respectively.

### *4.1 Cooperative environment*

In the cooperative environment, for the MMs, the authors use random forest, support vector regression (SVR), decision tree and polynomial regression. All the models are used with their default values. Table II below summarizes the results. Comparing the two methods of the GMs and the MMs, the best performance gained from MMs. In the MMs, the random forest produces the best results following the decision tree. These two methods also perform

**Table II.** Results in the cooperative environment

| | | MAP@100 | | PRES@100 | | RECALL@100 | |
|---|---|---|---|---|---|---|---|
| | | Realistic scenario | Optimal scenario | Original scenario | Optimal scenario | Original scenario | |
| MMs | Random forest | 0.0777 | 0.1078 | 0.3358 | 0.4369 | 0.3468 | 0.4514 |
| | SVR | 0.0612 | 0.0780 | 0.2679 | 0.3372 | 0.2779 | 0.3501 |
| | Decision tree | 0.0745 | 0.0967 | 0.3283 | 0.4333 | 0.3391 | 0.4481 |
| | Polynomial $x^2$ | 0.0454 | 0.0641 | 0.2036 | 0.2578 | 0.2105 | 0.2663 |
| | Polynomial $x^3$ | 0.0299 | 0.0414 | 0.1243 | 0.2237 | 0.1287 | 0.2329 |
| GMS | Random forest | 0.0465 | 0.0518 | 0.2406 | 0.2679 | 0.2483 | 0.2764 |
| | SVR | 0.0218 | 0.0284 | 0.0788 | 0.1966 | 0.0820 | 0.2063 |
| | Decision tree | 0.0414 | 0.0446 | 0.2315 | 0.2601 | 0.2391 | 0.2688 |
| | Linear regression | 0.0517 | 0.0714 | 0.2401 | 0.2752 | 0.2474 | 0.2825 |
| | DNN | 0.0693 | 0.0815 | 0.2353 | 0.2741 | 0.2416 | 0.2816 |
| | CORI | 0.0650 | 0.0820 | 0.2102 | 0.2806 | 0.2161 | 0.2889 |
| | SSL | 0.0725 | 0.0819 | 0.2464 | 0.2823 | 0.2528 | 0.2902 |
| | SAFE | 0.0606 | 0.0873 | 0.2200 | 0.2829 | 0.2262 | 0.2903 |
| | Centralized | 0.0793 | | 0.2592 | | 0.2660 | |
| | Random merging | 0.0141 | | 0.0979 | | 0.1833 | |

better than the algorithms SSL, CORI and SAFE in all three metrics. Most notably, the results are statistically significant at the 99% confidence interval, and the improvements are up to +60 per cent. Additionally, polynomial regression does not perform better than SSL, meaning that linear mapping between the resources' documents scores and centralized documents scores is better than polynomial mapping.

For the GMs, the authors use random forest, SVR, decision tree, linear regression and a DNN. All the ML models were used with their default values. The authors create the DNN with four hidden layers, using the "Adam" optimizer with a learning rate of 0.01, and the mean squared error for the loss function. The best score from GMs in terms of MAP is from the DNN. The best PRES and RECALL scores are from the random forest. All the GMs achieve better PRES and RECALL results than CORI and SAFE except SVR. Especially for the random forest, the results are significant at the 90% confidence interval. This is important because the patent industry is recall-oriented, as a single missing patent document may have a substantial economic impact. SSL achieves higher scores than all GMs. The authors observe that SVR performs relatively poorly in GMs. This might be connected with the hyperplanes that SVR creates and the big differences in document scores that different resources might assign.

In addition, SSL performs better than CORI and SAFE. In summary, looking at all three metrics, random forest from the MMs gives the best performance compared to all the models followed by the decision tree. In Table III, the authors choose the best models in terms of RECALL, from the MMs and the GMs and summarize the proportional differences compared to the baseline algorithms CORI, SSL and SAFE. The centralized approach gives the highest MAP score in comparison to all FS methods. MMs random forest, MMs decision tree and MMs SVR achieve higher PRES and RECALL scores than the centralized approach. The optimal scenario confirms that the MMs random forest is the best model. Comparing SVR's original and optimal scores, the authors observe just a small increase in the optimal score. This information in conjunction with the low score of the GMs SVR suggests that this specific model is not suitable for results merging using the GMs method. This is also proved by the PRES and RECALL score which are lower than the random merging.

### *4.2 Uncooperative environment*

In the uncooperative environment, the results are just ranked lists of documents; therefore, the authors assign local scores with the two methods described in the previous section.

| | | MMs random forest | GMs random forest |
|---|---|---|---|
| MAP@100 | CORI | +19.5% | –28.4% |
| | SSL | +7.1% | –35.8% |
| | SAFE | +28.2% | –23.2% |
| PRES@100 | CORI | 59.7%‡ | +14.4* |
| | SSL | +36.2%‡ | –2.3% |
| | SAFE | +52.6%‡ | +9.3% |
| RECALL@100 | CORI | +60.4%‡ | +14.9%* |
| | SSL | +37.1%‡ | –1.7% |
| | SAFE | +53.3%‡ | +9.7% |

**Notes:** Asterisk (*), dagger (†) and double-dagger (‡) represent statistical significance at the 90%, 95% and 99% confidence intervals, respectively.

**Table III.** Proportional differences in the cooperative environment

| | | MAP@100 | | PRES@100 | | RECALL@100 | |
|---|---|---|---|---|---|---|---|
| | | Realistic scenario | Optimal scenario | Original scenario | Optimal scenario | Original scenario | Optimal scenario |
| MMs | Random forest | 0.0837 | 0.1061 | 0.2674 | 0.4376 | 0.2738 | 0.4525 |
| | SVR | 0.0709 | 0.0778 | 0.2348 | 0.3415 | 0.2413 | 0.3551 |
| | Decision tree | 0.0774 | 0.0927 | 0.2672 | 0.4389 | 0.2740 | 0.4542 |
| | Polynomial $x^2$ | 0.0437 | 0.0287 | 0.1182 | 0.1771 | 0.1200 | 0.1855 |
| | Polynomial $x^3$ | 0.0440 | 0.0325 | 0.1252 | 0.2231 | 0.1275 | 0.2336 |
| GMS | Random forest | 0.0460 | 0.0512 | 0.2434 | 0.2705 | 0.2513 | 0.2791 |
| | SVR | 0.0217 | 0.0292 | 0.0846 | 0.2026 | 0.0882 | 0.2121 |
| | Decision tree | 0.0420 | 0.0452 | 0.2318 | 0.2566 | 0.2394 | 0.2647 |
| | Linear regression | 0.0436 | 0.0590 | 0.2444 | 0.2653 | 0.2523 | 0.2725 |
| | DNN | 0.0412 | 0.0606 | 0.2434 | 0.2701 | 0.2510 | 0.2771 |
| | CORI | 0.0714 | 0.0729 | 0.1940 | 0.2832 | 0.1969 | 0.2912 |
| | SSL | 0.0623 | 0.1170 | 0.2168 | 0.4697 | 0.2219 | 0.4857 |
| | SAFE | 0.0606 | 0.0873 | 0.2200 | 0.2829 | 0.2262 | 0.2903 |
| | Centralized | 0.0793 | | 0.2592 | | 0.2660 | |
| | Random merging | 0.0141 | | 0.0979 | | 0.1833 | |

**Table IV.**
Results when artificial and source selection scores assigned to documents

*4.2.1 Artificial and source selection scores.* In this experimental setup, the authors assign artificial and source selection scores using formula (2). Table IV presents the results when artificial and source selection scores are used. The same models as before are used. For MMs, the authors implement random forest, SVR, decision tree and polynomial regression. The best performance algorithm is again the MMs random forest, followed by the MMs decision tree. These two models also achieve higher scores than CORI, SSL and SAFE in all three metrics with statistical significance for PRES and RECALL at different confidence levels. Table V presents the statistical differences between the best MMs and GMs models compared to CORI, SSL and SAFE. Furthermore, SSL's scores are higher than both polynomial regressions, so once more, the linear mapping is more appropriate than the polynomial mapping.

For the GMs, the authors used random forest, SVR, decision tree, linear regression and DNN with the same parameters as in the cooperative environment. The highest MAP score in the GMs is from the random forest, and the highest PRES and RECALL are from the linear regression. The baselines CORI, SSL and SAFE achieved better MAP than all GMs; however, the same does not apply for PRES and RECALL. Comparing CORI and SSL, the

| | | MMs random forest | GMs linear regression |
|---|---|---|---|
| MAP@100 | CORI | +17.2% | −38.9%‡ |
| | SSL | +34.3% | −30%† |
| | SAFE | +38.1% | −28%* |
| PRES@100 | CORI | +37.8%‡ | +25.9%‡ |
| | SSL | +23.3%† | +12.7% |
| | SAFE | +21.5%* | +11% |
| RECALL@100 | CORI | +39%‡ | +28.1%‡ |
| | SSL | +23.3%† | +13.7% |
| | SAFE | +21%* | +11.5% |

**Table V.**
Proportional differences when artificial and source selections scores are assigned to documents

**Notes:** Asterisk (*), dagger (†) and double-dagger (‡) represent statistical significance at the 90%, 95% and 99% confidence intervals, respectively.

authors observe that CORI outperformed SSL at MAP. This finding is important as it suggests that CORI can be more robust than SSL under specific conditions. In Section 4.1, using the cooperative environment settings, the SSL performs better than CORI, which is consistent with the literature (Si and Callan, 2003).

The optimal scenario shows that the best overall model is SSL. This suggests that in a perfect scenario where there is no source selection, SSL would be the best option for results merging. However, source selection is an integral part of FS, and this proves the robustness of the MMs random forest and its superiority over SSL and the other baselines in real-world settings for merging the results.

*4.2.2 Artificial scores.* In the final set of experiments, the authors assign artificial scores to the documents using the Equation (1). Table VI presents the results and Table VII the proportional differences between the best recall-performing models and the baseline algorithms. The findings are similar to the previous uncooperative environment. The scores of the MMs suggest that random forest is the most appropriate model which outperformed all other models in all metrics. The second best is the MMs decision tree and both of them performed better than the baselines

| | | MAP@100 | | PRES@100 | | RECALL@100 | |
|---|---|---|---|---|---|---|---|
| | | Realistic scenario | Optimal scenario | Original scenario | Optimal scenario | Original scenario | Optimal scenario |
| MMs | Random forest | 0.0812 | 0.1042 | 0.2541 | 0.4368 | 0.2606 | 0.4515 |
| | SVR | 0.0686 | 0.0779 | 0.2207 | 0.3423 | 0.2270 | 0.3559 |
| | Decision tree | 0.0751 | 0.0913 | 0.2519 | 0.4389 | 0.2586 | 0.4541 |
| | Polynomial $x^2$ | 0.0414 | 0.0287 | 0.1041 | 0.1779 | 0.1057 | 0.1864 |
| | Polynomial $x^3$ | 0.0417 | 0.0325 | 0.1111 | 0.2240 | 0.1132 | 0.2345 |
| GMS | Random forest | 0.0284 | 0.0297 | 0.2014 | 0.2133 | 0.2106 | 0.2235 |
| | SVR | 0.0215 | 0.0285 | 0.0811 | 0.2037 | 0.0845 | 0.2136 |
| | Decision tree | 0.0272 | 0.0291 | 0.1885 | 0.2204 | 0.1976 | 0.2311 |
| | Linear regression | 0.0308 | 0.0384 | 0.2269 | 0.2575 | 0.2367 | 0.2672 |
| | DNN | 0.0301 | 0.0408 | 0.2297 | 0.2571 | 0.2401 | 0.2668 |
| | CORI | 0.0681 | 0.0777 | 0.1699 | 0.2827 | 0.1726 | 0.2905 |
| | SSL | 0.0601 | 0.1177 | 0.2026 | 0.3287 | 0.2076 | 0.4901 |
| | SAFE | 0.0606 | 0.0873 | 0.2200 | 0.2829 | 0.2262 | 0.2903 |
| | Centralized | 0.0793 | | 0.2592 | | 0.2660 | |
| | Random merging | 0.0141 | | 0.0979 | | 0.1833 | |

**Table VI.** Results when artificial scores are assigned to documents

| | | MMs random forest | GMs DNN |
|---|---|---|---|
| MAP@100 | CORI | +19.2% | −55.8%‡ |
| | SSL | +35.1% | −49.9%‡ |
| | SAFE | +33.9% | −50.3%‡ |
| PRES@100 | CORI | +49.5%‡ | +35.1%‡ |
| | SSL | +25.4%† | +13.3% |
| | SAFE | +15.4% | +4.4% |
| RECALL@100 | CORI | +50.9%‡ | +39.1%‡ |
| | SSL | +25.5%† | +15.6%* |
| | SAFE | +15.2% | +6.14% |

**Notes:** Asterisk (*), dagger (†) and double-dagger (‡) represent statistical significance at the 90%, 95% and 99% confidence intervals, respectively.

**Table VII.** Proportional differences when artificial scores are assigned to documents

CORI, SSL and SAFE. The best results in the GMs are from the linear regression comparing the MAP and from the DNN comparing PRES and RECALL. The optimal scores confirm that linear regression would be the best option if there were no source selection phase.

The main difference with the previous uncooperative environment is the lower scores when assigning local scores using Equation (1) compared to the respective scores when using Equation (2). This means that the weighted local scores are more suitable for the results merging problem and adding the source selection relevancy affected the results positively. Thus, our assumption that adding the source selection score will reduce the bias of the less relevant collection described in Section 3.4 is true and Equation (2) is eventually a better option than assigning artificial scores using Equation (1).

## 5. Discussion

The baseline models CORI, SSL and SAFE in general go along with the same effectiveness as reported in their original papers. It is important to mention that in their original papers, non-patent datasets were used. Our experiments focus on the domain-specific dataset CLEF-IP thus, the authors examine the transferability of the baseline models in the patent domain. In the uncooperative environment, the real-settings environment, the best model from the existing algorithms is SAFE followed by SSL, and finally, the authors have CORI. In some cases, CORI outperforms both SSL and SAFE, such as in the MAP metric in the uncooperative environment. The superiority of CORI in terms of MAP perhaps has to do with the nature of the patent documents (i.e. long documents and many domain-specific words). Since CORI considers document frequencies in its source selection formula, it could get a boost from both long documents and domain-specific word characteristics compared to other methods.

Another result to discuss is that in the optimal scenario, the SSL model performed better in the real-settings uncooperative environment compared to the cooperative environment where the SSL model was initially developed. This means the authors convert the well-known problem of the lack of relevancy scores in nowadays search engines to a boost to the results or, in other words, to an advantage to the final ranking.

The critical finding of this research is that the authors prove that ML models can be used to replace standard methods and models that used for results merging for many years. Our methods outperformed state-of-the-art estimation methods for results merging, and they proved that they are more effective for federated patent search.

Finally, the finding that random forest is the best model in all three different environments also proves that the mapping between documents' scores from resources and documents' scores from the CSI is not linear as has been assumed by now and not polynomial either. The random forest can fit and predict the centralized global document scores better.

## 6. Conclusion

This article examines state-of-the-art results merging models in the patent domain and presents MLRM. The authors find another field where the interpolation of ML methods set new state-of-the-art models. Our method, especially MMs random forest, outperformed all the baselines in all different environments with significant improvements up to +60 per cent and become a state-of-the-art model for results merging. The finding of the decision tree as the second-best model is reasonable as a random forest is just an ensemble of decision trees. The authors use 100 trees for a forest in the experiments so this is a parameter they need to further investigate.

Comparing the methods that solve the lack of relevancy scores, it is clear that adding the source selection score increases the performance of the models. Thus, assigning artificial scores and multiplying them with source selection scores (formula (2)) is better than just assigning artificial scores in linear descending order (formula (1)). This is happening because multiplying with the source selection score will increase and decrease the final score of the relevant and non-relevant documents, respectively.

Our best model achieved higher scores than the centralized approach during the uncooperative environment at all metrics and achieved better PRES and RECALL scores in the cooperative environment. Thus, FS is not only a helpful tool to retrieve patents saved in the different resources but also, a better tool than the most widely used centralized approach.

As already mentioned, the dataset the authors used contains patent documents, so our results refer to the patent industry. Since there are special characteristics in the patent documents, such as the structure and length of the documents compared with general documents and the many domain-specific words (Verberne *et al.*, 2010), it would be interesting to test our methods in other datasets and extend their coverage. However, the focus of this article is the patent domain, so the authors left it for future work. Additionally, for future work, the authors plan to use reusable models so that their training will be saved, and it will be able to re-use the trained models. Finally, they plan to implement our results merging method to real professional search systems such as PerFedPat (Salampasis and Hanbury, 2014) and examine their usage with professional users.

**ORCID iDs**

Vasileios Stamatis http://orcid.org/0000-0001-5370-9695

Michail Salampasis http://orcid.org/0000-0003-4087-125X

## References

Avrahami, T.T., Yau, L., Si, L. and Callan, J. (2006), "The FedLemur project: federated search in the real world", *Journal of the American Society for Information Science and Technology*, Vol. 57 No. 3, pp. 347-358.

Callan, J. (2002), "Distributed information retrieval", in Croft, W.B. (Ed.), *Advances in Information Retrieval*, *The Information Retrieval Series*, Vol. 7, Springer, Boston, MA, pp. 127-150. doi: 10.1007/0-306-47019-5_5

Callan, J. and Connell, M. (2001), "Query-based sampling of text databases", *ACM Transactions on Information Systems*, Vol. 19 No. 2, pp. 97-130.

Callan, J.P., Lu, Z. and Croft, B. (1995), "Searching distributed collections with inference networks", *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '95)*, Association for Computing Machinery, NY, USA, pp. 21-28.

Clarke, N.S. (2018), "The basics of patent searching", *World Patent Information*, Vol. 54, pp. S4-S10.

Craswell, N., Hawking, D. and Thistlewaite, P. (1999), "Merging results from isolated search engines", *Proceedings of the 10th Australasian Database Conference (ADC'99)*, *Australian Computer Science Communications*, *18-21 January 1999*, Springer, Auckland, New Zealand, Vol. 21 No. 2, pp. 189-200.

Giachanou, A. and Salampasis, M. (2014), "IPC Selection using collection selection algorithms", in Lamas, D. and Buitelaar, P. (Eds), Multidisciplinary Information Retrieval, Springer International Publishing, Cham, Vol. 8849, IRFC, LNCS, pp. 41-52.

Giachanou, A., Salampasis, M. and Paltoglou, G. (2015), "Multilayer source selection as a tool for supporting patent search and classification", *Information Retrieval Journal*, Vol. 18, pp. 559-585.

He, C., Hong, D. and Si, L. (2011), "A weighted curve fitting method for result merging in federated search", *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '11)*, Association for Computing Machinery, NY, USA.

Hong, D. and Si, L. (2012), "Mixture model with multiple centralized retrieval algorithms for result merging in federated search", *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '12)*, Association for Computing Machinery, NY, USA.

Hung, V.T. (2019), "New re-ranking approach in merging search results", *Informatica*, Vol. 43, No 2, pp. 235-241.

Khode, A. and Jambhorkar, S. (2019), "Effect of technical domains and patent structure on patent information retrieval", *International Journal of Engineering and Advanced Technology*, Vol. 9 No. 1, pp. 6067-6074.

Lee, C.-J., Ai, Q., Croft, B.W. and Sheldon, D. (2015), "An optimization framework for merging multiple result lists", *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (CIKM '15)*, Association for Computing Machinery, NY, USA.

Lupu, M. and Hanbury, A. (2013), "Patent retrieval", *Foundations and Trends in Information Retrieval*, Vol. 7 No. 1, pp. 1-97.

Mahdabi, P., Gerani, S., Huang, J.X. and Crestani, F. (2013), "Leveraging conceptual Lexicon: query disambiguation using proximity information for patent retrieval", *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 13)*, Association for Computing Machinery, NY, USA.

Mao, J., Mukherjee, R., Raghavan, P. and Tsaparas, P. (2004), "Method and apparatus for merging result lists from multiple search engines", US patent No. US 6,728,704 B2.

Paltoglou, G., Salampasis, M. and Satratzemi, M. (2007), "Results merging algorithm using multiple regression models", *Proceedings of the 29th European Conference on IR Research (ECIR'07)*, Springer-Verlag, Berlin, Heidelberg, pp. 173-184.

Paltoglou, G., Salampasis, M. and Satratzemi, M. (2008), "A Results Merging algorithm for distributed information retrieval environments that combines regression methodologies with a selective download phase", *Information Processing and Management*, Vol. 44 No. 4, pp. 1580-1599.

Piroi, F., Lupu, M., Hanbury, A. and Veronika, Z. (2011), "CLEF-IP 2011: retrieval in the intellectual property domain", *CLEF 2011 Labs and Workshop, Notebook Papers*, Amsterdam.

Salampasis, M. (2017), "Federated patent search", in Lupu, M., Mayer, K., Kando, N. and Trippe, A.J. (Eds), *Current Challenges in Patent Information Retrieval*, pp. 213-240.

Salampasis, M. and Hanbury, A. (2014), "PerFedPat: an integrated federated system for patent search", *World Patent Information*, Vol. 38, pp. 4-11.

Salampasis, M., Paltoglou, G. and Giahanou, A. (2012), "Report on the CLEF-IP 2012 experiments: search of topically organized patents", in Forner, P., Karlgren, J. and Womser-Hacker, C. (Eds), *CLEF (Online Working Notes/Labs/Workshop)*, [Wil88], Peter Willett, CEUR Workshop Proceedings, Aachen, Germany.

Shalaby, W. and Zadrozny, W. (2019), "Patent retrieval: a literature review", *Knowledge and Information Systems*, Vol. 61, pp. 631-660.

Sheldon, D., Shokouhi, M., Szummer, M. and Craswell, N. (2011), "Lambdamerge: merging the results of query reformulations", *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (WSDM '11)*, Association for Computing Machinery, NY, USA.

Shokouhi, M. and Zobel, J. (2009), "Robust result merging using sample-based score estimates", *ACM Transactions on Information Systems*, Vol. 27 No. 3, pp. 1-29.

Si, L. and Callan, J. (2003), "A semisupervised learning method to merge search engine results", *ACM Transactions on Information Systems*, Vol. 21 No. 4, pp. 457-491.

Stamatis, V. and Salampasis, M. (2020), "Results merging in the patent domain", *Proceedings of the 24th Pan-Hellenic Conference on Informatics (PCI 2020)*, Association for Computing Machinery, NY, USA.

Taylor, M., Radlinski, F. and Shokouhi, M. (2016), "Merging search results", US patent No. US 9,495.460 B2.

Verberne, S., D'hondt, E., Oostdijk, N. and Koster, C.H. (2010), "Quantifying the challenges in parsing patent claims", *1st International Workshop on Advances in Patent Information Retrieval (AsPIRe'10)*, Association for Computing Machinery, NY, USA.

Vijaya, P., Raju, G. and Ray, S.K.R. (2016), "Artificial neural network-based merging score for meta search engine", *Journal of Central South University*, Vol. 23, pp. 2604-2615.

Voorhees, E.M., Gupta, N.K. and Johnson-laird, B. (1995), "The collection fusion problem", *Proceedings of the Third Text Retrieval Conference (TREC-3)*, National Institute of Standards and Technology, MD, USA, pp. 225-500.

Walid, M. and Gareth, J.J. (2010), "PRES: a score metric for evaluating recall-oriented information retrieval applications", *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '10)*, Geneva.

Yang, P., Fang, H. and Lin, J. (2018), "Anserini: reproducible ranking baselines using Lucene", *Journal of Data and Information Quality*, Vol. 10 No. 4, pp. 1-20.

## Further Reading

MAREC Data Set [Online] (2009), available at: https://researchdata.tuwien.ac.at/records/2zx6e-5pr64 (accessed 15 April 2020).

**Corresponding author**
Vasileios Stamatis can be contacted at: vstamatis@it.teithe.gr