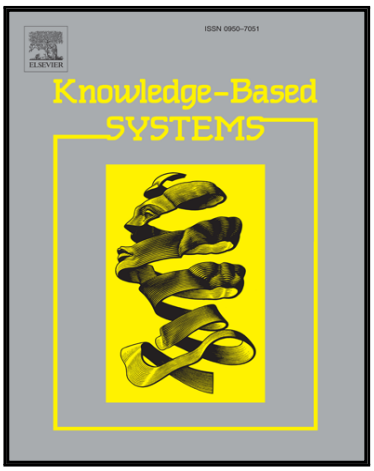


A recommender system of reviewers and experts in reviewing problems

Jarosław Protasiewicz, Witold Pedrycz, Marek Kozłowski, Sławomir Dadas, Tomasz Stanisławek, Agata Kopacz, Małgorzata Gałeżewska

PII: S0950-7051(16)30138-1
DOI: 10.1016/j.knosys.2016.05.041
Reference: KNOSYS 3538

To appear in: *Knowledge-Based Systems*

Received date: 9 February 2016
Revised date: 18 May 2016
Accepted date: 21 May 2016

Please cite this article as: Jarosław Protasiewicz, Witold Pedrycz, Marek Kozłowski, Sławomir Dadas, Tomasz Stanisławek, Agata Kopacz, Małgorzata Gałeżewska, A recommender system of reviewers and experts in reviewing problems, *Knowledge-Based Systems* (2016), doi: 10.1016/j.knosys.2016.05.041

A recommender system of reviewers and experts in reviewing problems☆

Jarosław Protasiewicz[a,*], Witold Pedrycz[b], Marek Kozłowski[a], Sławomir Dadas[a], Tomasz Stanisławek[a], Agata Kopacz[a], Małgorzata Gałężewska[a]

[a] *National Information Processing Institute, Warsaw, Poland*
[b] *Department of Electrical & Computer Engineering, University of Alberta, Edmonton, Canada*

Abstract

In this study, we propose the architecture of a content-based recommender system aimed at the selection of reviewers (experts) to evaluate research proposals or articles. We introduce a comprehensive algorithmic framework supported by various techniques of information retrieval. We propose a well-rounded methodology that explores concepts of data, information, knowledge, and relations between them to support a formation of a suitable recommendation. In particular, the developed system helps collecting data characterizing potential reviewers, retrieving information from relational and unstructured data, and formulating a set of recommendations. The designed system architecture is modular from the functional perspective and hierarchical from the technical point of view. Each essential part of the system is treated as a separate module, whereas each layer supports a certain functionality of the system. The modularity of the architecture facilitates its maintainability. The process of information retrieval includes classification of publications, author disambiguation, keywords extraction, and full-text indexing, whereas recommendations are based on the combination of a cosine similarity between keywords and a full-text index. The proposed system has been verified through a case study run at the National Center for Research and Development, Warsaw, Poland.

Keywords: reviewer assignment problem, recommender system, data acquisition, information retrieval, content-based filtering.

1. Introduction

When people think about a reviewing process they usually focus on articles because they are looking for suitable journals as a way to disseminate high quality and timely knowledge. Not only publications are important in the science

*Corresponding author:
Email address: jaroslaw.protasiewicz@opi.org.pl (Jarosław Protasiewicz)

world. Research and development projects are crucial to the improvement of security and prosperity of countries, organizations and companies. As innovative activities are highly risky, therefore, these projects are predominantly financed by public funds, either sponsors' or philanthropists' donations, or private companies searching for new technologies. Forasmuch as granting bodies have a finite amount of money to redistribute they need to prioritize applications in accordance with social needs, economic reasons, scientific goals, and quality of proposals. Usually, evaluation of project proposals and distribution of available funding are based on reviews prepared by academic reviewers and professional experts. On the other hand, already completed projects need an assessment to check whether their objectives and targets were achieved. Moreover, companies look for professionals to assess investment plans, realize complicated projects, and alike. Finally, these demands imply a need for coping with the recommendation issue of reviewers, experts, and professionals suitable for a specified problem, which could be a project proposal, an article, a completed project, or just a demand for professionals.

It should be emphasized that a lack of conflict of interest, independence, and competencies of people producing reviews, recommendations, and opinions are crucial to the quality of evaluations. However, we should also realize, that those people have limited knowledge, experience, and perspective of looking at the works of other people. These limitations can cause a misinterpretation of the author's viewpoint and can lead to the rejection of an excellent scientific work, or a potentially successful project proposal. In order to understand the nature of the issues mentioned above, we should consider the psychological and social aspects of the review process. More specifically, there are three heuristics of cognitive distortions: availability, anchoring, and representativeness (Tversky and Kahneman, 1974). What is meant by this is that some biases may occur while choosing reviewers, experts, and professionals manually. Thus, this process should be supported by disinterested and automatic methods. Moreover, granting bodies expect that selection of reviewers will be unbiased and the time of processing applications should be as short as possible. To fulfill the requirements, we realized that we have to provide a possible large dataset of scientists as well as efficient methods for selecting them. These are the reasons why we decided to work on a recommender system of reviewers and experts.

We may view the assignment of people to problems as an extended version of the problem of generalized assignment. There are a number of sophisticated solutions to this problem, which are well documented in the literature (for details, see Section 2). However, some approaches contain only theoretical propositions, or the existing experimental evidence may be insufficient to be considered reliable, e.g. they are tested on data originating from a particular conference. Despite these concerns, some developments of practical relevance are worth noting. For instance, Chien and Chen (2008) presented an empirical study in the semiconductor industry, Rodriguez and Bollen (2008) proposed a fully automatic solution tested on data coming from a selected conference, whereas Wang et al. (2013) used real data but only for algorithms checking. Decision support systems proposed by Tian et al. 2002, Fan et al. 2009, Sun et al. 2008, Xu et al.

2010 are the most interesting solutions as they have been reported as practically used indicating their level of maturity and usefulness. However, these tools require human assistance, for instance, they need a manual commitment in classifying reviewers and proposals, and eventually preparing knowledge rules. It should be noted that Fan et al. (2009) introduced automatic proposals grouping process, which brought some improvement to this area. Another problem is that algorithms take advantage of human experience expressed in a structured way, however, in reality, such data may be nonstructural. A good example here are keywords describing people's experience and objects like a manuscript, a project, etc. It could be difficult to match efficiently people and objects using, for example, cosine measure between keywords because different terms can have similar meaning, so a strict fit is impossible. Tayal et al. (2013) proposed fuzzy logic to circumvent this problem. A more promising way to follow would be the use of unstructured data or semantic relations between terms coming from the Computer Science Bibliography (DBLP) (Mishra and Singh, 2011), manuscript references (Rodriguez and Bollen, 2008), and home pages (Basu et al., 2001) or the application of context-aware systems (Li et al., 2015).

The observations made above underline a genuine need for further research and development of new approaches to the selection of reviewers and experts. In this study, we put forward a proposal along with associated experimental studies aimed at capturing experience, intuition, and informal observations. Firstly, expertise of people may be described not only in terms structured data, e.g. keywords, but also unstructured data could be useful with this regard. There is some rationale assumption in the present literature, and this is coincident with our observations. Secondly, we believe that areas of expertise declared by experts might be inconsistent with those inferred from their professional track record. Thirdly, the previous research has proposed algorithms that indeed need some human assistance. However, we are convinced that it is possible to choose automatically reviewers or experts using a recommender system that works autonomously without any manual adjustment.

In this study, having these assumptions in mind, we propose a methodology that explores concepts of data, information, knowledge, and relations between them. The methodology supports a formation of a recommender system, which collects data concerning researchers coming from various sources including public databases and the Internet. Next, information is retrieved from relational and unstructured data to build expertise profiles of researchers and professionals. Finally, it recommends reviewers and experts for a specified problem on a basis of knowledge concerning potential candidates. The similarity between a problem under discussion and peoples' expertise is quantified through the combination of cosine measure and a full-text index. It should be noted that Basu et al. (2001), Flach et al. (2010), Ryabokon et al. (2012) used a cosine measure for matching between expertise of people and a problem. However, our approach augments this as the similarity not only involves but combines it cosine similarity between keywords with a full-text index, therefore incorporating unstructured data into the final recommendation.

Recommender systems are mainly used in e-commerce (Bobadilla et al.,

2013b). We propose a new area for application such systems, which is the recommendation of people (experts and reviewers) who may be able to assess an article or a project. Among known types of filtering used in these systems, i.e. collaborative, demographic, content-based, and hybrid (for more, see Section 2), our approach utilizes content-based filtering. The collaborative recommendation systems make suggestions based on the past behavior of other users and the similarity between users and items. This kind of procedure demands historical data, i.e. users' ratings. In our case, there is the lack of historical data describing reviewer's assignments. Therefore, we decided to build the content-based recommendation engine, where we can avoid a cold-start problem. The key of the systems based on the content is that information coming from recommended objects is similar to the user's profile data. Such tools are mainly used to recommend documents, Web pages, publications, jokes, or news. Some examples are SYSKILL & WEBERT, which recommend Web pages or PTV, which recommends TV programs to the user (Martínez et al., 2009). We have not found any other content-based system used to recommend reviewers or experts for project proposals or articles. Our work seems to be the pioneer according to those two conditions, i.e. the reviewers' selection domain and the content-based type.

The main objective of this study is to design an architecture of a content-based recommender system along with a comprehensive algorithmic framework that supports a thorough information retrieval and offers a sound framework of ranking potential reviewers. The system should work autonomously without any or with a limited human input. One should stress, though, that the proposed recommender system is meant to support human and offering a decision-support environment.

The study elaborates on the overall architecture of the system and functionalities of each module. The main outcome of this study is the recommender system of reviewers and experts, which is the improved version of the decision support system for selection of reviewers (Protasiewicz, 2014). Moreover, the recommender system was deployed in the National Centre for Research and Development of Poland (NCBR).

The paper is structured as follows. Section 3 presents an overview of the architecture of the proposed system, whereas Section 4 elaborates on the pertinent details showing the underlying algorithmic aspects of the system. Section 5 covers specific technical details as well as some experimental results. Finally, conclusions are presented.

2. Related studies

A reviewing of scientific works has been present since the Middle Ages. Its first mention appeared in IX century when Ishaq bin Ali al-Rahwi in his book *Ethics of the Physician* suggested that physician should rate methods used in treatment to improve their standards and quality (Spier, 2002). In modern times (XVIII century), Henry Oldenburg established a review process of *Philosophical Transactions* of the *Royal Society* magazine, where a group of experts in a

given field evaluated manuscripts to take a publishing decision. It could be said that this review process is the ascendant of what nowadays we call peer review. Since that time, three kinds of reviewing have been developed: open peer review, single-blind review, and double-blind review. A review has to be effective, productive, honest, and socially responsible irrespective of a chosen reviewing method (August and Muraskin, 1999).

It is believed that human decisions may be influenced by cognitive distortions like availability, anchoring, and representativeness (Tversky and Kahneman, 1974). Since reviewing is also human activity, these distortions may cause some biases in an assessment of other people works. Table 1 contains a short summary of potential distortions that may occur during a reviewers selection, reviewing process, and final decision regarding a project or an article. The summary is based on the existing literature (Bornmann and Daniel, 2009, Eisenhart, 2002, Hemlin, 2009, Hojat and Rosenzweig, 2004, Jacoby et al., 1989, Langfeldt, 2004, Marsh et al., 2008, Rivara et al., 2007).

Distortion type	**A**	**B**	**C**
interest conflict	x	x	x
tendency to crude assessment		x	
discrimination of novel and controversial ideas		x	
thematic and ideological biases		x	x
focusing on frequently or recently appearing names		x	
focusing only on particular details		x	
tendency to confirmation of clarified opinions		x	
influence of an author's affiliation or position	x	x	x
gender effect		x	
reviewers suggested by authors		x	
syndrome of group thinking and a congruence effect		x	x
focusing on the first information and a contrast effect		x	x
tendency to support positively verified hypotheses		x	

Table 1: The distortions that might occur during a reviewers selection (A), a reviewing process (B), and a final decision regarding a project or an article (C).

Numerous attempts to solve the problem of finding a person that is fully qualified to fulfill a specific task have been reported in the literature. Among them, some works are especially worth noting due to their relevance to the study

presented here, i.e. optimization algorithms, heuristics, artificial intelligence, machine learning, semantic data models, and decision support systems.

An optimization approach is suitable to well-defined problems, where finding a right person is equivalent to the minimization of a certain objective function. The most difficult issues to solve are multi-objective optimization problems, which occur when many matching measures are simultaneously used. Luckily, these tasks could be simplified, for instance, Wang et al. (2013) converted a multi-objective optimization problem to a mixed integer programming model and solved it by a two-phase stochastic-based greedy algorithm. Not only pure linear programming is suitable for proposing a ranking of reviewers (Hartvigsen et al., 1999) but also some its combinations, e.g., Cook et al. (2005) combined simplex and heuristic algorithms, whereas Goldsmith and Sloan (2007) applied a polynomial-time algorithm to find the maximum flow through a network because an assignment problem was reduced to the instance of a network flow problem.

On the other hand, weakly defined problems, where not every variable or constraint is known, may require some less strict algorithms. The literature shows that heuristic algorithms and method of artificial intelligence are suitable to search optimal combinations of people and tasks effectively, for example, tabu-search (Kolasa and Krol, 2011), genetic algorithms (Harper et al., 2005, Kolasa and Krol, 2011), a greedy randomized adaptive search procedure combined with a genetic algorithm (Sun et al., 2008), an ant colony algorithm (Kolasa and Krol, 2011), a greedy and evolutionary algorithm (Merelo-Guervos and Castillo-Valdivieso, 2004), heuristic knowledge rules and a genetic algorithm (Fan et al., 2009), and fuzzy sets (Tayal et al., 2013).

The development of electronic media implies that more and more data describing people is available on-line. It is known that much useful data concerning people becomes available in various sources, but it is impossible to analyze it by humans due to its large size. Fortunately, machine learning methods efficiently deal with large datasets and can recommend the most likely answers, e.g., Chien and Chen (2008) applied decision trees and association rules to data mining to extract people profiles and rules, which next were used for personnel selection. Azar et al. (2013) tested the application of decision trees to staff choise.

Additionally, semantic data models are more profitable than raw data to gain knowledge about people and relations between them. There is a substantial amount of sources like co-author networks, linked data, and the semantic web that provide us with useful information about scientists, experts, and professionals. Co-author networks and linked data are applied to assignment problems in the works as follows. Mishra and Singh (2011) proposed an expert semantic finder in order to look for experts and also to discover an experts collaboration network using their publications from the DBLP. Aleman-Meza et al. (2007b) showed how to build a collaboration network based on co-authorship and a computer science taxonomy, whereas Rodriguez and Bollen (2008) retrieved potential reviewers by building a co-author network from references included in publications. Furthermore, semantic and web data have substantially enriched assignment methods, e.g., Basu et al. (2001) looked at interests of reviewers using their homepages, whereas Ryabokon et al. (2012) retrieved information

about connections between authors. Liu and Dew (2004) utilized information coming from heterogeneous sources and mixed a keyword search with a concept search. Song et al. (2005) proposed a relational and evolutionary graph model that uses relational and textual data for describing and finding experts. Aleman-Meza et al. (2007a) discussed the ExpertFinder framework and enriched it by vocabularies coming from the semantic web.

It should be emphasized that reviewer or expert selection tools can be considered as a decision support mechanism. In particular, this approach is represented by (Fan et al., 2009, Tian et al., 2002, Sun et al., 2008, Xu et al., 2010), while Tian et al. (2005) also included organizational issues in their system. Moreover, Green and Callaham (2011) proposed a three-tier reviewer assessment system based on reviewers' quality and reliability, which helps manage a pool of reviewers co-working with a particular journal. Papagelis et al. (2005) proposed a management system that helps match papers and reviewers, and explore reviewers' interests in papers, analyze conflicts of interests and balance reviewers workload in a conference. Sun et al. (2008) have presented a complete decision support system to assign the most appropriate experts to the relevant R&D project. They assumed that reviewers should be assigned independently for each proposal. Xu et al. (2010) improved this approach by adding the procedure of grouping proposals and then assigning reviewers to such selected groups.

Regardless of which algorithm has been used and whatever type of data is being processed, we have found that the most mature and applicable solutions are those which are based on the idea of a decision support system. Tian et al. (2005) and later Sun et al. (2008), then Fan et al. (2009) and Xu et al. (2010) have proposed a complete methodology and systems based on that conception. The systems are designed to support granting processes used by the Natural Science Foundation of China. The authors noted that applied methodology, which have been tested in real cases, can reduce potential political inference and eliminate subjective mistakes, as well as streamlines and standardize the process of a reviewers assignment. On the other hand, there are open systems, for instance, Rodriguez et al. (2006) developed a Repository-Centric Peer-review Model, which is composed of a central repository for storing publications using the Open Archiver Initiative protocol and methods for selecting reviewers, whereas Flach et al. (2010) proposed and created an open system for the purpose of International Knowledge Discovery and Data Mining conference.

A Recommendation System (RS) is an application capable of presenting a user a suggestion of objects, obtained on the basis of his/her profile, previous preferences, and the tastes of a community which has likings and opinions similar to the given user. Recommendation engines offer new items (products or content) to users based on various data (Crespo et al., 2011). Bobadilla et al. (2013b) proposed a taxonomy of recommendation systems in terms of information sources, methods of data processing, filtering algorithms, and others. Information sources may be explicit like users' ratings or implicit, e.g. users' profiles or any historical data reflecting their behaviour. These data are utilized by model-based methods involving the use of previously constructed models or memory-based methods, which mainly use some metrics measuring distance be-

tween items. The most crucial part of any recommendation system is a filtering algorithm, which may be one of the following types (Bobadilla et al., 2013b, Crespo et al., 2011):

1. Collaborative filtering (CF): in this approach users are represented by an N-dimensional vector of items, and the recommender looks for users who have similar rating patterns as the target user, then, it uses the ratings from those like-minded users to make a recommendation for the target user.
2. Demographic filtering: it is based on the idea that people having similar biological or cultural features (age, sex, country, etc.) may have something in common.
3. Content-based filtering: in this approach recommendations are based on data provided by users in the past. More specifically, content coming from objects somehow related to the users is analyzed in terms of a chosen similarity measure. For instance, the system recommends similar objects to the user profile, the algorithm constructs a search query to find favorite items (as experts) by the same categories, or/ and with similar keywords.
4. Hybrid filtering: it combines other methods, mainly CF with demographic filtering or CF with content-based filtering.

The most popular algorithms used in recommendations are based on collaborative filtering (Herlocker et al., 1999). This technique makes suggestions based on the historical data of all the users and the estimated resemblance between them. A collaborative recommendation concerns similarity degree between users. Typical metrics used for the computation of customers' metrics include Pearson correlation coefficient, adjusted cosine similarity, Spearman's rank correlation coefficient, and mean squared difference. Due to wide popularity of CF, there are plenty of its improvements with the main focus on similarity measures like the application of genetic algorithms to achieve an optimal and fast similarity function (Bobadilla et al., 2011), or the development of a low-cost similarity metric in terms of hardware needed to computations (Bobadilla et al., 2013a), or combining numerical votes with additional information related to common and uncommon votes (Bobadilla et al., 2010). A cold-start problem is well known in CF. It is when a new user or a new item without ratings appears and users may stop using RS due to a low quality of recommendations. The issue may be circumvented by proposing weights combinations of known measures optimized by a neural network (Bobadilla et al., 2012). We have to mention that Amazon uses its own item-to-item collaborative recommendation. The algorithm is focused on finding similar items, not similar customers, and hence, it scales independently of the number of customers.

The collaborative filtering can be used in a teaching environment, as the supportive tool for subject's student browsing interfaces (Martínez et al., 2009). In parallel with the advances in collaborative approaches, there is also a place for the application of graph theory and network analysis (Cordobés de la Calle et al., 2015). Regardless of a filtering method utilized by an RS, there is a wide range of such system's applications to the recommendation of various items in

e-commerce like music, books, documents, television programs (Bobadilla et al., 2013b), or even e-learning courses (Bobadilla et al., 2009).

3. The functional architecture of the proposed system

In this section, we present the functional and overall architecture of the proposed system and describe its functioning. Our objective is to show and discuss why a modular and hierarchical architecture has been chosen instead of a flat architecture containing a set of cooperating modules (Mishra and Singh, 2011) or a layered information system (Liu and Dew, 2004, Papagelis et al., 2005) including a decision support system (Tian et al., 2002, 2005, Xu et al., 2010). We propose the system architecture that is modular from the functional perspective and hierarchical from the technical point of view. Each essential part of the system is regarded as a separate module, whereas each layer realizes the technical function of the system (these aspects to be discussed in Section 5).

3.1. Architecture and processes

We propose a recommender system of reviewers and experts, which substantially enhances a decision support system for selection of reviewers (Protasiewicz, 2014). Like its previous version, the system is composed of three logical modules that focus on the concept of data, information, and knowledge. Firstly, a data acquisition module gathers data concerning both researchers and experts from public databases and the web. Next, an information retrieval module retrieves relevant information from relational and unstructured data to build an expertise profile of each individual. Finally, a recommendation module proposes either reviewers or experts that could serve the best for a proposal or an article evaluation (Fig. 1). Information and data are stored in a local database, and they are available via the user interface.

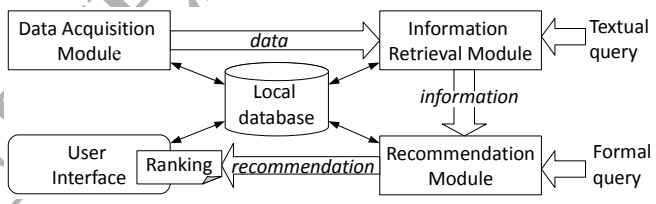


Figure 1: The functional architecture of the system (Protasiewicz, 2014).

The proposed model of data-information-knowledge is behind the autonomous recommender system containing the adaptive and self-growing knowledge database. In essence, the functionality of the system can be captured as follows. The system contains a single main background process and on-demand processes responsible for responding online to inquiries regarding reviewers or experts. The background process gathers data coming from various sources and then transforms these data to information corresponding to people's profiles (Fig. 2).

Practically, the system should respond to the following query: "Identify reviewers or experts to evaluate a project or to assess a proposal." The appropriate

$$\text{sources} \xrightarrow[data]{acquire} \text{data} \xrightarrow[information]{retrieve} \text{information (profiles)}$$

Figure 2: The main background process working throughout modules of the system.

on-demand process handles such query. The system can receive a project description (a textual query) that has to be assessed or keywords (a formal query) specifying that project (Fig. 1).

3.2. The data acquisition module

The main objective of the data acquisition module is to deal with various data sources to create the database of researchers' scientific achievements, which is as complete as possible. To accomplish this task, the module incorporates into the system the set of automatic or manual processes, which collect and import data concerning researchers and experts (Fig. 3). There are three possible types of data or information, i.e. (i) structured and (ii) unstructured data, (iii) information provided by users.

Structured datasets such as web services, data dumps, etc. are the most abundant data supply. Since they are well defined and regular, they can be acquired and parsed by several processes adapted to specific interfaces and data structures. They are referred to extractors and importers. The data of the second type are diverse collections of unstructured data such as web pages. They are handled by a focused crawler. In the sequel, information retrieval processes are needed to extract relevant information to researchers' expertise. The last source is information modified or added by the system users through a web interface. For instance, the users can edit their profiles, publications, and keywords. This source is more reliable than the others. Thus, provided information does not require any further transformations.

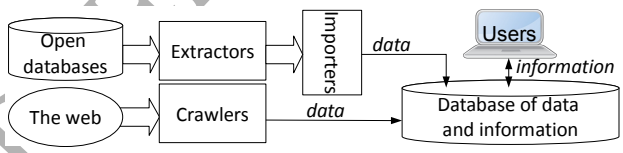


Figure 3: The functional architecture of the data acquisition module.

3.3. The information retrieval module

The information retrieval module is responsible for data transformation into relevant pieces of information about reviewers and experts. The transformation is handled by four main processes, i.e. classification, disambiguation, keyword extraction, and full-text indexing. An additional technical task of preprocessing is also carried out (Fig. 4). There are two separate ways of data transformation. The first one deals with relational data, whereas the second one copes with unstructured data. It should be underlined that, in fact, relational data consists of only publications. On the other hand, unstructured data can concern any document that describes a person.

Generally, the module works as follows. Firstly, publications are preprocessed. This involves stopwords removal, lemmatization, and TFxIDF transformation. Next, the keyword extraction process retrieves the most important words from titles and abstracts of publications in order to build a profile for each considered person. Subsequently, the publications are classified into scientific disciplines, which should improve disambiguation of authors. In parallel to these processes, any unstructured data is indexed, and persons' profiles are enriched by additional information about their achievements.

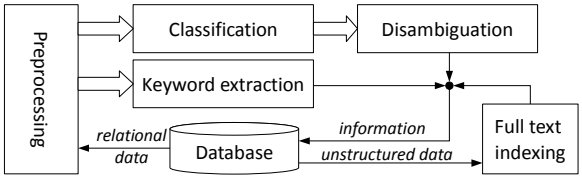


Figure 4: The functional architecture of the information retrieval module.

3.4. The recommendation module

The recommendation module is the most important part of the system. It combines a request for the recommendation with the overall information related to potential reviewers and experts in order to create a kind of knowledge that suggests which researchers are the most suitable for the specified problem (Fig. 5).

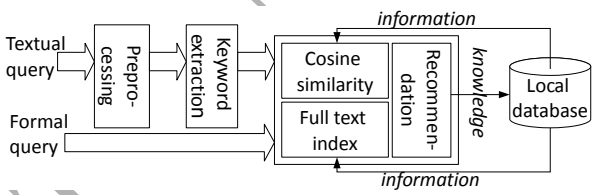


Figure 5: The functional architecture of the recommendation module.

The request for the recommendation can be sent to the system either as a formal query or a textual query. The first one occurs when the system receives a text, which is supposed to be reviewed (Fig. 6). The system has to process it by invoking the information retrieval module that uses of preprocessing procedures and keyword extraction algorithms to find the most relevant words in the text. Such kind of text summarization is information that only now can be sent directly to a recommendation algorithm. The formal query occurs when specific keywords and scientific fields constitute the query. They are conveyed directly to the recommendation algorithm (Fig. 7). In both cases, the system constructs a knowledge base of information (people's profiles) with keywords (a query), and returns a ranking list.

The underlying fundamentals of the recommendation algorithm and other algorithms are covered in Section 4.

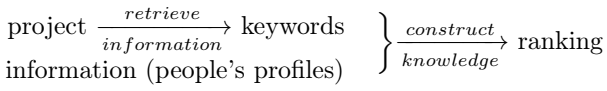


Figure 6: A textual query to the system.

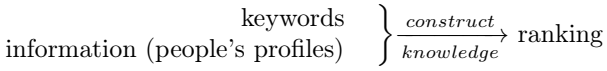


Figure 7: A formal query to the system.

4. Algorithmic developments

Here we cover the underlying fundamentals of the recommendation algorithm and some other closely related algorithms. Since the issues of publications classification (Protasiewicz et al., 2015) and keyword extraction (Kozlowski and Protasiewicz, 2014) have been previously presented, we just outline the main ideas of these methods. A recommendation algorithm has also been discussed in the literature (Protasiewicz, 2014), however, we provide its detailed description as being the most important part of the system.

4.1. Data extraction and crawling

The data acquisition module collects raw data coming from various sources by using the data extraction and crawling algorithms. Similarly to the overall approach to the system, we have proposed the modular architecture of the unit. It ensures separation of processes, i.e. extraction from crawling as well as it provides the partition of data sources. As a result, both algorithms return records of data, which can be stored in a database (Fig. 8, 9).

A data extraction algorithm. The data extractors implement algorithms of data extracting and importing that collect data from external databases of publications available in open access. Each of them is essentially an individual process, which parses input data from a source, extracts information about scientific publications, and transforms it into intermediate representation, which then is serialized as an XML file. Finally, the importers transform these files into records and save them in a local database (Fig. 8). In this way, the extractors can operate independently of each other, which means both downloading and parsing data in the context of a specific data source.

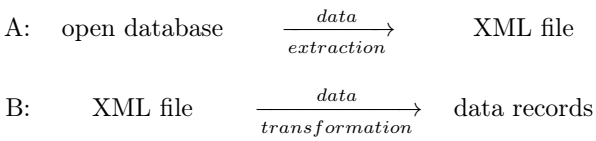


Figure 8: Data extraction and importing processes of the data acquisition module; A - a data extractor, B - an importer.

We have to mention that data categories for extractors are limited only to publications. While the data acquisition is primarily a technical process, there

are some problems that need to be solved by using heuristics. For example, the importer is able to discover that an imported publication is the duplicate of the already existing record, even if there are small differences in the titles of publications. In this case, the importer tries to merge the publication into the existing one, and at the same time, tries to fill missing attributes from the original record, such as keywords, abstracts, or other important fields.

A crawling algorithm. A web crawler that we have proposed is the type of a focused crawler. It collects additional data about reviewers' publications from their private web pages, however, data categories are limited only to publication. In contrast to extractors, the crawler works on unstructured documents and does not take any assumptions about the structure of downloaded documents. Therefore, it has to use heuristics and data mining techniques to recognize relevant entities coming from unstructured data. This is a highly demanding task considering the diversity and variety of web documents. Thus, the process has been split into the following two phases (Fig. 9).

The first phase involves crawling of pages and sub-pages of every individual scientist to find candidate documents that could potentially contain information about publications. The candidates are identified by a Uniform Resource Locator (URL) of a scientist' page, and the description of this URL (the text inside of an HTML link element), as well as the full text of this web page. Only pages containing words that are somewhat related to research terms are processed at the next step. Next, the HTML documents are transformed into their text representation in such a way that they still preserve some information about the structure of original documents. For example, paragraphs, block elements, rows in tables, list elements are written as separate lines of text.

At the second phase, a Conditional Random Fields (CRF) model scans each line in each document and decides whether the processed document contains information about researchers publication or not. As the CRF is a discriminative and general graph-based model (Sutton and McCallum, 2011), we assume that it is more suitable to tackle this task in comparison to a generative approach as such as, for instance, Hidden Markov models. The CRF model has been developed by using the manually labeled set of transformed pages. Besides using selected words coming from the text as features, we have extended the model with handcrafted features that were based on the line length, gazetteers and popular surnames in different countries, regular expressions for finding structures, acronyms and words that are distinctive for article citations. This approach allowed us to find lines containing information related to publications in HTML documents.

Since web crawlers and extractors are well documented in the literature, we focus here on the most interesting features that have been added when developing the system. The extractors are modular, they form the third layer of modularity situated below the second layer realizing communication among algorithms in modules, and the first layer of the system realizing cooperation between its parts. The core of extractors is fixed, however, each extractor has a parser layer that could be changed with regard to the demands of a data source.

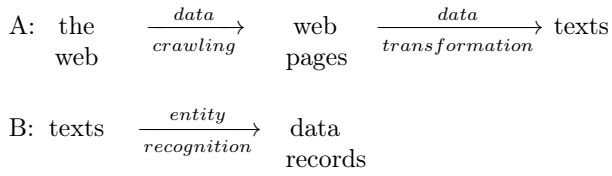


Figure 9: A crawling processes of the data acquisition module; A - a crawler, B - a CRF model.

Additionally, we have improved a typical crawler by Conditional Random Field algorithm, so we have realized the focused crawler that is capable of finding, locating, and acquiring publications from personal web pages of scientists.

4.2. Classification of publications

The goal of classification is to organize publications according to hierarchical science categories. The results of classification should improve a disambiguation process of authors completed by the information retrieval module. Moreover, they may help manage and visualize groups of publications, people, and keywords in the system.

More precisely, we want to assign the publications (textual documents) $d_i \in \mathbf{D}$, $i = 1, ..., I$, into relevant scientific areas (classes) $c^l_{j_l} \in \mathbf{C}$ that are organized in a hierarchical tree $\mathbf{C}$ composed of $l = 0, 1, ..., L$ levels and $j = 1, 2, ...J$ classes located at each level. Formally, we assign a relevant class $c^l_j \in \mathbf{C}$ to each document. A document d_i may be composed of several parts of a text, such as a title, an abstract or an introduction, sections, references, ect. These elements may occur many times in various languages in a single document. We call them features $f^t_i(lang)$, $t = 1, 2, ..., T$, where t is the feature type, *lang* is its language, and i is the document index. Finally, the task is to transform documents into their features and produce a list of pairs in the form:

$$d_i \in \mathbf{D} \Rightarrow f^t_i(lang) \Rightarrow \{d_i, c^l_j\}. \quad (1)$$

Classification approaches. There is a large number of efficient classification algorithms, as discussed in the previous studies (Protasiewicz et al., 2015). On this basis, we decided to construct classification models by using a Multinomial Naïve Bayes algorithm, which, in spite of eventually questionable assumption of attribute independence, works surprisingly well in practice. Moreover, we apply a so-called hierarchical approach to classification. It assumes the use of different classifiers in the succeeding hierarchy levels, so a document d_i can be classified more precisely at each classification level. Input data to be categorized are vectors of TFxIDF values formed from publication abstracts, titles, and keywords provided by the authors. We have to note that publications might be monolingual or multilingual, which means that they may concurrently contain parts (features) formed in various languages. Being aware of this fact we apply three strategies of classification, namely monolingual, maximum probability, and multilingual.

Monolingual classification. We assume that all document's features f_i^t are given in the same language. The documents that contain features in different languages at the same time are discarded. In this case, we just need to choose which classifier should be used for a particular document.

Maximum probability approach. It may happen that a document contains features that are expressed in many languages at the same time. A maximum probability approach assumes that there is a set of monolingual classifiers $m = 1, ..., M$, and the classifier is chosen for which the output produces the highest probability among all classifiers:

$$d_i \in \mathbf{D} \Rightarrow max\{p_j^l(m)\} \Rightarrow \{d_i, c_j^l\}. \tag{2}$$

Multilingual classification. A more advanced approach to the classification problem of multilingual documents is realized in a system composed of three layers: a data preprocessing module, a set of monolingual classifiers, and a multilingual decision module (Fig. 10). The data preprocessing unit receives documents d_i and transforms them into documents' features f_i^t. The data are analyzed by monolingual classifiers $m = 1, ..., M$. As a result, each monolingual classifier m generates the probabilities $p_{i,j}^l$ of belonging of each document d_i to each possible category c_j^l. More precisely, the probabilities are calculated separately for all documents d_i, $i = 1, ..., I$, models $m = 1, ..., M$, classification levels $l = 1, ..., L$, and classes c_j^l, $j = 1, ..., J$. Finally, the multilingual decision module combines the outputs of monolingual classifiers.

The multilingual decision module is based on an ensemble of logistic regression classifiers. Since they use the *one vs. others* classification rule, there are $J - 1$ classifiers trained for each classification level l. Each classifier generates the following decision:

$$\begin{array}{ll} d_i \in c_j^l & if\ h(\varphi_i^l) > h_{th} \\ d_i \notin c_j^l & if\ h(\varphi_i^l) \leq h_{th} \end{array}, \tag{3}$$

where h_{th} is a decision threshold, and $h(\varphi_i^l)$ is a logistic regression function calculated as follows

$$h(\varphi_i^l) = \frac{1}{1 + e^{-\varphi_i^l}}. \tag{4}$$

Its argument φ_i^l is a weighted sum of probabilities that are generated by the monolingual classifiers

$$\varphi_i^l = \sum_{j=1}^{J} \sum_{m=1}^{M} \left(w_{j,m}^l \cdot p_{i,j,m}^l \right), \tag{5}$$

where $w_{j,m}^l$ are the weights of a regression model produced by using quadratic programming.

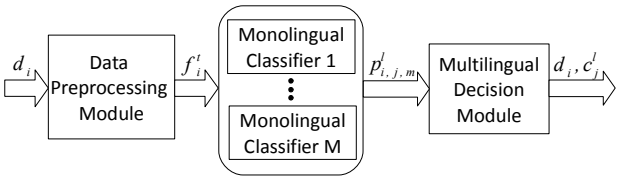


Figure 10: The multilingual classification system of the information retrieval module.

4.3. Authors disambiguation

The authors disambiguation is the important phase of creating complete profiles of researchers. Its main purpose is to identify publications of the authors by being provided only with their surnames and names or its initial letters in these existing publications. The identification relies on matching a person from a reference database of researchers to metadata coming from a publication. It is nontrivial task, as there might exist more than a single person matching a particular author's name. We explore two approaches to the authors disambiguation: a simple identification algorithm and a clustering algorithm. The first one is a simple rule-based algorithm that can only match publications to already existing researchers' profiles. The second one is a modified version of a Hierarchical Agglomerative Clustering algorithm, which is able not only to identify known author but also to create a new researchers' profile if there is no suitable candidate in the database.

A rule-based algorithm. A rule-based disambiguator takes into account the number of attributes that can be directly extracted from a publication d_i like names and surnames, co-authorship, affiliations, scientific fields and keywords. The algorithm works as illustrated in Figure 11. It takes a publication and retrieves its attributes. Next, the matching rules are sequentially executed for each author's name originating from the reference database. If any rule has sufficiently identified all authors of the publication, then the remaining rules are not executed. After applying all the rules, the next publication is considered irrespective of whether all authors in the previous publication were disambiguated or not.

$$\text{publication} \xrightarrow[retrieval]{features} \left.\begin{matrix}\text{features}\\ \text{candidate}\end{matrix}\right\} \xrightarrow[application]{rules} \text{authors}$$

Figure 11: A rule based disambiguation of authors realized by the information retrieval module.

The matching rules have been designed on a basis of our experience, available data, and preliminary experiments. They consider (1) names and surnames, (2) co-authorship, (3) affiliations, (4) scientific fields, (5) profiles of both a researcher and a publication.

Clustering. The second approach is an adapted version of Hierarchical Agglomerative Clustering (HAC) (Hastie et al., 2009). The algorithm groups publications and then assigns authors to these groups. In this way, the algorithm is able

to discover new profiles of unknown researchers. It uses all the attributes used by the rule-based algorithm as well as additional ones that takes into account data coming from publications, i.e. a year, a journal name, a conference name, a title, and an abstract. The clustering procedure iteratively analyzes succeeding names of scientists (Figure 12). At the beginning, a scientist's name is selected from the entire set of researchers. Concurrently, publications containing this author name are selected from the overall set of publications. Then, the HAC algorithm invokes the following steps: calculates similarities between the publications; iteratively clusters the most similar publications or its groups; creates a dendrogram and limits it to a defined cut-off level; if one necessary creates new researcher profiles and assigns to them publications according to the groups.

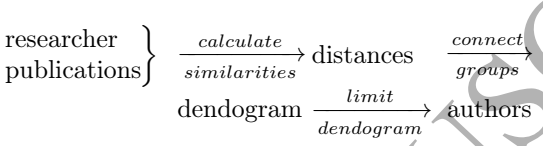


Figure 12: A clustering algorithm for disambiguation of authors in the information retrieval module.

The resemblance of a publications' pair d_i and d_j is expressed by a similarity measure $sim(d_i, d_j)$ that models a probability of two publications being authored by the same individual. The similarity is defined as a logistic function over a set of partial similarities s_l, $l = 1, .., L$:

$$sim(d_i, d_j) = \frac{1}{1 + e^{-\sum_{l=1}^{L} s_l}} \tag{6}$$

The partial similarities might by quantified by some additional model, could be assigned by an expert, or could result from some statistical analysis. We proposed four partial similarities, namely s_1 - identified author's names or its initials; s_2 - normalized difference between years in which compared works under analysis were published; s_3 - a journal name or a conference name; s_4 - co-authors, scientific fields, keywords, affiliations.

4.4. Keywords extraction

The keywords extraction retrieves the most relevant terms from a given document. Polish texts are explored by Polish Keyword Extractor (PKE) (Kozlowski and Protasiewicz, 2014), whereas the English texts are analyzed by Keyphrases Extraction Algorithm (KEA) (Witten et al., 2000). The process also involves automatic translation of keywords from Polish into English and vice versa by using publicly available dictionaries as well as Polish and English Wikipedia. In what follows, we describe in detail the PKE method. However, we omit the description of KEA since it could be found in the literature.

The PKE is a machine learning approach to keyword extraction from Polish scientific documents inspired by the Rapid Automatic Keyword Extraction (RAKE) (Rose et al., 2010) and KEA (Witten et al., 2000) algorithms. RAKE is an unsupervised, domain-independent, and language-independent method for

extracting keywords from individual documents. It is based on the observation that keywords frequently are compound words. KEA is a supervised method, which exploits Naive Bayes model to compute a probability of the term being a keyphrase.

The proposed algorithm is a single document-oriented method, so it analyzes each document separately rather than a whole set at once. Unlike many approaches, PKE is a knowledge-limited method what means that it does not use external knowledge resources. In contrast to the original methods, PKE is equipped with Polish lemmatizer, part-of-speech filters, and two candidate selection methods (Pattern-Recursive selector, PoS selector) as well as two candidate evaluation methods (supervised, unsupervised), what leads to four possible configurations.

The pattern-recursive selector is based on four ordered PoS patterns: noun-adjective, noun-unknown, noun-noun, noun, which are invoked recursively. The PoS selector is based on the pattern described by the complex regular expression. The text is split into sequences of contiguous words at phrase delimiters and stop word positions. Next, for each candidate are built its sub-combinations, which extend the number of potential keywords. All of the potential keywords are filtered out by the PoS regular expression:

$$(noun)(noun|adjective|unknown)^* \tag{7}$$

The unsupervised candidates evaluation approach is based on a scoring function $freq(w)^2$. Compound keyword's scores are the sum of its member's scores. Scored candidates are sorted, and top T terms are selected as keywords for the selected document. On the other hand, in the supervised evaluation approach, each candidate is categorized as a proper or non-proper keyword. Candidates are described by vectors of four features: TFxIDF, the first occurrence in an abstract, the first occurrence in a sentence, the size of a keyword. Overall, keywords are extracted by PKE algorithm following a sequence of steps:

1. A given text is split into sentences, so each sentence is represented as a sequence of words.
2. The words are normalized (lemmatized) and tagged with PoS properties.
3. Keyword candidates are selected to find the finite number of potential significant words.
4. Finally, the candidates are evaluated by models (a classifier or a statistic function), and a limited number of them with the highest scores is selected.

4.5. Recommendation

A recommendation algorithm is the essential part of a recommendation module. The algorithm is based on a cosine similarity regardless whether it uses keywords or a full-text index as inputs for similarity calculations. However, due to significant differences in these two ways of information representation, we describe two different algorithms, i.e. a keywords cosine similarity (Protasiewicz, 2014) and a full-text index as well as a combination of both of them.

A keywords cosine similarity. Let us assume that we have a document d describing, for instance, a project or an article. The task is to find the most relevant persons $l = 1, 2, ..., L$, either reviewers or experts, who are able to make substantive assessment of the document d. Moreover, we assume that the document is summarized, which comes in the form of a vector of terms $\mathbf{q} = \{q_i\}_{i=1}^{I}$ that defines a request either for reviewers or experts. On the other hand, a person l is defined by a feature vector $\mathbf{f}_l$ containing keywords k_m, $m = 1, 2, ..., M$ that defines person's expertise areas, and contains their weights w_m^n, $n = 1, 2, 3$ that define the membership degree of a keyword k_m to a person k, which is equivalent to person's expertise levels:

$$\mathbf{f}_l = \left[\left\{k_1, w_1^1, w_1^2, w_1^3\right\}, \left\{k_2, w_2^1, w_2^2, w_2^3\right\}, \\ ...., \left\{k_M, w_M^1, w_M^2, w_M^3\right\}\right]^T \tag{8}$$

The keywords k_m are selected from various sources, so a keyword can have many weights depending on its sources. Thus, their weights are constructed by some algorithms depending upon these sources. Firstly, we assume that expertise areas provided by a particular person is the most reliable keywords source. But since people are fallible in assessing others' expertise because of cognitive distortions, thus we decided that such keywords should have the highest possible weight $w_m^1 = 1$ to avoid misleading judgments. Secondly, the keywords may be provided by authors as index terms in their publications w_m^2 or may be retrieved from researchers' publications (from an abstract and a title) w_m^3.

In the second case, it is possible to designate keyword's weights algorithmically basing on publications. We assume that research areas could change during a scientific career; therefore, the keywords from older publications are less influential than those coming from recent works. They are exponentially smoothed respectively to distance between the present year y^{now} and the publication year y_m^{pub} of an article containing the keyword k_m, and an additional constant c:

$$w_m^{time} = e^{-(y_m^{pub} - y^{now})/c} \tag{9}$$

Moreover, the expertise level of a person may depend on how often the person publishes on a particular topic, which can be expressed as a keyword repetition count in the person's list of publications. But such straightforward assumption could be inaccurate since a keyword may originate from an old work or it might not be such important as other terms. Thus, additionally such weights are smoothed according to the formula:

$$\bigwedge_{n=2,3} w_m^n = \left(1 + e^{-b \cdot \sum_{y=1}^{Y} (w_m^{time} \cdot count_m^y \cdot z_m)}\right)^{-1} \tag{10}$$

where c_m^y is the repetition count of the publication in the year y that contains the keyword w_m. A variable z_m is a candidate ratio for $m - th$ keyword, which is assigned in two ways: $z_m = 1$ when the keyword is selected from index terms of publications ($n = 2$) or z_m is equal to the probability that is calculated during keywords extraction from publications ($n = 3$). This is done using Bayes' theorem to calculate whether a word is a suitable candidate for a keyword.

Having defined the problem, it is possible to express recommendation of reviewers and experts as a list sorted in a descending order according to the similarity measure (a simple cosine score) $c\_score_l(\mathbf{q}, \mathbf{f}_l)$ between the vector of terms $\mathbf{q}$ defining the query and feature vectors $\mathbf{f}_l$ describing reviewers:

$$c\_score_l(\mathbf{q}, \mathbf{f}_l) = \frac{\mathbf{q} \cdot \mathbf{f}_l}{|\mathbf{q}| \cdot |\mathbf{f}_l|} \tag{11}$$

It should be noted that for every keyword constituting problem q_i, $i = 1, 2..., I$ is assigned value 1, whereas for keywords from a feature vector k_m $m = 1, 2, ..., M$ is assigned the product of a weight and value 1 if a keyword appears in the researcher's profile or value 0 otherwise. Moreover, one keyword can have up to three weights (they can originate at the same time from index terms, an abstract, and can be provided by a person), in such case, the maximum possible weight is taken into account.

A full-text index. Full-text retrieval systems have become a popular way of providing support for text databases. The full-text model enables to locate the occurrences of any words, phrases in any document of the collection. The full-text search is distinguished from searches based on metadata or parts of the original texts represented in databases (such as titles, abstracts, selected sections, or bibliographic references). Currently, the alternatives such as semantic indexing are confined to very specific domains while most text retrieval systems are based on full-text models. Full-text-searching techniques became common in online bibliographic databases in the 1990s. Many websites and application programs (such as word processing software) provide full-text-search capabilities. Some web search engines, such as AltaVista, employ full-text-search techniques while others index only a portion of the web pages examined by their indexing systems (Baeza-Yates and Navarro, 2004).

The main component of a full-text retrieval system is a text search engine. The search engine examines all of the words in every stored document as it tries to match search criteria (text specified by a user). All searches are performed against an index built over the collection of documents. The index is a data structure designed to speed up searches. Most of the full-text search engines exploit indexes belonging to the family of indexes known as an inverted index. An inverted index is simply a list of all the words appearing in the text, where each word has attached a list of all its text positions, in increasing text order. Searching for a simple word in an inverted index is as easy as looking for it in the vocabulary and returning its list. Searching for a phrase (a set of obligatory words) requires fetching the lists of each word and intersecting them so as to find the places where they appear in the text (Baeza-Yates and Navarro, 2004).

At the core of full-text search engines architecture is the idea of a document containing fields of text. Let us assume that the expertise of each person $l = 1, 2, ..., L$ is described by a document that is a fields vector $\mathbf{p}_l = \left[p_l^1, p_l^2, ..., p_l^J\right]$, where each field p_l^j, $j = 1, 2, ..., J$ describes expertise of the person l represented by reviews, research projects, career, patents, publications, keywords, and so on.

In other words, it is a profile $\mathbf{p}_l$ of the person l. Recommendation based on full-text indexes assumes that a document representing of a person is persisted into the full text index. A similarity between a given query $\mathbf{q}$, which can be either a vector of keywords or a text, and person's profiles $\mathbf{p}_l$ is based on a cosine measure, but since it operates on texts, it covers more factors than a cosine measure between keywords. For example, Apache Lucene scoring formula is as follows (Lucene, 2015):

$$l\_score_l(\mathbf{q}, \mathbf{p}_l) = c_{\mathbf{q},\mathbf{p}_l} \cdot \|\mathbf{q}\| \cdot \sum_{i=1}^{I} \left(TF_{q_i,\mathbf{p}_l} \cdot IDF_{q_i}^2 \cdot b_{q_i} \cdot \|q_i, \mathbf{p}_l\| \right) \tag{12}$$

where $c_{\mathbf{q},\mathbf{p}_l}$ is a factor counted for each candidate person l, which express how many times terms q_i constituting a request $\mathbf{q}$ appear in a profile $\mathbf{p}_l$ defining the persons' expertise; $\|\mathbf{q}\|$ is the norm (usually Euclidean) of a request $\mathbf{q}$ for reviewers and experts. $TF_{q_i,\mathbf{p}_l}$ is the term's frequency which denotes how many times a single term q_i appears in the profile $\mathbf{p}_l$, whereas $IDF_{q_i}^2$ is an inverse document frequency. b_{q_i} is a boosting parameter that could be applied for each term q_i to underline or ignore its relevance. The last factor, a norm$\|q_i, \mathbf{p}_l\|$ depends on boosting parameters related to documents, their fields, and field lengths.

It should be underlined that the Lucene practical scoring formula (12) is the expansion of the simple cosine similarity measure. We have included such detailed description of this formula because we wanted to show how it works in the case of our recommendation task.

The combination of two measures. The ranks generated by the keywords cosine similarity may be joined with ranks provided by the full-text search engine. The final ordering is derived from the sum of the positions of each recommended expert and normalizing them to the range from [0, 1]:

$$score_l = \alpha \cdot c\_score_l(\mathbf{q}, \mathbf{f}_l) + \beta \cdot l\_score_l(\mathbf{q}, \mathbf{d}_l) \tag{13}$$

where $\alpha + \beta = 1$. It is possible to pay more attention to the first or the second component by adjusting the values of α and β parameters. if $\alpha = 0$ the ranking is based only on the full-text index, and while $\beta = 0$ the ranking is based only on scientists' keywords.

5. Implementation and illustrative examples

In what follows, we elaborate on pertinent implementation details of the system as well as include some selected results and comments regarding the individual algorithms as well as the entire system.

5.1. Technical architecture

The system consists of four layers, i.e. databases, access, business, and interface layers. Their functionalities and inter-connections are presented in

Figure 13. The database layer is composed of a typical relational database and a NoSQL database, which stores data to which one needs fast access. Access to data is provided by Java Persistence API with the exception of the NoSQL database, which offers direct access to its data. Moreover, unstructured data is indexed by a full-text search engine. The engine (Apache Lucene) provides a fast search of data by using text indexes. The business layer hosts all computations and controls users' actions. Services are responsible for data processing to response user requests, which can originate from web interfaces (people via www) or REST interfaces (other systems). In contrast, processes deal with computations like publications classification, authors disambiguation, keywords extraction, etc. The processes are initiated by a system administrator using a management console, or services, or according to a certain defined schedule.

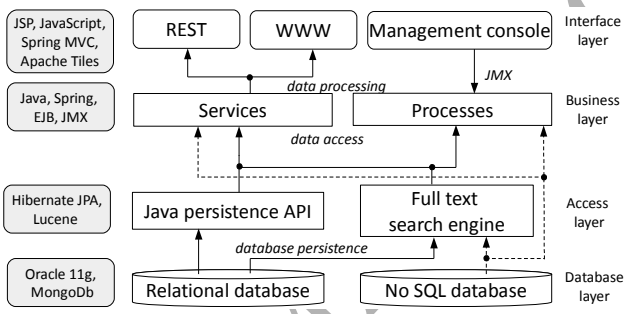


Figure 13: Overall layered architecture of the system.

From the functional perspective, the system architecture is modular (see Section 3). From the technical perspective, it is of multilayer nature. Each substantial part of the system is regarded as a separate module, whereas each layer corresponds to the technical function of the system. Although it is a typical construction of an information system, there are also three interesting solutions realized here. First, a relational database coexists with a NoSQL storage, so it is possible to accelerate time-consuming operations. Second, relational data access is combined with full-text access; these data can be browsed in users interfaces, as well as can be quickly searched and presented. Third, client services are separated from computational processes. In this way, the long-term computations do not affect the performance of services supporting short-term users' actions. Finally, due to the modular architecture, it is easy to maintain and develop the system because its parts can be considered separately.

5.2. Data acquisition

The data are acquired by the data extractors and the focused web crawlers, which are essential parts of the data acquisition module. Their algorithms have been presented in Subsection 4.1. Although theoretically they may be able to work with any kind of data regarding researchers, the implemented versions deal only with publications. These documents are characterized by attributes such as a title, authors, an abstract, index terms, a source, but without a guarantee that every attribute is of relevance for each publication.

The data extractors cope with sources of structured data such as file dump and web interfaces. Although each extractor uses a common programming interface, in addition, it has a unique parser dedicated to a specific data source to which it connects because the sources can be in any format. Then, the number of implemented extractors is equal to the number of data sources. They support sources such as DBLP, PubMed, CEJSH, and other local sources. The module provides a management interface allowing to configure, to run, and to monitor extractors. The execution of data extraction is realized in a series of the following steps:

1. Select a unique scientist's surname from a reference database (we have used the database of Polish scientists called *Nauka-Polska)*.
2. For the given name execute all implemented extractors to retrieve publications from all supported sources and possible connected with this name.
3. Store data in a local database and execute step 1 until there are some names to process.

The web crawler tackles unstructured data sources. These are mainly personal web pages that may contain publications. Implementation of the crawler involves not only data acquisition but also publications retrieval using a Conditional Random Fields model. To simplify a development task, we used the CRF model based on ParsCit (Kan et al., 2010). The crawler proceeds as follows:

1. Select a unique scientist's Uniform Resource Locator (URL) from a reference database (again we have used the database *Nauka-Polska* of Polish scientists).
2. Locate the URL on the Internet, crawl the web page and its subpages, and retrieve publications.
3. Store data in a local database and execute step 1 until there are available URLs to process.

We processed about 160 thousand scientist's names originating from the *Nauka-Polska* database. The data extractors and the web crawler collected about 5 million publications, however, almost all of them were acquired by the data extractors because the crawling of scientist's web pages turned out to be error prone and we found its effectiveness low.

We have to mention that any additional documents describing individuals' experiences are provided by them using the dedicated user interfaces.

5.3. Classification of publications

The classification models are a part of the information retrieval module. They are trained by the Multinomial Naïve Bayes (NMB) algorithm. As a result, a publication d_i is assigned to relevant classes c_j^l of a science classification model that is Ontology of Scientific Journals (OSJ). It is the three-level hierarchy containing 6 top-level domains $c_{j=1,..,6}^{l=1}$, 22 fields $c_{j_2=1,..,22}^{l=2}$, and 176 subfields $c_{j_3=1,..,176}^{l=3}$, however, we omit the subfields as being too detailed for our purposes. The classification models are meant to work on the dataset of publications

originating from our local database created by the data acquisition module. The documents can be in English or Polish, or they can mix these languages, so the input vectors for the classification models are formed by joining all publication attributes irrespectively of the language.

In two initial experiments, we assumed that all publications are in the same language. The issue of multilingualism of publications will be covered in the succeeding experiments.

The first experiment included classification of publications to top-level domains that are Applied Sciences, Art & Humanities, Economic & Social Sciences, Health Sciences, Natural Sciences, and General domain. We have checked how various ways of constructing a training set affect the classification results, i.e. proportional vs. equal distribution of data over scientific domains. In the proportional distribution, the number of training examples in a domain depends on the number of data, whereas in the equal distribution the number of training examples is the same in all domains. The results are shown in Table 2. We found out that there is no significant difference between the proportional and equal distribution of data over domains in the training set. Moreover, it turned out that the General domain has too ambiguous meaning, thus it should be excluded, whereas Natural Sciences and Applied Sciences are so similar that should be joined together.

Data representation	Precision
Proportional distribution	0.62
Equal distribution	0.64
Equal distribution and General domain omitted	0.75
Equal distribution and General domain omitted, and Natural Sciences joined with Applied Sciences	0.84

Table 2: The results of classification realized by the Multinomial Naive Bayes model at the first level of Ontology of Scientific Journals classification (science domains) depending on proportional and equal data representations in the training set as well as the domains modifications.

The next experiment involved classification of publications to science fields. Since they form the second level of the OSJ classification, there are two hierarchical classification models. The first one works on the domain level, however, we used the results from the previous experiment as there is no need to repeat that experiment. The second model classifies on the science fields level. A data set for the classification model at this level required manual improvements. In three cases, selected fields of science had to be joined as being too ambiguous. The results are shown in Table 3. Based on obtained results, we decided that the Bayesian hierarchical classifier comes as a viable implementation in the system. The distribution of training examples should be equal over the science domains as well as the science fields.

A maximum probability model and a multilingual classification system are

The fields in a domain	Precision
Applied Sciences	0.74
Art & Humanities	0.82
Economic & Social Sciences	0.84
Health Sciences	0.74
Natural Sciences	0.84

Table 3: The results of classification obtained at the second level of OSJ classification (science fields) expressed as an average precision over the science fields in their domains.

two approaches to classification multilingual and monolingual publications, which were discussed in Subsection 4.2. In order to evaluate the performance of these classifiers, we have conducted the following experiments. We trained two monolingual classifiers by using the MNB algorithm. The number of training samples in each top-level domain was equal to 10,000 in the case of Polish publications, and 25,000 in the case of English publications. The monolingual models generated vectors of probabilities, which constituted a training set for the multilingual decision module. This set was used to train and test an ensemble of logistic regression models (the decision module) working in the *one-vs-others* mode. The training set contained about 33,000 of publications containing Polish and English features at the same time. The results are shown in Table 4, where we display the average F-score and precision values. The multilingual system produces slightly better results in comparison to the maximum probability model.

Approach	Domains		Fields	
	F	P	F	P
Multilingual system	0.93	0.90	0.86	0.76
Maximum probability	0.92	0.92	0.80	0.77

Table 4: Average F-score (F) and average precision (P) obtained by the multilingual system and the maximum probability model in science domains and science fields.

The multilingual system combines monolingual probabilities in the multilingual decision module. Thus, it is robust to the problems of multilingualism of texts as well as disproportionate representation of texts among languages. The system is advisable for those cases when a document (publication) has parts written in various languages, however, it would work well with monolingual documents. Moreover, it does not require modification of dataset as it does a simple monolingual MNB model (in the experiments some domains were omitted, and some were joined). We believe that the proposed classification system can categorize huge data sets of multilingual and monolingual publications. The system is meant to boost the author disambiguation process.

5.4. Authors disambiguation

Authors of publications are identified by using a rule-based algorithm and a clustering algorithm, which were discussed in Subsection 4.3. The algorithms are

implemented in the form of the information retrieval module. Having designed and implemented both algorithms, we have conducted a series of experiments to check their performance.

First, we decided to test the rule-based algorithm. The data acquisition module has collected about 5 million publications to which correspond about 19 million authors that should be disambiguated. As it is the huge amount of data, it is impossible to label a test set manually. Thus, we applied a certain procedure that should select the representative and form a reliable test set:

- One scientist was selected for each scientific field, and authorship of all publications of this person was verified; in this way 63 researchers have been verified.
- The most popular last names with initials were selected from the whole set and authorship of all publications of these persons was checked; in this way 48 researchers have been verified.

We executed the rule-based algorithm using four rules, namely (1) names and surnames, (2) co-authorship, (3) affiliations, (4) scientific fields, on the test set contained 111 authors and their 2,921 publications. The algorithm made 34 incorrect assignments of authors to publications. Precision reached 99% and recall was 65%. After that, we decided to check the last rule, namely (5) profiles of both a researcher and a publication, but in a standalone manner algorithm without considering other rules. It achieved precision equal to 82%, so it rather would not improve the overall performance because this rule prefers scientists containing more publications than others.

Second, we have tested the HAC clustering algorithm. The experiments were conducted for 50 randomly selected surnames, where for each surname the number of scientists varies from 100 to 200. The test set contained 20,000 of publications. The minimal size of cluster was set to three publications. The results of experiments are shown in Table 5, which contains the values of precision, recall, and F-score with regard to different cut-off levels of the constructed dendrograms.

Cut-off	Precision	Recall	F-score
5	97.96	54.27	69.84
10	98.87	34.04	50.64
20	99.42	9.98	18.14
30	99.35	2.99	5.81

Table 5: The results of experiments of hierarchical clustering for different cut-off levels.

The precision of the disambiguation process is very high irrespectively of the cut-off level. However, recall decreases dramatically with the increase of the cut-off level, so many authors have not been identified. F-score shows the trade-off between these two conflicting measures. In this case, the most important is not to make wrong assignments, because it is not proper to attach a researcher

to a work of another researcher. Thus, we decided to use the cut-off level equal to 20, which gives the sufficient levels precision and recall.

The rule-based algorithm is reflected of some domain knowledge, whereas the clustering approach algorithmically reveals structure from groups of publications. Both algorithms demonstrate their usefulness for possible practical deployment. The rule-based algorithm runs faster and is more precise than the clustering algorithm. However, the clustering algorithm can identify even unknown yet authors, whereas, the rule-based algorithm can assign to publications only researchers whose profiles exist in a database.

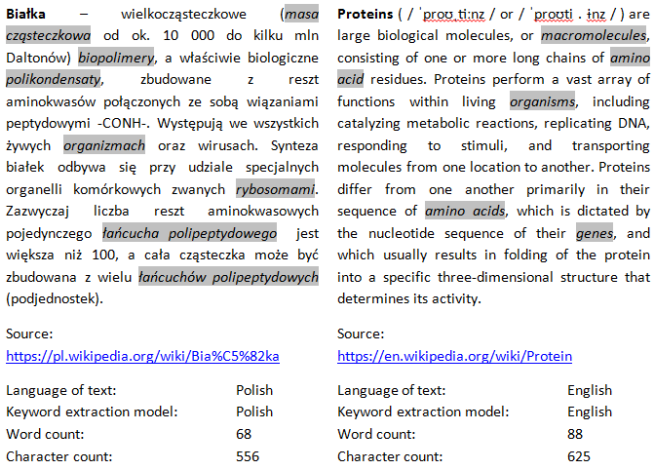
5.5. Keyword extraction

The keyword extraction concerns English and Polish texts. Since tools for English texts are well known, there is no need to discuss them; the reader may refer to (Rose et al., 2010, Witten et al., 2000). The keywords from texts in Polish are extracted by a new algorithm named Polish Keyword Extractor (PKE), which deserves a detailed discussion.

The PKE algorithm has been tested on a set of abstracts of papers written in Polish. Unfortunately, there is no a simple and reliable way to judge whether the keywords retrieved from a given text are the most appropriate because the assessment is done manually and involves a certain dose of subjective judgment. An estimation approach has been considered with this regard: we decided to juxtapose the keywords extracted from abstracts with the keywords given by authors as index terms of these publications. The best results were reported by the pair of a pattern-recursive selector and supervised evaluator. The experiments have also shown that PKE achieves better quality (precision, recall, F-measure) than RAKE and KEA.

Figure 14 shows two examples of keywords extraction. On the left is presented the result of processing a text in Polish by the PKE, whereas, on the right-hand side is shown the output of the KEA, applied to the analysis of a text in English. The processed texts come from Wikipedia, and the keywords retrieved by both algorithms are marked in gray. These examples depict clearly that it is difficult to assess the quality of keywords extraction because someone would agree that selected words are relevant and the most important, but there might be some other opinion as well. For instance, one might have selected the term *nuleido sequence* instead of the *genres*, or the word *polikondesaty* instead of *reszt aminokwasów* (in Polish).

The quality of keyword extraction is based on our judgment and experience that particular words encapsulate a given text optimally. Nonetheless, to some extent one can have confidence in the judgment of experts. Thus, we are able to measure the PKE quality. Based on the experimental evidence the PKE outperforms both the RAKE and KEA making this algorithm a useful mechanism in the information retrieval module. On the other hand, taking into consideration the ambiguity of human assessments, we are going to treat the keywords extracted by the algorithm as the suggested words only that may properly describe a particular researcher's experience. Having such suggestions, scientists



Białka – wielkocząsteczkowe (*masa cząsteczkowa* od ok. 10 000 do kilku mln Daltonów) *biopolimery*, a właściwie biologiczne *polikondensaty*, zbudowane z reszt aminokwasów połączonych ze sobą wiązaniami peptydowymi -CONH-. Występują we wszystkich żywych *organizmach* oraz wirusach. Synteza białek odbywa się przy udziale specjalnych organelli komórkowych zwanych *rybosomami*. Zazwyczaj liczba reszt aminokwasowych pojedynczego *łańcucha polipeptydowego* jest większa niż 100, a cała cząsteczka może być zbudowana z wielu *łańcuchów polipeptydowych* (podjednostek).

Source:
https://pl.wikipedia.org/wiki/Bia%C5%82ka

Language of text:	Polish
Keyword extraction model:	Polish
Word count:	68
Character count:	556

Proteins (/ ˈprəʊ.tiːnz / or / ˈprəʊti . ɪnz /) are large biological molecules, or *macromolecules*, consisting of one or more long chains of *amino acid* residues. Proteins perform a vast array of functions within living *organisms*, including catalyzing metabolic reactions, replicating DNA, responding to stimuli, and transporting molecules from one location to another. Proteins differ from one another primarily in their sequence of *amino acids*, which is dictated by the nucleotide sequence of their *genes*, and which usually results in folding of the protein into a specific three-dimensional structure that determines its activity.

Source:
https://en.wikipedia.org/wiki/Protein

Language of text:	English
Keyword extraction model:	English
Word count:	88
Character count:	625

Figure 14: An example of using the Polish Keyword Extractor for a Polish text (on the left) and the Keyphrases Extraction Algorithm for an English text (on the right); the keywords are highlighted.

can finally decide whether the words describe their experience by marking them as their specializations.

5.6. Recommendation

The recommendation algorithms were presented in Subsection 4.5. They are implemented in the recommendation module, which forms a final phase of the whole recommender system. We have to note that it is impossible to provide a comprehensive validation of this system because the results of reviewers or experts selection usually cannot be available publicly. Therefore, we present the cases that somehow are close to reality as well as discuss selected interesting issues arising from the system practical customization and deployment in the National Center for Research and Development in Poland (NCBR).

5.6.1. A test case of the keywords cosine similarity algorithm

In order to validate the system we have prepared a dataset containing two descriptions of research & development projects and five expert's profiles. The projects were selected from the projects database of FP7 Programme of European Commission[1]. The first project shown on the list comes from the energy domain while the second one is related to the environment domain. Five terms for each project were extracted. The projects and their keywords are shown in Table 6.

The profiles were created from the database of reviewers and experts, and

[1] http://cordis.europa.eu/projects/home_en.html, 2015-11-18

	Project name	Domain	Keywords (weights)
1	All-oxide photo-voltaic cells	energy	solar cell (1), solar energy (1), photovoltaics (0,841), semiconductor (0,703), data mining (0,693)
2	Olive oil waste as a fuel	environ-ment	olive oil (1), renewable energy (0,814), biomass (0,692), biodiesel (0,664), fossil fuel (0,6)

Table 6: The projects selected for validation of the system. There are specified project names and corresponding domains. The last column contains the keywords and their weights that were extracted from project descriptions by the information retrieval module; project 1 - http://cordis.europa.eu/result/rcn/159946_en.html, project 2 - http://cordis.europa.eu/result/rcn/159944_en.html.

their publications[2]. The experts represent five domains, i.e. energy, environment, health, security, information & communication technologies (ICT).

The rankings for both projects were created using cosine measure. The recommended experts are presented below.

1. Project "All-oxide photovoltaic cells" (the energy domain)

   Rank 1 Nawojka (energy)

   Rank 2 Carlos (ICT)

   Rank 3 Alan (security)

2. Project "Olive oil waste as a fuel" (the environment domain)

   Rank 1 Nawojka (energy)

   Rank 2 John (environment)

What is interesting that the first expert (Nawojka) selected for the project in the environment domain represents the energy domain, whereas the second expert (John) represents the environment domain. What is more, the experts chosen for the project in the energy domain are professionals in various domains, i.e. energy (Nawojka), ICT (Carlos), and security (Alan). As we see, Nawojka seems to be the most suitable expert for both projects, but Ruth does not suit to any of them. Of course, these examples are of illustrative nature, nevertheless they show that experts selection based on keywords and cosine similarity requires that experts and projects should be described by many keywords and so keywords have to be properly weighted.

[2]http://sssr.opi.org.pl

	Expert	Domain	Keywords (weights)
1	Nawojka	energy	renewable energy (1) photoluminescence (1) solar energy (1) energy consumption (0,5) biomass (0,3)
2	John	environment	environment protection (1) ecology (1) sustainable development (1) renewable energy (0,5) biomass (0,2)
3	Ruth	health	health care (1) mental health (1) public health (0,5) cancer (0,2) adolescent (0,1)
4	Alan	security	internet (1) security requirement (1) cryptography(0,5) authentication (0,5) data mining (0,2)
5	Carlos	ICT	big data (1) data mining (0,8) machine learning (0,7) information system (0,5) internet (0.5)

Table 7: The experts created for validation of the system.

5.6.2. Deployment of the system

The system was customized and deployed in the NCBR. The project was aimed at improving the selection process of the reviewers for the Polish science funding body. For this purpose, we combined the recommendation algorithm based on the keywords cosine similarity with a full-text index engine.

A full-text index engine. In the proposed approach, we use the Lucene search engine. Apache Lucene is a high-performance, full-featured text search engine library written in Java. It is a technology suitable for nearly any application that requires full-text search, especially of the cross-platform nature. At the core of Lucene's logical architecture is the idea of a document containing fields of text. In our case, the document represents an expert, where each field describes its achievements. In particular, the fields are previous reviews, R&D and commercial projects, previous jobs, a curriculum vitae, patents, publications, and

keywords. Each document representing one expert is stored in the Lucene index. Search engine returns ranked results (experts). The ranked reviewers are further combined with the results retrieved from the recommendations based on cosine similarity.

Boosting fields of the full-text index. Boosting queries is done by weighting document's fields according to the scoring formula (12). Namely, the fields representing publications have a weight equal to 1.5, a reviews weight is equal to 3.0, and the rest of fields have default weights equal to 1.0. This step is done to give higher priority to data concerning articles and reviews written by the expert.

After several trials, we realized that NCBR granting bodies prefer reviewers and experts possessing specific features like university degree and scientific fields. Thus, such scientists are more likely recommended that some others. We have applied some additional boosting factors assuming that a field without boosting is multiplied by 1:

- expert possessing a university degree: professor - 1.00; doctor of science (DSci) - 0.98; PhD - 0.95; MSc - 0.70; the lack of a title - 0.70.
- matching of scientific fields between a scientist and an object to assess: exact match - 1, 30; an indirect match - 1.00; no match - rejection.

If there is no match of scientific fields between a scientist and an object to assess, this person is not taken into account as a reviewer or an expert regardless of other criteria.

The NCBR case study. The recommendation model was evaluated using a manually prepared list of reviewers for the 20 keywords queries. Cooperating with the NCBR specialists, we built 20 queries concerning different domains of science (e.g., "machine learning, bio-engineering", "sociology, economic sociology", "civil engineering, bridges"). Next, for each of those queries we selected top 100 reviewers recommended by the system. Such prepared initial lists of reviewers were evaluated by NCBR employees. They were aimed at creating a proper ordering of the proposed reviewers (called further gold standard). It means that each of reviewers has its original position (from the automatic process), and the corrected position coming from the manual assessment.

We define the function of punishment, which is a sum of weighted differences between recommended ranks and the gold standard. The differences between automatic and manual ranks are weighted in three different ways:

- reviewer is at the same position on both rankings - the weight is equal to 1.00;
- reviewer is not pertinent to the query (it was crossed out by NCBR specialist as not suitable as a reviewer) - the weight is equal to -100.00;
- reviewer is at a position below top 30 experts but manually was ranked in the first 30 - the weight is equal to -10.00;

- reviewer is at a position below top 30 experts and manually was also ranked in the second part of the ordered list - the weight is equal to -2.00.

Each recommendation was evaluated using the described above function of punishment. We introduced several improvements to maximize this objective function. First of all, we introduced additional recommendation method based on full-text search. Next, we tuned it by using boosting index fields (reviews, publications), also, we gave more priority to experts from NCBR databases and those with a higher scientific degree and finally we combined the both ranking results. We processed iteratively; each enhancement was verified against the function of punishment.

5.6.3. Discussion

Two recommendation algorithms have been used in this study. The first algorithm is based on a cosine similarity between keywords representing a problem. The keywords present in the expertise profiles represent expertise areas, whereas its weights represent expertise levels. It is a simple summarization of the track record of the scientist. A request for reviewers and experts is also summarized and represented as a list of keywords. Thanks to it the ranking formula is straightforward and easily understandable what is important when we have to explain the position of a particular expert on the recommendation list. On the other hand, the recommendation quality is influenced by both precision and recall of information retrieval processes, where a text summarization has been performed. When queries and expertises are represented by keywords, we lose some information because of two factors: (i) information granularity that rely on this how general the keywords are, and (ii) errors meaning the some keywords might be not introduced whereas others are unnecessary.

To address these concerns, we introduced a full-text index to form a supportive recommendation tool. Although it is also based on the cosine similarity, it works on the entire text representing expertise of scientists, so the problem of losing information is eliminated. Moreover, full-text search engines are usually developed by wide, open source communities such as Apache Lucene. Thus, its implementation is rather reliable, what was essential for the practical deployment of the recommendation system. In contrast to the cosine measure, they use complicated similarity formulas, which are rather unintelligible for a wide audience so that recommendations would be perceived as being unintuitive and muddled. Additionally, too detailed information granularity can lead to focusing on unimportant achievements of scientist and may result in wrong recommendations.

The raised issues of the incorrectly selected level of information granularity and comprehensibility of recommendation formulas led us to propose a hybrid solution, where both recommendation algorithms are used at the same time, but their influence on final recommendations is weighted. A user can then consciously decide which method is more appropriate for a particular task.

6. Conclusions

In this study, we have proposed the architecture of the content-based recommender system for selection of reviewers (experts). The design stressed a modular nature of the architecture to facilitate the flexibility and maintainability of the system. The system comprises three well-defined functional modules, namely (i) data acquisition, (ii) information retrieval, and (iii) recommendation mechanisms. The data acquisition module helps to retrieve data from structured sources, whereas a focused crawler (using the model of Conditional Random Fields) deals with unstructured sources. Classification of publications is realized by simple Naive Bayes classifiers as well as by a multilingual classification system that is composed of monolingual classifiers and a multilingual decision module. Author disambiguation is carried by a rule-based algorithm and a modified version of Hierarchical Agglomerative Clustering. Both algorithms demonstrated their usefulness for possible practical deployment. The rule-based algorithm runs faster and produces more precise results than those produced by the clustering algorithm. However, the clustering algorithm can identify even unknown yet authors, whereas, the rule-based algorithm can assign to publications only researchers whose profiles exist in a database. Based on the acquired experimental evidence, the PKE outperforms both the RAKE and KEA in the case of Polish texts, making this algorithm a useful mechanism in the information retrieval module. The ranking mechanism offers the user an additional support in selecting potential reviewers; it is of particular importance by noting that the system supports decision-making processes.

There are a number of interesting research pursuits exhibiting direct application advantages, namely (i) identifying and expressing consistency of data sources, (ii) building a multilingual version of the systems, (iii) enhancing the quality of user interface, and (iv) incorporating advanced ranging schemes and further facilitating user-system interaction.

References

Aleman-Meza, B., Bojars, U., Boley, H., Breslin, J. G., Mochol, M., Nixon, L. J., Polleres, A., Zhdanova, A. V., 2007a. Combining rdf vocabularies for expert finding. In: In Proceedings of the 4th European Semantic Web Conference (ESWC2007), number 4519 in Lecture Notes in Computer Science. Springer, pp. 235–250.

Aleman-Meza, B., Hakimpour, F., Arpinar, I. B., Sheth, A. P., Sep. 2007b. Swetodblp ontology of computer science publications. Web Semantics: Science, Services and Agents on the World Wide Web 5 (3), 151–155.

August, D., Muraskin, L. D., 1999. Strengthening the standards: Recommendations for oeri peer review. Summary report. Prepared for the National Educational Research Policy and Priorities Board, US Department of Education.

Azar, A., Sebt, M., Ahmadi, P., Rajaeian, A., 2013. A model for personnel selection with a data mining approach: A case study in a commercial bank. SA Journal of Human Resource Management 11 (1).

Baeza-Yates, R., Navarro, G., 2004. Text searching: Theory and practice, Formal languages and applications.

Basu, C., Hirsh, H., Cohen, W. W., 2001. Technical paper recommendation: A study in combining multiple information sources. Journal of Artificial Intelligence Research 14, 231–252.

Bobadilla, J., Ortega, F., Hernando, A., Alcalá, J., 2011. Improving collaborative filtering recommender system results and performance using genetic algorithms. Knowledge-based systems 24 (8), 1310–1316.

Bobadilla, J., Ortega, F., Hernando, A., Bernal, J., 2012. A collaborative filtering approach to mitigate the new user cold start problem. Knowledge-Based Systems 26, 225–238.

Bobadilla, J., Ortega, F., Hernando, A., Glez-de Rivera, G., 2013a. A similarity metric designed to speed up, using hardware, the recommender systems k-nearest neighbors algorithm. Knowledge-Based Systems 51, 27–34.

Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A., 2013b. Recommender systems survey. Knowledge-Based Systems 46, 109–132.

Bobadilla, J., Serradilla, F., Bernal, J., 2010. A new collaborative filtering metric that improves the behavior of recommender systems. Knowledge-Based Systems 23 (6), 520–528.

Bobadilla, J., Serradilla, F., Hernando, A., et al., 2009. Collaborative filtering adapted to recommender systems of e-learning. Knowledge-Based Systems 22 (4), 261–265.

Bornmann, L., Daniel, H.-D., 2009. Reviewer and editor biases in journal peer review: an investigation of manuscript refereeing at angewandte chemie international edition. Research Evaluation 18 (4), 262–272.

Chien, C.-F., Chen, L.-F., 2008. Data mining to improve personnel selection and enhance human capital: A case study in high-technology industry. Expert Systems with Applications 34 (1), 280 – 290.

Cook, W. D., Golany, B., Kress, M., Penn, M., Raviv, T., 2005. Optimal allocation of proposals to reviewers to facilitate effective ranking. Management Science 51 (4), 655–661.

Cordobés de la Calle, H., Chiroque, L. F., Fernández Anta, A., García Leiva, R. A., Morere, P., Ornella, L., Pérez, F., Santos, A., 2015. Empirical comparison of graph-based recommendation engines for an apps ecosystem. International Journal of Interactive Multimedia and Artificial Intelligence 3 (2), 33–39.

Crespo, R. G., Martínez, O. S., Lovelle, J. M. C., García-Bustelo, B. C. P., Gayo, J. E. L., De Pablos, P. O., 2011. Recommendation system based on user interaction data applied to intelligent electronic books. Computers in Human Behavior 27 (4), 1445–1449.

Eisenhart, M., 2002. The paradox of peer review: Admitting too much or allowing too little? Research in Science Education 32 (2), 241–255.

Fan, Z.-P., Chen, Y., Ma, J., Zhu, Y., 2009. Decision support for proposal grouping: A hybrid approach using knowledge rule and genetic algorithm. Expert Systems with Applications 36 (2, Part 1), 1004–1013.

Flach, P. A., Spiegler, S., Golénia, B., Price, S., Guiver, J., Herbrich, R., Graepel, T., J.Zaki, M., May 2010. Novel tools to streamline the conference review process: Experiences from sigkdd'09. SIGKDD Explorations Newsletter 11 (2), 63–67.

Goldsmith, J., Sloan, R. H., 2007. The ai conference paper assignment problem. In: In Proceedings AAAI Workshop on Preference Handling for Artificial Intelligence, Vancouver. pp. 53–57.

Green, S. M., Callaham, M. L., 2011. Implementation of a journal peer reviewer stratification system based on quality and reliability. Annals of Emergency Medicine 57 (2), 149 – 152.e4.

Harper, P. R., de Senna, V., Vieira, I. T., Shahani, A. K., May 2005. A genetic algorithm for the project assignment problem. Comput. Oper. Res. 32 (5), 1255–1265.

Hartvigsen, D., Wei, J. C., Czuchlewski, R., 1999. The conference paper-reviewer assignment problem. Decision Sciences 30 (3), 865–876.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning. Springer,, New York.

Hemlin, S., 2009. Peer review agreement or peer review disagreement: Which is better? Journal of Psychology of Science and Technology 2 (1), 5–12.

Herlocker, J. L., Konstan, J. A., Borchers, A., Riedl, J., 1999. An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pp. 230–237.

Hojat, M., Rosenzweig, S., 2004. Editorial. journal peer review in integrative medicine discipline. Seminars in Integrative Medicine 2 (1).

Jacoby, L. L., Kelley, C., Brown, J., Jasechko, J., 1989. Becoming famous overnight: Limits on the ability to avoid unconscious influences of the past. Journal of personality and social psychology 56 (3), 326–338.

Kan, M.-Y., Luong, M.-T., Nguyen, T. D., Oct. 2010. Logical structure recovery in scholarly articles with rich document features. Int. J. Digit. Library Syst. 1 (4), 1–23.

Kolasa, T., Krol, D., 2011. A survey of algorithms for paper-reviewer assignment problem. IETE Technical Review 28 (2), 123–134.

Kozlowski, M., Protasiewicz, J., 2014. Automatic extraction of keywords from polish abstracts. In: 4th Young Linguists' Meeting in Poznań, Volume: Book of Abstracts. pp. 56–57.

Langfeldt, L., 2004. Expert panels evaluating research: decision-making and sources of bias. Research Evaluation 13 (1), 51–62.

Li, L., Wang, Y., Liu, G., Wang, M., Wu, X., 2015. Context-aware reviewer assignment for trust enhanced peer review. PloS one 10 (6).

Liu, P., Dew, P., 2004. Using semantic web technologies to improve expertise matching within academia. In: Proceedings of I-KNOW, Graz, Austria. pp. 70–378.

Lucene, 2015. Apache lucene - similarity measure.
URL `http://lucene.apache.org/core/3\_0\_3/api/all/org/apache/lucene/search/Similarity.html`

Marsh, H. W., Jayasinghe, U. W., Bond, N. W., 2008. Improving the peer-review process for grant applications: Reliability, validity, bias, and generalizability. The American Psychologist 63 (3), 160–168.

Martínez, Ó. S., Bustelo, B. C. P. G., Crespo, R. G., Franco, E. T., 2009. Using recommendation system for e-learning environments at degree level. IJIMAI 1 (2), 67–70.

Merelo-Guervos, J. J., Castillo-Valdivieso, P., 2004. Conference paper assignment using a combined greedy/evolutionary algorithm. In: Yao, X., Burke, E. K., Lozano, J. A., Smith, J., Merelo-Guervós, J. J., Bullinaria, J. A., Rowe, J. E., Tiòo, P., Kabán, A., (Eds.), H.-P. S. (Eds.), Parallel Problem Solving from Nature - PPSN VIII. Vol. 3242 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 602–611.

Mishra, D., Singh, S. K., 2011. Taxonomy - based discovery of experts and collaboration networks. VSRD International Journal of Computer Science & Information Technology I (10), 698 – 710.

Papagelis, M., Plexousakis, D., Nikolaou, P. N., 2005. Confious: Managing the electronic submission and reviewing process of scientific conferences. In: Ngu, A. H., Kitsuregawa, M., Neuhold, E. J., Chung, J.-Y., Sheng, Q. Z. (Eds.), Web Information Systems Engineering - WISE 2005. Vol. 3806 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 711–720.

Protasiewicz, J., 2014. A support system for selection of reviewers. In: Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on. IEEE, pp. 3062–3065.

Protasiewicz, J., Stanisławek, T., Dadas, S., 2015. Monolingual and multilingual approaches for classification of multilingual documents.

Rivara, F. P., Cummings, P., Ringold, S., Bergman, A. B., Joffe, A., Christakis, D. A., 2007. A comparison of reviewers selected by editors and reviewers suggested by authors. The Journal of pediatrics 151 (2), 202–205.

Rodriguez, M. A., Bollen, J., 2008. An algorithm to determine peer-reviewers. In: Proceedings of the 17th ACM conference on Information and knowledge management. CIKM '08. ACM, New York, NY, USA, pp. 319–328.

Rodriguez, M. A., Bollen, J., de Sompel, H. V., 2006. The convergence of digital-libraries and the peer-review process. Journal of Information Science 32 (2), 149–159.

Rose, S., Engel, D., Cramer, N., Cowley, W., 2010. Automatic keyword extraction from individual documents. In: Mining: Applications and Theory. pp. 19–37.

Ryabokon, A., Polleres, A., Friedrich, G., Falkner, A. A., Haselböck, A., Schreiner, H., 2012. (re)configuration using web data: A case study on the reviewer assignment problem. In: Krötzsch, M., Straccia, U. (Eds.), Web Reasoning and Rule Systems. Vol. 7497 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 258–261.

Song, X., Tseng, B. L., Lin, C.-Y., Sun, M.-T., 2005. Expertisenet: Relational and evolutionary expert modeling. In: Ardissono, L., Brna, P., Mitrovic, A. (Eds.), User Modeling 2005. Vol. 3538 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 99–108.

Spier, R., 2002. The history of the peer-review process. Trends in Biotechnology 20 (8), 357–358.

Sun, Y.-H., Ma, J., Fan, Z.-P., Wang, J., 2008. A hybrid knowledge and model approach for reviewer assignment. Expert Systems with Applications 34 (2), 817–824.

Sutton, C., McCallum, A., 2011. An introduction to conditional random fields. Machine Learning 4 (4), 267–373.

Tayal, D. K., Saxena, P., Sharma, A., Khanna, G., Gupta, S., 2013. New method for solving reviewer assignment problem using type-2 fuzzy sets and fuzzy functions. Applied Intelligence, 1–20.

Tian, Q., Ma, J., Liang, J., Kwok, R. C., Liu, O., 2005. An organizational decision support system for effective & project selection. Decision Support Systems 39 (3), 403–413.

Tian, Q., Maa, J., Liua, O., 2002. A hybrid knowledge and model system for r&d project selection. Expert Systems with Applications 39 (3), 265–271.

Tversky, A., Kahneman, D., 1974. Judgment under uncertainty: Heuristics and biases. Science 185 (4157), 1124–1131.

Wang, F., Zhou, S., Shi, N., 2013. Group-to-group reviewer assignment problem. Computers & Operations Research 40 (5), 1351 – 1362.

Witten, I., Paynter, G., Frank, E., Nevill-Manning, C. G. . C., 2000. Kea: Practical automatic keyphrase extraction. Working Paper 00/5, Department of Computer Science, The University of Waikato.

Xu, Y., Ma, J., Sun, Y.-H., Hao, G., Zhao, W. X. D., 2010. A decision support approach for assigning reviewers to proposals. Expert Systems with Applications 37 (10), 6948–6956.