Empowering Bridging Term Discovery for Cross-Domain Literature Mining in the TextFlows Platform

Matic Perovšek[1,2](✉), Matjaž Juršič[1,2], Bojan Cestnik[1,3], and Nada Lavrač[1,2,4]

[1] Jožef Stefan Institute, Ljubljana, Slovenia
matic.perovsek@ijs.si
[2] Jožef Stefan International Postgraduate School, Ljubljana, Slovenia
[3] Temida d.o.o, Ljubljana, Slovenia
[4] University of Nova Gorica, Nova Gorica, Slovenia

Abstract. Given its immense growth, scientific literature can be explored to reveal new discoveries, based on yet uncovered relations between knowledge from different, relatively isolated fields of research specialization. This chapter proposes a bisociation-based text mining approach, which shows to be effective for cross-domain knowledge discovery. The proposed cross-domain literature mining functionality, including text acquisition, text preprocessing, and bisociative cross-domain literature mining facilities, is made publicly available within a new browser-based workflow execution engine TextFlows, which supports visual construction and execution of text mining and natural language processing (NLP) workflows. To support bisociative cross-domain literature mining, the TextFlows platform includes implementations of several elementary and ensemble heuristics that guide the expert in the process of exploring new cross-context bridging terms. We have extended the TextFlows platform with several components, which—together with document exploration and visualization features of the CrossBee human-computer interface—make it a powerful, user-friendly text analysis tool for exploratory cross-domain knowledge discovery. Another novelty of the developed technology is the enabled use of controlled vocabularies to improve bridging term extraction. The potential of the developed functionality was showcased in two medical benchmark domains.

Keywords: Literature mining · Literature-based discovery · Cross-context linking terms · Creativity support tools · Human-computer interaction · Workflows

1 Introduction

Understanding complex phenomena and solving difficult problems often requires knowledge from different domains to be combined and cross-domain associations to be taken into account. These kinds of context crossing associations, called *bisociations* [1], are often needed for creative, innovative discoveries.

A. Holzinger (Ed.): ML for Health Informatics, LNAI 9605, pp. 59–98, 2016.
DOI: 10.1007/978-3-319-50478-0_4

Bisociative knowledge discovery is a challenging task motivated by a trend of over-specialization in research and development, which usually results in deep—but relatively isolated—knowledge islands. Scientific literature too often remains closed and cited only in professional sub-communities. In addition, the information that is related across different contexts is difficult to identify using associative approaches, like the standard association rule learning [2] known from the data mining and machine learning literature. Therefore, the ability of literature mining methods and software tools to support the experts in their knowledge discovery processes—especially in searching for yet unexplored connections between different domains—is becoming increasingly important. Cross-domain literature mining is closely related to bisociative knowledge discovery as defined in [3]. Assuming two different domains of interest, a crucial step in cross-domain knowledge discovery is the identification of interesting bridging terms (B-terms), appearing in both literatures, which carry the potential of revealing the links connecting the two domains.

This chapter presents a powerful approach to literature based cross-context knowledge discovery that supports the process of bridging term extraction. The developed methodology helps the experts in searching for hidden links that connect seemingly unrelated domains. The main novelty of the presented approach is the combination of document acquisition and text preprocessing facilities with a new facility for term extraction through ensemble-based ranking of terms according to their bisociative potential, which may contribute to novel cross-domain discoveries. The proposed methodology is implemented in a web-based text mining platform TextFlows[1]. To this end, the TextFlows platform was connected to the human-computer interface of system CrossBee [4,5]. In the methodology presented in this chapter, the CrossBee web application—which we originally developed as an off-the-shelf solution for finding bisociations bridging two domains—is used as a user interface to facilitate bridging term discovery through sophisticated document visualization and exploration. This work proposes a further extension of the methodology by facilitating the use of controlled vocabularies, enhancing the heuristics capability to rank the actual B-terms at the top of the ranked term list. With all these features, the TextFlows platform, which now includes the reusable text analytics workflows combined with the CrossBee document exploration interface, has become a publicly available creativity support tool (CST), supporting creative discovery of new cross-domain hypotheses.

The chapter is organized as follows. Section 2 provides a brief glossary of key terms that will facilitate a common understanding of the main topics presented here. Section 3 presents the state-of-the-art in the area of literature-based discovery. Section 4 illustrates the problem of bridging term ranking and B-term exploration through a use case scenario, followed by an overview of the methodology. Section 5 comprises the core contribution of this chapter. The TextFlows

[1] Our new text mining platform, named TextFlows, is publicly available for use at http://textflows.org. The source code (open sourced under the MIT Licence) is available at https://github.com/xflows/textflows. Detailed installation instructions are provided with the source code.

platform, acting as the enabling technology for implementing the developed cross-domain link discovery approach, is described in Sect. 5.1. The elementary and ensemble heuristics used in bridging term discovery are described in Sect. 5.2. Section 5.3 presents details of document acquisition, text preprocessing and literature based discovery workflows implemented in TextFlows. Controlled vocabulary extension of the methodology is presented in Sect. 5.4. Evaluation of the developed methodology on two medical benchmark problems is provided in Sect. 6, Finally, Sect. 7 concludes with a summary of most important features of the presented approach and some directions for further work.

2 Glossary

Bisociation: the combination of knowledge from seemingly unrelated domains into novel cross-domain knowledge.

Bridging term: a term common to two disjoint domains, which is a candidate for the discovery of new knowledge or for formulation of new hypotheses, acting as a "bridge" between the two domains.

Literature-based discovery: using academic literature to find previously uncovered connections in existing domain knowledge.

Outlier detection: finding irregular or unusual data instances (documents in the case of literature mining) that do not conform to the expected distribution.

3 State-of-the-Art

According to Koestler [1], bisociative thinking occurs when a problem, idea, event or situation is perceived simultaneously in two or more "matrices of though" or domains. When two matrices of thought interact with each other, the result is either their fusion in a novel intellectual synthesis or their confrontation in a new aesthetic experience. He regarded many different mental phenomena that are based on comparison (such as analogies, metaphors, jokes, identification, anthropomorphism, and so on) as special cases of bisociation. More recently, this work was followed by the researchers interested in so-called bisociative knowledge discovery [6], where—according to Berthold—two concepts are bisociated if there is no direct, obvious evidence linking them and if one has to cross different domains to find the link, where a new link must provide some novel insight into the problem addressed.

In the area of literature based discovery (LBD), Swanson [7] and Smalheiser [8] developed an approach to assist the user in literature based discovery by detecting interesting cross domain terms with a goal to discover unknown relations between previously unrelated concepts. The online system ARROWSMITH [8] takes as input two sets of titles of scientific papers from disjoint domains A and C and lists terms that are common to A and C; the resulting bridging terms (B-terms) are further investigated by the user for their potential to generate new scientific hypotheses. They defined the so-called *closed discovery process*, where domains A and C are specified by the expert at the beginning of the discovery process.

Inspired by this early work, literature mining approaches were further developed and successfully applied to different problems, such as finding associations between genes and diseases [9], diseases and chemicals [10], and others. [11] describe several quality-oriented web-based tools for the analysis of biomedical literature, which include the analysis of terms (biomedical entities such as disease, drugs, genes, proteins and organs) and provide concepts associated with a given term. A recent approach by Kastrin et al. [12] is complementary to the other LBD approaches, in that it uses different similarity measures (such as common neighbors, Jaccard index, and preferential attachment) for link prediction of implicit relationships in the Semantic MEDLINE network.

Early work by Swanson has shown that databases such as PubMed can serve as a rich source of yet hidden relations between usually unrelated topics, potentially leading to novel insights and discoveries. By studying two separate literatures—the literature on migraine headache and the articles on magnesium—[13] discovered “Eleven neglected connections”, all of them supportive for the hypothesis that magnesium deficiency might cause migraine headache. Swanson’s literature mining results have been later confirmed by laboratory and clinical investigations. This well-known example has become a gold standard in the literature mining field and has been used as a benchmark in several studies, including those presented in [14–16] as well as in our own past work [17,18]. Research in literature mining, conducted by Petrič et al. [17,18], suggests that bridging terms are more frequent in documents that are in some sense different from the majority of documents in a given domain. For example, [18] have shown that such documents, considered outlier documents of their own domain, contain a substantially larger amount of bridging-linking terms than the normal, non-outlier documents.

The experimental data used to test the methodology proposed in this work are papers from the combined migraine-magnesium domain, studied extensively by Swanson and his followers, as well as the combined autism-calcineurin domain pair explored in [17,19].

Our contribution in this chapter follows two lines of our past research. First, it continues the work on cross-domain document exploration in [17,18], which explore outlier documents as means for literature based discovery. Note that the problem of finding outliers has been extensively studied also by another researcher [20] and has an immense use in many real-world applications. Second, and most importantly, the chapter continues our work on cross-domain bisociation exploration with CrossBee [5], which is most closely related to the work described here. CrossBee is an off-the-shelf solution for finding bisociative terms bridging two domains, which—as will be shown—can be used as the default user interface to the methodology presented in this chapter. Given that the CrossBee user interface is an actual ingredient of the technology developed in this work, its user interface is described in some more detail than other LBD systems mentioned in this section.

The CrossBee HCI functionality includes the following facilities: (a) *Performance evaluation* that can be used to measure the quality of results, e.g., through plotting ROC curves when the actual bridging terms are known in advance. (b) *Marking of high-ranked terms* by emphasizing them, thus making them easier to spot throughout the application. (c) *B-term emphasis* can be used to mark the terms predefined as B-terms by the user. (d) *Domain separation* colors all the documents from the same domain with the same color, making an obvious distinction between the documents from the two domains. (e) *User interface customization* enables the user to decrease or increase the intensity of the following features: high-ranked term emphasis, B-term emphasis and domain separation; this facility was introduced to enable the user to set the intensity of these features, given that in cooperation with the experts we discovered that some of them like the emphasizing features while others do not.

Note that the CrossBee web interface was designed for end-users who are not computer scientists or data miners and who prefer using the system by following a fixed sequence of predefined methodological steps. However, for a more sophisticated user of developer, the weakness of CrossBee is the lack of possibility to experiment with different settings as well as the lack of possibility to extend the methodology with new ideas and then compare or evaluate the developed approaches. As another weakness, the CrossBee web application does not offer a downloadable library and documentation distribution or extensive help. These weaknesses were among the incentives for our new developments, resulting in the TextFlows platform and its elaborate mechanisms for detecting and exploring bisociative links between the selected domains of interest.

4 Methodology Overview

In cross-domain knowledge discovery, estimating which of the terms have a high potential for interesting discoveries is a challenging research question. It is especially important for cross-context scientific discovery such as understanding complex medical phenomena or finding new drugs for yet not fully understood illnesses.

In our approach we focus on the closed discovery process, where two disjointed domains A and C are specified at the beginning of the discovery process and the main goal is to find bridging terms (see Fig. 1) which support validation of the novel hypothesized connection between the two domains. Given this motivation, the main functionality of the presented approach is bridging term (B-term) discovery, implemented through ensemble based term ranking, where an ensemble heuristic composed of six elementary heuristics was constructed for term evaluation.

To ensure the best user experience in the process of bridging term discovery we have combined the visual programming interface of the TextFlows workflow construction and execution platform with the bridging term exploration system CrossBee; CrossBee provides a user interface for term and document visualization that additionally supports the expert in finding relevant documents and exploration of the top-ranked bisociative terms.

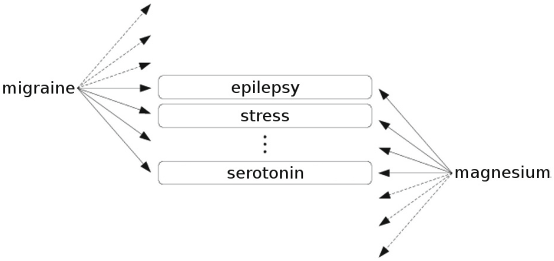
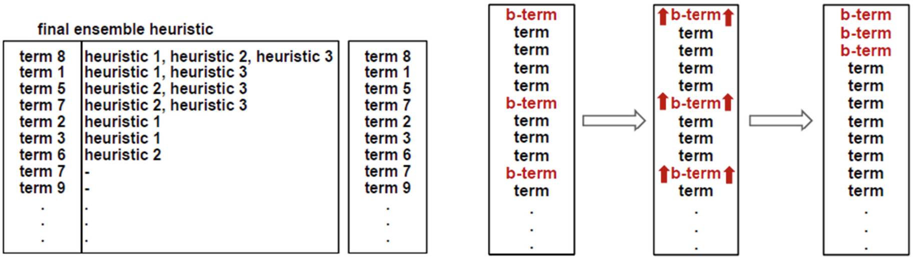


Fig. 1. Bridging term discovery when exploring migraine and magnesium document corpora, with B-terms as identified in [13] in the middle.

4.1 Methodology Illustration

The ensemble based term ranking methodology (using the final ensemble heuristic) is illustrated in Fig. 2.

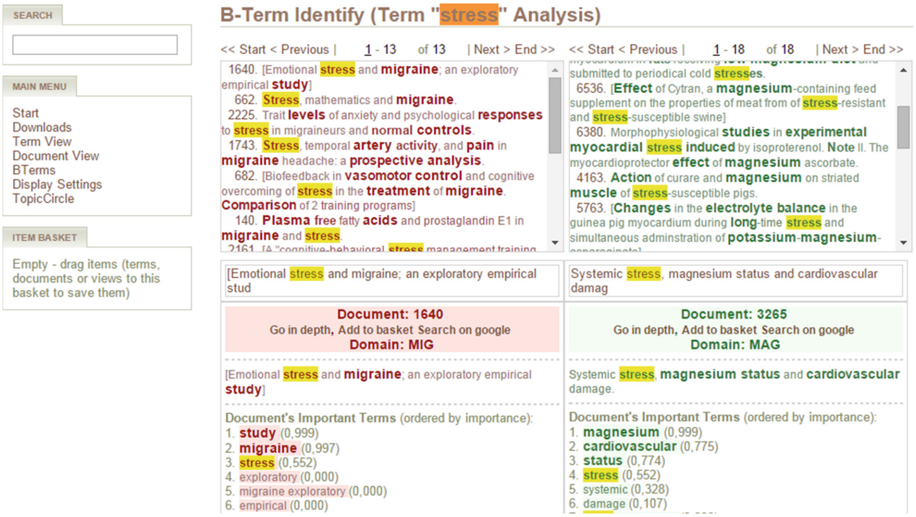

final ensemble heuristic

term 8	heuristic 1, heuristic 2, heuristic 3
term 1	heuristic 1, heuristic 3
term 5	heuristic 2, heuristic 3
term 7	heuristic 2, heuristic 3
term 2	heuristic 1
term 3	heuristic 1
term 6	heuristic 2
term 7	-
term 9	-
⋮	⋮

Fig. 2. Term ranking approach: first, ensemble heuristics vote for terms, next, terms are sorted according to their potential B-term (as shown on left). Consequently, bridging terms with the highest bridging term potential should receive the highest scores (as shown on the right side).

The user starts the bridging term discovery process in TextFlows by either constructing a new workflow for cross-domain discovery or by opening an existing workflow (such as the workflow shown in Fig. 4 of Sect. 4.2). In the first case, the user is required to input either a PubMed query or a file with documents from the two domains, where each line contains a document with exactly three tab-separated entries: (a) document identifier, (b) domain acronym, and (c) the document text. The user is able to tailor the preprocessing steps to his own needs by simply altering the workflow using the TextFlows visual programming user interface, which enables simple addition, connection and removal of components from the workflow canvas. In this way, the user can also modify the ensemble of elementary heuristics, outlier documents identified by external outlier detection software, the already known bisociative terms (B-terms), and others. When the user runs the workflows (by clicking the run button) the system starts with the process of text preprocessing, followed by the computation of elementary heuristics, the ensemble bisociation scores and term ranking.

After performing the calculation of bisociative potentials for every term in the vocabulary in TextFlows, the user is directed to the user-friendly tool CrossBee where one can efficiently investigate cross-domain links pointed out by the ensemble-based ranking methodology. CrossBee's document focused exploration empowers the user to filter and order the documents by various criteria, including detailed document view that provides a more detailed presentation of a single document including various term statistics. Methodology performance analysis supports the evaluation of the methodology by providing various data which can be used to measure the quality of the results, e.g., data for plotting the ROC curves. High-ranked term emphasis marks the terms according to their bisociation score calculated by the ensemble heuristic. When using this feature all high-ranked terms are emphasized throughout the whole application thus making them easier to spot (see different font sizes in Fig. 3). B-term emphasis marks the terms defined as B-terms by the user (yellow terms in Fig. 3). Domain separation is a simple but effective option which colors all the documents from the same domain with the same color, making an obvious distinction between the documents from the two domains (different colors in Fig. 3). User interface customization enables the user to decrease or increase the intensity of the following features: high-ranked term emphasis, B-term emphasis and domain separation.


SEARCH
MAIN MENU
Start
Downloads
Term View
Document View
BTerms
Display Settings
TopicCircle
ITEM BASKET
Empty - drag items (terms, documents or views to this basket to save them)
<< Start < Previous | 1 - 13 of 13 | Next > End >>
<< Start < Previous | 1 - 18 of 18 | Next > End >>
Go in depth, Add to basket Search on google
Go in depth, Add to basket Search on google

B-Term Identify (Term "stress" Analysis)

1640. [Emotional stress and migraine; an exploratory empirical study]
662. Stress, mathematics and migraine.
2225. Trait levels of anxiety and psychological responses to stress in migraineurs and normal controls.
1743. Stress, temporal artery activity, and pain in migraine headache: a prospective analysis.
682. [Biofeedback in vasomotor control and cognitive overcoming of stress in the treatment of migraine. Comparison of 2 training programs]
140. Plasma free fatty acids and prostaglandin E1 in migraine and stress.
2161. [A "cognitive-behavioral stress management training

submitted to periodical cold stresses.
6536. [Effect of Cytran, a magnesium-containing feed supplement on the properties of meat from of stress-resistant and stress-susceptible swine]
6380. Morphophysiological studies in experimental myocardial stress induced by isoproterenol. Note II. The myocardioprotector effect of magnesium ascorbate.
4163. Action of curare and magnesium on striated muscle of stress-susceptible pigs.
5763. [Changes in the electrolyte balance in the guinea pig myocardium during long-time stress and simultaneous adminstration of potassium-magnesium-

[Emotional stress and migraine; an exploratory empirical stud

Systemic stress, magnesium status and cardiovascular damag

Document: 1640
Domain: MIG

[Emotional stress and migraine; an exploratory empirical study]

Document's Important Terms (ordered by importance):
1. study (0,999)
2. migraine (0,997)
3. stress (0,552)
4. exploratory (0,000)
5. migraine exploratory (0,000)
6. empirical (0,000)

Document: 3265
Domain: MAG

Systemic stress, magnesium status and cardiovascular damage.

Document's Important Terms (ordered by importance):
1. magnesium (0,999)
2. cardiovascular (0,775)
3. status (0,774)
4. stress (0,552)
5. systemic (0,328)
6. damage (0,107)

Fig. 3. One of the useful features of the CrossBee interface is the side-by-side view of documents from the two domains under investigation. The analysis of the "stress" term from the migraine-magnesium domain is shown. The presented view enables efficient comparison of two documents, the left one from the migraine domain and the right one from the magnesium domain. (Color figure online)

4.2 Methodology Outline

This section describes how the complex methodology was developed as a workflow in the TextFlows platform, by presenting the entire pipeline of natural language processing (NLP) and literature based discovery (LBD) components. The top-level overview of the methodology, shown in Fig. 4, consists of the following steps: document acquisition, document preprocessing, heuristics specification, candidate B-term extraction, heuristic terms scores calculation, and visualization and exploration. An additional ingredient shown in Fig. 4—methodology evaluation—is not directly part of the methodology, however it is an important step of the developed approach.

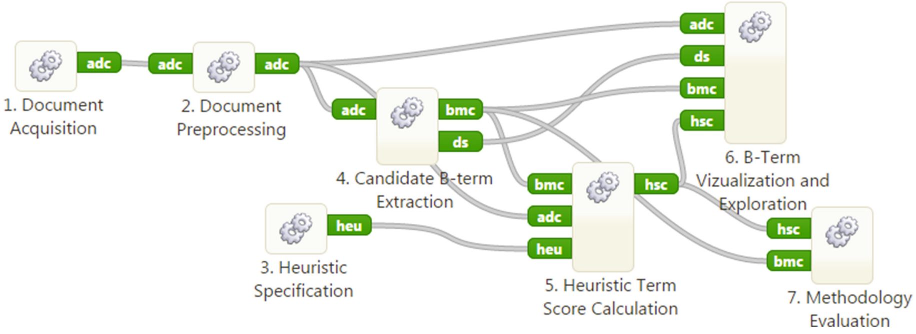

adc
1. Document Acquisition
adc
2. Document Preprocessing
adc
adc
4. Candidate B-term Extraction
bmc
ds
heu
3. Heuristic Specification
bmc
adc
heu
5. Heuristic Term Score Calculation
hsc
adc
ds
bmc
hsc
6. B-Term Vizualization and Exploration
hsc
bmc
7. Methodology Evaluation

Fig. 4. Methodological steps of the cross-domain literature mining process.

Top-level procedural explanation of the workflow shown in Fig. 4 is given below, while detailed explanations of individual steps of the workflow are described in Sect. 5.3.

1. Document acquisition is the first step of the methodology. Its goal is to acquire documents of the two domains, label them with domain labels and pack both domains together into the annotated document corpus format.
2. The document preprocessing step is responsible for applying standard text preprocessing to the document corpus. The main parts are tokenization, stopword tagging, and token stemming/lemmatization.
3. The heuristic specification step enables detailed specification of the heuristics to be used for B-term ranking. The user specifies one or more heuristics, which are to be applied to evaluate the B-term candidates. Note that each individual heuristic can be composed of other heuristics, therefore an arbitrary complex list of heuristics can be composed in this step.
4. The candidate B-term extraction step takes care of extracting the terms which are later scored by the specified heuristics. There are various parameters which control which kind of terms are extracted from the documents (e.g., the maximal number of tokens to be joined together as a term, minimal term

corpus frequency, and similar). The outputs are the *BoW Dataset* (i.e. the documents in the standard Bag-of-Words (BoW) vector format) and a *Bow Model Constructor*. The latter stores the list of all candidate B-terms along with the information about the input documents from annotated document corpus as well as the exact data how each document was parsed. This data is needed e.g., by the CrossBee web application when displaying the documents since it needs to be able to exactly locate specific words inside a document, in order to color or emphasize such words.

5. Heuristic term score calculation is the most important step of the methodology. It takes the list of extracted B-term candidates and the list of specified heuristics and calculates a heuristic score for each candidate term for each heuristic. The heuristics calculation is optimized so that common information used by different heuristics is calculated only once. The output is structurally still a list of heuristics, however now each of them contains a bisociation score for each candidate B-term.
6. Visualization and exploration is the final step of the methodology. It has three main functionalities. It can either take the heuristically scored terms, rank the terms, and output the terms in the form of a table, or it can take the heuristically scored terms along with the parsed document corpus and send them both to the CrossBee web application for advanced visualization and exploration. Besides improved bridging concept identification and ranking, CrossBee also provides various content presentations which further speed up the process of bisociation exploration. These presentations include e.g., side-by-side document inspection (see Fig. 3), emphasizing of interesting text fragments, and uncovering similar documents.
7. Methodology evaluation was introduces as an additional step, which can be used during the development of the methodology. Its purpose is to calculate and visualize various metrics that were used to assess the quality of the methodology. Requirement to use these facilities is to allow the actual (predefined) B-terms of the domain of investigation to act as gold standard B-terms available for evaluating the quality of B-term extraction and ranking.

Evaluation of the methodology was actually performed on two problems: the standard migraine-magnesium problem well-known in LBD, and a more recent autism-calcineurin literature mining problem. The evaluation of the methodology (its results are presented in detail in Sect. 6) provides evidence that the users empowered with the CrossBee functionality of term ranking and visualization are able to perform the crucial actions in cross-domain discovery more effectively than with conventional text mining tools.

Note that the described pipeline represents an actual executable workflow implemented in the online cloud-based workflow composition and execution environment TextFlows. The entire workflow, whose components are explained in detail in Sect. 5.3, is available for public reuse[2].

[2] http://textflows.org/workflow/486/.

5 Methodology Implementation

After presenting the main functionality of the TextFlows platform, this section presents the core mechanism of bisociative term detection, i.e., the designed heuristics and the workflows implementing the methodology in TextFlows. The section concludes by presenting the methodology empowered by using a controlled vocabulary in the search for bridging term.

5.1 The TextFlows Platform

We developed the TextFlows platform[3] as an open-source, web-based text mining platform that supports the construction and execution of text mining and natural language processing workflows. TextFlows was designed as a cloud-based web application that can be accessed and controlled from anywhere while the processing is performed in a cloud of computing nodes. TextFlows differs from comparable text mining platforms by its design that allows that during run-time the TextFlows platform resides on a server (or on a cluster of machines) while its graphical user interface that allows workflow construction is served as a web application accessible from any modern web browser. Furthermore, the platform's distinguishing feature is the ease of sharing and publicizing workflows constructed in TextFlows, together with an ever growing roster of reusable workflow components and entire workflows. As completed workflows, data, and results can also be made public by the author of the workflow, the platform was used to serve as an integration platform for development of various components supporting the literature based cross-domain discovery process, and for construction and evaluation of workflows, implementing the methodology proposed in Sect. 4.2.

Following a modular design, workflow components in TextFlows are organized into packages which allows for easier distributed development. The TextFlows packages implement several text mining algorithms from LATINO[4][22], NLTK [23] and scikit-learn [24] libraries. Moreover, TextFlows is easily extensible by adding new packages and workflow components. Workflow components of several types allow graphical user interaction during run-time, and visualization of results by implementing views in JavaScript, HTML or any other format that can be rendered in a web browser (e.g., Flash, Java Applet).

Below we explain the concept of workflows in more detail, describe the key text mining concepts of TextFlows and present the newly implemented package with workflow components supporting literature based discovery.

[3] Our platform TextFlows is a fork of data mining platform ClowdFlows [21], adapted to text mining and enriched with text analytics and natural language processing algorithms. As a fork of ClowdFlows, it benefits from its service-oriented architecture, which allows the user to utilize arbitrary web-services as workflow components. In addition to the new functionality, its novelty is a common text representation structure and the development of 'hubs' for algorithm execution.

[4] LATINO (Link Analysis and Text Mining Toolbox) is open-source—mostly under the LGPL license—and is available at https://github.com/LatinoLib/LATINO/.

Workflows. Executable graphical representations of complex procedures can be represented as workflows. The workflow model is the main component of the TextFlows platform and consists of an abstract representation of workflows and workflow components. The graphical user interface used for constructing workflows follows a visual programming paradigm which simplifies the representation of complex procedures into a spatial arrangement of building blocks. The most basic unit component in a TextFlows workflow is a processing component, which is represented as a widget in the graphical representation. Considering its inputs and parameters, every such component performs a task and stores the results on its outputs. Different processing components are linked via connections through which data is transferred from a widget's output to another widget's input. An alternative widget input for a widget are parameters, which the user enters into the widgets text fields. The graphical user interface implements an easy-to-use way of arranging widgets on a canvas to form a graphical representation of a complex procedure.

Workflows in TextFlows are processed and stored on remote servers from where they can be accessed from anywhere, requiring only an internet connection. By default each workflow can only be accessed by its author, although the user can also choose to make it publicly available. The TextFlows platform generates a specific URL for each workflow that has been saved as public. The users can then simply share their workflows by publishing the corresponding URL. Whenever a public workflow is accessed by another user, a copy of the workflow is created on the fly and added to his private workflow repository. The workflow is copied with all the data to ensure the experiments can be repeated. This enables the user to tailor the workflow to his needs without modifying the original workflow.

Key Text Mining Concepts in TextFlows. The key concepts in text mining are a corpus or a document collection, a single document, and document features [25]. Below we describe the model of corpora, documents and annotations on documents in TextFlows, which are the fundamental parts of our methodology. When designing TextFlows, the emphasis was on providing common representations which are passed among the majority of widgets:

Annotated corpus. A document collection is any grouping of text documents to be used for text analytics purposes. In TextFlows the Python[5] class that represents a corpus of documents is called *AnnotatedDocumentCorpus (ADC).* An ADC instance contains the collection of documents and its meta-data such as the authors, creation date, facts and notes about the dataset, etc. Features are stored in a simple key-value Python dictionary, where keys are strings and the values can store any Python object.

Annotated document. A single textual data unit within a collection—a document—is represented by the *AnnotatedDocument* class. An *AnnotatedDocument* instance may vary in size from a single sentence to a whole book. As with

[5] https://www.python.org/.

ADC, *AnnotatedDocument* instances also contain meta-data, such as author, date of publication, document length, assigned keywords, etc.

Annotation. Instances of the *Annotation* class are used to mark parts of the document, e.g., words, terms or sentences. Each *Annotation* instance has two pointers, one to the start and one to the end of the annotated stretch in the document text. These instances also have a type attribute used for grouping annotations of similar nature and contain key-value dictionaries of features, used by taggers to annotate parts of document with specific tags, e.g., annotations of type "token" that have a feature named "StopWord" with value "true", represent stop words in the document.

The Widget Repository. The following paragraphs present a subset of the TextFlows repository of widgets, which will be used in the workflows that implement the methodology proposed in Sect. 4.2.

Corpus and vocabulary acquisition. Document acquisition is usually the first step of every text mining methodology. TextFlows employs widgets which enable loading document corpora, labeling of documents with domain labels and converting them into the ADC structure. Document corpora can be loaded from files, where the dataset can be either a single text file, with each line representing a separate document, or a zip of files in which a document is represented as a file. Also supported is the upload of Word (.doc or .docx) and PDF files. Together with the text of the document the files may optionally contain document meta-data.

Corpus manipulation and visualization. TextFlows implements several widgets for manipulation of ADC data objects. These widgets allow the user to add new features, extract existing features from the document corpus, split document corpora (by either specifying conditions or by indices), merge different corpora, etc. A special widget in the platform is the *Document Corpus Viewer* widget, which visualizes the ADC data objects (note that TextFlows design emphasizes the importance of the ADC common document corpus representation which is passed among the majority of widgets). The interactive *Document Corpus Viewer* widget allows the user to check the results of individual widgets by visualizing the ADC data object from their outputs.

Text preprocessing. Preprocessing is a very important part of any form of knowledge extraction from text documents. Its main task is the transformation of unstructured data from text documents into a predefined well-structured data representation by extracting a high quality feature vector for every document in a given document corpus.

Our implementation employs the LATINO[6] [22], scikit-learn [24] and NLTK[7] [23] software libraries for its text preprocessing (and other processing) needs. These libraries *inter alia* contain the majority of elementary text preprocessing procedures as well as a large number of advanced procedures which support the conversion of a document corpus into a table of instances, thus converting every document into a table row representation of an instance.

The TextFlows preprocessing techniques are based on standard text mining concepts [25] and are implemented as separate categories. Every category possesses a unique hub widget, which has the task of applying a preprocessing technique from its category to the ADC data object. Every such widget is library independent, meaning that it can execute objects from either LATINO, NTLK or scikit-learn libraries. A standard collection of preprocessing techniques implemented in TextFlows includes: tokenization, stopword removal, Part-of-speech (PoS) tagging, as well as stemming and lemmatization.

In the data mining modeling phase (i.e. document classification or heuristic calculation), each document from the ADC structure needs to be represented as a set of document features it contains. In TextFlows the *Construct BoW Dataset and BoW Model Constructor* widget takes as an input an ADC data object and generates a sparse BoW model dataset (which can then be handed e.g. to a classifier). The widget takes as an input also several user defined parameters, which are taken into account when building the feature dataset. Besides the sparse BoW model dataset this widget also outputs a *BowModelConstructor* instance. This additional object contains settings which allow repetition of the feature construction steps on another document corpus. These settings include the input parameters, as well as the learned term weights and vocabulary.

Literature based discovery. This category of widgets supports the literature based discovery process. The package contains several widgets which specify different elementary heuristics. As will be described in Sect. 5.2, the basic heuristics are grouped into one of four categories: frequency-based, TF-IDF-based, similarity-based, outlier-based. Each category is represented by its own widget and the user is able to manually select its elementary heuristics through an interactive dialog. The literature based discovery package also contains several widgets which specify operations between elementary widgets, such as minimum, maximum, sum, norm, etc.

The library also contains two widgets which support the specification of ensemble heuristics, which will be described in Sect. 5.2: *Ensemble Heuristic Vote* and *Ensemble Average Position* widget. The first defines an ensemble voting heuristic (it calculates term votes according to Eq. 1 of Sect. 5.2), while the latter specifies an ensemble that calculates normalized sum of term position scores of the inputted heuristics (see Eq. 2 of Sect. 5.2).

[6] LATINO (Link Analysis and Text Mining Toolbox library) is open-source—mostly under the LGPL license—and is available at https://github.com/LatinoLib/LATINO/.

[7] Natural Language Toolkit.

The most important widget from this package is the *Calculate Term Heuristic Scores* widget which takes as an input several heuristics specifications and performs the actual calculations. The decision for such an approach—having one widget which calculates all the heuristics—is that several elementary heuristics require the same intermediate results. These results can be cached and calculated only once, which results in faster computation. To this end, the TextFlows platform uses Compressed Sparse Row (CSR) matrices[8] to be able to store the matrix of features in memory and also to speed up algebraic operations on vectors and matrices.[9]

Literature based discovery package also contains the *Explore in CrossBee* widget which exports the final ranking results and the annotated document corpus into web application CrossBee, which offers manual exploration of terms and documents. Also, the *Rank Terms* widget can be used to display the ranked terms in the form of a table along with their respective scores.

5.2 Implemented Heuristics for Bridging Term Discovery

This section presents different groups of elementary and ensemble heuristics, which are used for B-term ranking in the core step of the proposed methodology, i.e. in the heuristic term score calculation step.

The heuristics are defined as functions that numerically evaluate the term quality by assigning it bisociation score to a term (measuring the potential that a term is actually a B-term). For the definition of an appropriate set of heuristics, we define a set of special (mainly statistical) properties of terms, which aim at distinguishing B-terms from regular terms; thus, these heuristics can also be viewed as advanced term statistics. All heuristics operate on the data retrieved from the documents in text preprocessing. Ranking all the terms using the scores calculated by an ideal heuristic should result in ranking all the B-terms at the top of a ranked list. This is an ideal scenario, which is not realistic; however, ranking by heuristic scores should at least increase the proportion of B-terms at the top of the ranked term list. Formally, a heuristic is a function with two inputs, i.e. a set of domain labeled documents D and a term t appearing in these documents, and one output, i.e. a score that represents the term's bisociation potential.

We will use the following notation: to state that the term's bisociation score b is equal to the result of a heuristic named $heurX$, we can denote it as $b = heurX(D, t)$. However, since the set of input documents is static when dealing with a concrete dataset, we can—for the sake of simplicity—omit the set of input

[8] Compressed Sparse Row (CSR) matrices are implemented in the scipy.sparse package http://docs.scipy.org/doc/scipy/reference/sparse.html.

[9] The *Calculate Term Heuristic Scores* widget also takes as input the *BowModelContructor* object and the *AnnotatedDocumentCorpus*. The parse settings from the *BowModelConstructor* object are used to construct Compressed Sparse Row (CSR) matrices, which represents the BoW model. TextFlows uses mathematical libraries numpy and scipy to efficiently perform the heuristics calculations.

documents from a heuristic notation and use only $b = heurX(t)$. Whenever we need to explicitly specify the set of documents to which the function is applied (never needed for a heuristic, but sometimes needed for auxiliary functions used in the formula for the heuristic), we write it as $funcX_D(t)$. For specifying the function's input document set, we have two options: either use D_u that stands for the (union) set of all the documents from all the domains, or use $D_n : n \in \{1..N\}$, which stands for the set of documents from the given domain n. In general, the following statement holds: $D_u = \cup_{n=1}^{N} D_n$, where N is the number of domains. In the most common scenario, when there are exactly two distinct domains, we also use the notation D_A for D_1 and D_C for D_2, similarly to Swanson's notation of symbols A and C as representatives of the initial and the target domain in the closed discovery setting, mentioned in Sect. 3.

Base Heuristics. We divide the heuristics into different sets for easier explanation; however, most of the described heuristics work fundamentally in a similar way—they all manipulate solely the data present in term and document vectors and derive the terms bisociation score. The exceptions to this are the outlier-based heuristics, which first evaluate outlier documents and only later use the information from the term vectors for B-term evaluation.

We can thus define four sets of base heuristics: frequency based, TF-IDF based, outlier based and similarity based heuristics. In following sections we describe each set in more detail.[10]

Frequency-based heuristics. We first define two auxiliary functions:

- $countTerm_D(t)$: counts the number of occurrences of term t in a document set D (called term frequency in TF-IDF related contexts),
- $countDoc_D(t)$: counts the number of documents in which term t appears in document set D (called document frequency in TF-IDF related contexts).

We define the following base heuristics:

- $freqTerm(t) = countTerm_{D_u}(t)$: term frequency in the two domains,
- $freqDoc(t) = countDoc_{D_u}(t)$: document frequency in the two domains,
- $freqRatio(t) = \frac{countTerm_{D_u}(t)}{countDoc_{D_u}(t))}$: term to document frequency ratio,
- $freqDomnRatioMin(t) = \min(\frac{countTerm_{D_1}(t)}{countTerm_{D_2}(t)}, \frac{countTerm_{D_2}(t)}{countTerm_{D_1}(t)})$: minimum of term frequencies ratio of the two domains,
- $freqDomnProd(t) = countTerm_{D_1}(t) \cdot countTerm_{D_2}(t)$: product of term frequencies of the two domains,
- $freqDomnProdRel(t) = \frac{countTerm_{D_1}(t) \cdot fcountTerm_{D_2}(t)}{countTerm_{D_u}(t)}$: product of term frequencies of the two domains relative to the term frequency in all domains.

[10] Due to a large number of heuristics and auxiliary functions, we use the so called camel casing multi-word naming scheme for easier distinction; names are formed by word concatenation and capitalization of all non first words (e.g., *freqProdRel* and *tfidfProduct*).

TF-IDF-based heuristics. TF-IDF is a standard measure of term's importance in a document, which is used heavily in text mining research [26]. In the following heuristics definitions, we use the following auxiliary functions:

- $tfidf_d(t)$ stands for TF-IDF weight of term t in document d,
- $tfidf_D(t)$ represents TF-IDF weight of term t in the centroid vector of all documents d, $d \in D$, where the centroid vector is defined as an average of all document vectors and thus presents an average document of document collection D.

Heuristics based on TF-IDF are listed below:

- $tfidfSum(t) = \sum_{d \in D_u} tfidf_d(t)$: sum of all TF-IDF weights of term t in the two domains; this heuristic is analogous to freqTerm(t),
- $tfidfAvg(t) = \frac{\sum_{d \in D_u} tfidf_d(t)}{freqDoc_{D_u}(t)}$: average TF-IDF weights of term t across all domains,
- $tfidfDomnProd(t) = tfidf_{D_1}(t) \cdot tfidf_{D_2}(t)$: product of TF-IDF weights of term t in the two domains,
- $tfidfDomnSum(t) = tfidf_{D_1}(t) + tfidf_{D_2}(t)$: sum of term TF-IDF weights of term t in the two domains.

Similarity-based heuristics. Another approach to construct a relevant heuristic measure is to use the cosine similarity measure that is frequently used in text mining to compute the similarity of documents. We start by creating a representational BoW model as a document space and by converting terms into BoW document vectors. Next, we get the centroid vectors for both domains in the document space representation. Finally, we apply TF-IDF weighting on top of all the newly constructed vectors and centroids. We define the following auxiliary function:

- $simCos_D(t)$: calculates the cosine similarity of the document vector of term t and the document vector of a centroid of documents $d \in D$.

The base heuristics are the following:

- $simAvgTerm(t) = simCos_{D_u}(t)$: similarity of term t to an average term, i.e. the distance from the center of the cluster of all terms,
- $simDomnProd(t) = simCos_{D_1}(t) \cdot simCos_{D_2}(t)$: product of similarity of term t to the centroids of the two domains,
- $simDomnRatioMin(t) = \min(\frac{simCos_{D_1}(t)}{simCos_{D_2}(t)}, \frac{simCos_{D_2}(t)}{simCos_{D_1}(t)})$: minimum of term's frequency ratios of the two domains.

Outlier-based heuristics. Outlier detection is an established area of data mining [20]. Conceptually, an outlier is an unexpected event, entity or—in our case—an irregular document. We are especially interested in outlier documents since they frequently embody new information that is often hard to explain in the context of existing knowledge. Moreover, in data mining, an outlier is occasionally a primary object of study as it can potentially lead to the discovery of new knowledge. These assumptions are well aligned with the bisociation potential that we wish to optimize, thus, we have constructed several heuristics that harvest the information possibly residing in outlier documents.

We concentrate on a specific type of outliers, i.e. domain outliers, which are the documents that tend to be more similar to the documents of the opposite domain than to those of their own domain. The techniques that we use to detect outlier documents [18] is based on using classification algorithms to detect outlier documents. First we train a classification model for each domain and afterwards classify all the documents using the trained classifier. The documents that are misclassified—according to their domain of origin—are declared as outlier documents, since according to the classification model they do not belong to their domain of origin.

We defined three different outlier sets of documents based on three classification algorithms utilized. These outlier sets are:

- D_{CS}: documents misclassified by the Centroid Similarity (CS) classifier,
- D_{RF}: documents misclassified by the Random Forest (RF) classifier,
- D_{SVM}: documents misclassified by the Support Vector Machine (SVM) classifier.

Centroid similarity is a basic classifier model implemented in our system. It classifies each document to the domain whose centroid's TF-IDF vector is the most similar to the document's TF-IDF vector. The description of the other two classification models is beyond the scope of this chapter, as we used external procedures to retrieve these outlier document sets; a detailed description is provided by [18].

For each outlier set we defined two heuristics: the first counts the frequency of a term in an outlier set and the second computes the relative frequency of a term in an outlier set compared to the relative frequency of a term in the whole dataset. The resulting heuristics are listed below:

- $outFreqCS(t) = countTerm_{D_{CS}}(t)$: frequency of term t in the CS outlier set,
- $outFreqRF(t) = countTerm_{D_{RF}}(t)$: frequency of term t in the RF outlier set,
- $outFreqSVM(t) = countTerm_{D_{SVM}}(t)$: frequency of term t in the SVM outlier set,
- $outFreqSum(t) = countTerm_{D_{CS}}(t) + countTerm_{D_{RF}}(t) + countTerm_{D_{SVM}}(t)$: sum of frequencies of term t in all three outlier sets,
- $outFreqRelCS(t) = \frac{countTerm_{D_{CS}}(t)}{countTerm_{D_u}(t)}$: relative frequency of term t in the CS outlier set,
- $outFreqRelRF(t) = \frac{countTerm_{D_{RF}}(t)}{countTerm_{D_u}(t)}$: relative frequency of term t in the RF outlier set,
- $outFreqRelSVM(t) = \frac{countTerm_{D_{SVM}}(t)}{countTerm_{D_u}(t)}$: relative frequency of term t in the SVM outlier set,
- $outFreqRelSum(t) = \frac{countTerm_{D_{CS}}(t)+countTerm_{D_{RF}}(t)+countTerm_{D_{SVM}}(t)}{countTerm_{D_u}(t)}$: sum of relative term frequencies of term t in all three outlier sets.

Ensemble Heuristics Construction. Ensemble learning is a known approach used in machine learning for combining predictions of multiple models into a final prediction. It is well evidenced [27] that the resulting ensemble model is more accurate than any of the individual models used to build it as long as the models are similarly accurate, are better than random, and their errors are uncorrelated. There is a wide variety of known and well tested ensemble techniques, such as bagging, boosting, majority voting, random forest, naive Bayes, etc. [28]. However, these approaches are usually used for the problem of classification while the core problem presented in this work is ranking. Nevertheless, with the rise of the areas like information retrieval and search engines' web page rankings, ensemble ranking is also gaining attention in the ranking community [29].

One possible—and probably the most typical—approach to designing an ensemble heuristic from a set of base heuristics consists of two steps. In the first step, the task is to select member heuristics for the ensemble heuristic using standard data mining approaches like feature selection. In the second step, equation discovery is used to obtain an optimal combination of member heuristics. The advantage of such approach is that the ensemble creation does not require manual intervention. Therefore, we performed several experiments with this approach; however, the results of an ensemble were even more overfitted to the training domain. Consequently, we decided to manually—based on experience and experimentation—select appropriate base heuristics and construct an ensemble heuristic. As the presentation of numerous experiments, which support our design decisions, is beyond the scope of this chapter, we describe only the final solution, along with some reasoning about choosing the heuristics.

The ensemble heuristic for bridging term discovery, which we constructed based on the experiments, is constructed from two parts: the ensemble voting score and the ensemble position score, which are summed together to give the final ensemble score for every term in the corpus vocabulary. Each term score represents the term's potential for joining the two disjointed domains.

The ensemble voting score (s_t^{vote}) of a given term t is an integer, which denotes how many base heuristics voted for the term. Each selected base heuristic h_i gives one vote ($s_{t_j,h_i}^{vote} = 1$) to each term, which is in the first third in its ranked list of terms and zero votes to all the other terms ($s_{t_j,h_i}^{vote} = 0$). The voting threshold one third ($\frac{1}{3}$) was set empirically grounded on the evaluation of the ensemble heuristic on the migraine-magnesium domain and is based on the number of terms that appear in both domains (not one third of all the terms). Formally, the ensemble voting score of term t_j that is at position p_j in the ranked list of n terms is computed as a sum of individual heuristics' voting scores:

$$s_{t_j}^{vote} = \sum_{i=1}^{k} s_{t_j,h_i}^{vote} = \sum_{i=1}^{k} \begin{cases} 1, & p_j < n/3 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Therefore, each term can get a score $s_{t_j}^{vote} \in \{0, 1, 2, ..., k\}$, where k is the number of base heuristics used in the ensemble. The ensemble position score (s_t^{pos}) is calculated as an average of position scores of individual base heuristics. For each heuristic h_i, the term's position score $s^{pos}_{t_j,h_i}$ is calculated as $\frac{n-p_j}{n}$, which results in position scores being in the interval $[0, 1)$. For an ensemble of k heuristics, the ensemble position score is computed as an average of individual heuristics' position scores:

$$s^{pos}_{t_j} = \frac{1}{k}\sum_{i=1}^{k} s^{pos}_{t_j,h_i} = \frac{1}{k}\sum_{i=1}^{k} \frac{n-p_j}{n} \tag{2}$$

The final ensemble score is computed as:

$$s_t = s^{vote}_t + s^{pos}_t \tag{3}$$

Using the proposed construction we make sure that the integer part of the ensemble score always presents the ensemble vote score, while the ensemble score's fractional part always presents the ensemble position score. An ensemble position score is strictly lower than 1, therefore a term with a lower ensemble voting score can never have a higher final ensemble score than a term with a higher ensemble voting score. Consequently, every final ensemble score falls into interval $[0, k + 1)$, where k is the number of base heuristics used in the ensemble.

The described method for ensemble score calculation is illustrated in Tables 1–5. In Table 1 the base heuristics scores are shown for each term. Table 2 presents terms ranked according to the base heuristics scores. From this table, the voting and position scores are calculated for every term based on its position, as shown in Table 3. For example, all terms at position 2, i.e. t1, t6, and t6, get voting score 1 and position score 4/6. Table 4 shows the exact equation how these base heuristics voting and position scores are combined for each term. Table 5 displays the list of terms ranked by the calculated ensemble scores.

Table 1. Base heuristic scores

Term	h_1	h_2	h_3
t1	0.93	0.46	0.33
t2	0.26	0.15	0.10
t3	0.51	0.22	0.79
t4	0.45	0.84	0.73
t5	0.41	0.15	0.11
t6	0.99	0.64	0.74

Table 2. Terms ranked by base heuristics

Pos.	h_1	h_2	h_3
1	t_6	t_4	t_3
2	t_1	t_6	t_6
3	t_3	t_1	t_4
4	t_4	t_3	t_1
5	t_5	t_2	t_5
6	t_2	t_5	t_2

Table 3. Voting and position scores based on positions in the ranked lists

Pos.	$s^{vote}_{t_j,h_i}$	$s^{pos}_{t_j,h_i}$
1	1	(6−1)/6 = 5/6
2	1	(6−2)/6 = 4/6
3	0	(6−3)/6 = 3/6
4	0	(6−4)/6 = 2/6
5	0	(6−5)/6 = 1/6
6	0	(6−6)/6=0/6

Note that at the first sight, our method of constructing the ensemble score looks rather intricate. An obvious way to construct an ensemble score of a term could be simply to sum together individual base heuristics scores; however, the calculation of the ensemble score by our method is well justified by extensive experimental results on the migraine-magnesium dataset described in Sect. 6. The final set of elementary heuristics included in the ensemble is the following:

Table 4. Calculation of ensemble heuristic score

	(	$s^{vote}_{t_j,h_1}$	+	$s^{vote}_{t_j,h_2}$	+	$s^{vote}_{t_j,h_3}$	) + (	$s^{pos}_{t_j,h_1}$	+	$s^{pos}_{t_j,h_2}$	+	$s^{pos}_{t_j,h_3}$	)/k =	$s^{vote}_{t_j}$	+	$s^{pos}_{t_j}$	=	s_{t_j}
s_{t_1} =	(	1	+	0	+	0	) + (	4/6	+	3/6	+	2/6	)/3 =	1	+	9/18	=	1.50
s_{t_2} =	(	0	+	0	+	0	) + (	0/6	+	1/6	+	0/6	)/3 =	0	+	1/18	=	0.06
s_{t_3} =	(	0	+	0	+	1	) + (	3/6	+	2/6	+	5/6	)/3 =	1	+	10/18	=	1.56
s_{t_4} =	(	0	+	1	+	0	) + (	2/6	+	5/6	+	3/6	)/3 =	1	+	10/18	=	1.56
s_{t_5} =	(	0	+	0	+	0	) + (	1/6	+	0/6	+	1/6	)/3 =	0	+	2/18	=	0.11
s_{t_6} =	(	1	+	1	+	1	) + (	5/6	+	4/6	+	4/6	)/3 =	3	+	13/18	=	3.72

Table 5. Ranked list of terms produced by the ensemble

t6 (3.72), [t2, t3] (1.56), t1 (1.50), t5 (0.11), t2 (0.06)

- outFreqRelRF
- outFreqRelSVM
- outFreqRelCS
- outFreqSum
- tfidfDomnSum
- freqRatio

Detailed justification is presented in [30].

5.3 Workflows Implementing Individual Steps of the Methodology

The workflow for cross-domain literature mining, presented in Sect. 4.2, is publicly available for sharing and reuse within the TextFlows platform. The workflow integrates the computation of heuristics, described in Sect. 5.2, and is connected to the term exploration interface of the online system CrossBee, which supports the user in advanced document exploration by facilitating document analysis and visualization.

Document Acquisition Workflow (Step 1). The first step of the workflow from Fig. 4 is composed of several components described below. The components are responsible for the following tasks:

1.1. load literature A into annotated document corpus data structure

1.1.1. load raw text data from a file (this component could be replaced by loading documents from the web or by acquiring them using web services), where each line contains a document with exactly three tab-separated entries: (a) document identifier, (b) domain acronym, and (c) the document text,

1.1.2. build the annotated document corpus from the raw data, i.e. parse the loaded raw text data into a collection of documents and assign a domain label (e.g., literature A, docsA, migraine) to the documents to enable their identification after merging with literature B,

1.2. load literature B into the annotated document corpus data structure (individual components are aligned with the components 1.1),

1.3. merge the two literatures into a single annotated document corpus structure,
1.4. optional check of document acquisition by visual inspection of the created corpus.

The document acquisition workflow is shown in Fig. 5. The output is the annotated document corpus consisting of the acquired documents labeled with domain labels.

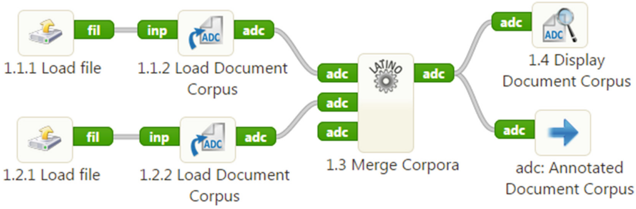

fil
inp
adc
1.1.1 Load file
1.1.2 Load Document Corpus
fil
inp
adc
1.2.1 Load file
1.2.2 Load Document Corpus
adc
adc
adc
adc
1.3 Merge Corpora
adc
1.4 Display Document Corpus
adc
adc: Annotated Document Corpus

Fig. 5. Document acquisition workflow.

Text Preprocessing Workflow (Step 2). The document acquisition step is followed by the text preprocessing step, which is itself a workflow implemented as shown in Fig. 6. The main components here are tokenization, stopwords labeling and token stemming or lemmatization. The output of this step is structurally equal to the input; however every document in the annotated document corpus now contains additional information about tokens, stopwords and lemmas.

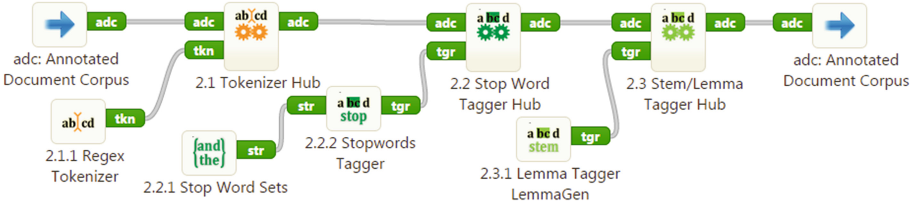


Fig. 6. Document preprocessing workflow.

The individual components perform the following tasks:

2.1 split documents to tokens (the basic units for further text processing),
2.1.1. create tokenizer object (simple tokenizer based on regular expressions),
2.2. tag stopword tokens by using a stopword tagger (component 2.2.2),
2.2.1. load standard English stopwords,
2.2.2. define the stopword tagger using the standard English stopwords only (the detected stopwords are used in candidate B-term extraction step),

2.3. lemmatize tokens by applying the LemmaGen lemmatizer[11] [31],
2.3.1. create an instance of LemmaGen lemmatizer.

Heuristics Specification Workflow (Step 3). While the heuristics specification step is the core part of our methodology, this step only specifies which heuristics are selected and how these heuristics should be combined into the ensemble heuristic. The actual calculation is performed later in the heuristic term score calculation step.

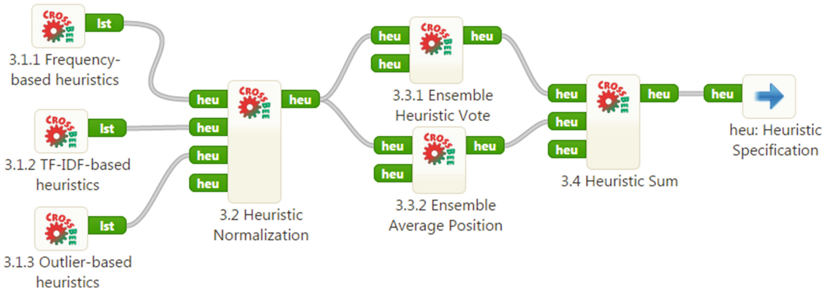

lst
3.1.1 Frequency-based heuristics
lst
3.1.2 TF-IDF-based heuristics
lst
3.1.3 Outlier-based heuristics
heu
heu
heu
heu
heu
3.2 Heuristic Normalization
heu
heu
heu
3.3.1 Ensemble Heuristic Vote
heu
heu
heu
3.3.2 Ensemble Average Position
heu
heu
heu
heu
3.4 Heuristic Sum
heu
heu: Heuristic Specification

Fig. 7. Heuristic specification.

Heuristic specification displayed in Fig. 7 is the outcome of our research about the base term heuristics and their combination into the ensemble heuristic presented in Sect. 5.2. Which heuristics to use and how to combine them is based on the experiments on the real data that we performed as a part of the research presented in this chapter—these experiments are presented in more detail in [30]. The findings resulted in the setting shown in Fig. 7, which is a good choice when applied on new data. Nevertheless, the setting and the choice of the base heuristics is fully customizable and can be freely configured to better suit the needs of new applications.

The output of this procedure is a specification of a complex ensemble heuristic, which computes the term bisociation scores. The components in the heuristic specification perform the following tasks:

3.1. define base heuristics (see Sect. 5.2 for details about the base heuristics selection),
3.1.1. define TF-IDF based heuristic *tfidfDomnSum*,
3.1.2. define term frequency based heuristic *freqRatio*,
3.1.3. define outlier based heuristics *outFreqRelRF, outFreqRelSVM, outFreqRelCS, outFreqRelSum*
3.2. for every inputted heuristic defines a new heuristic that normalizes the scores to the range [0,1) and outputs a list of new heuristic specifications,

[11] LemmaGen is an open source lemmatizer with 15 prebuilted european lexicons. Its source code and documentation is publicly available at http://lemmatise.ijs.si/.

3.3. combine the six heuristics into a single ensemble heuristic

3.3.1. define an ensemble voting heuristic that includes votes of the six heuristics (ensemble voting score, see Eq. 1),

3.3.2. define a calculated heuristic that calculates normalized sum of position scores of the six heuristics (ensemble position score, see Eq. 2),

3.4. define the final ensemble heuristic by summing the ensemble voting heuristics, which results in the number of terms heuristics' votes in the range from 0 to 6 (integer value), and the calculated normalized sum of heuristics scores in the range from 0 to less than 1 (final ensemble score, see Eq. 3).

Candidate B-term Extraction Workflow (Step 4). Another core step of the workflow is candidate B-term extraction, shown in Fig. 8. Although it contains only one component, it has a very important and complex goal of transforming the inputted annotated document corpus into the BoW model in order to represent documents in the form of feature vectors of term occurrences in the documents (for the purpose of visualization of documents and the need of highlighting and emphasizing of specific terms). Another task of this step is to capture the exact parsing procedure, which is needed in order to perform various computations which are performed in the advanced heuristic term scores calculation step. The outputted *BowModelContructor* object also contains the vocabulary of all terms.

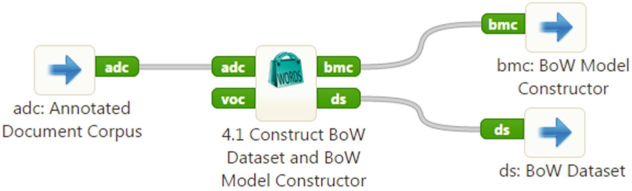

adc
adc: Annotated Document Corpus
adc
voc
bmc
ds
4.1 Construct BoW Dataset and BoW Model Constructor
bmc
bmc: BoW Model Constructor
ds
ds: BoW Dataset

Fig. 8. Candidate B-term extraction.

Heuristic Term Score Calculation Workflow (Step 5). Figure 9 shows a structurally simple methodological step of heuristic term score calculation that contains only one component. The inputs to the procedure are the annotated document corpus, the *BoWModelContructor* and the heuristics specification. Based on the information present in the *BoWModelContructor*, the algorithm calculates various frequency and TF-IDF document features vectors, which are used to calculate the specified heuristics scores for all the terms. The calculation results in the same heuristic structure as defined in the heuristic specification step, however the ensemble heuristic at the top level, as well as all elementary heuristics, now contain their calculated scores of the terms. The scores of the top-level heuristic are intended to represent terms' bisociation scores and are typically used as a basis for the final term ranking.

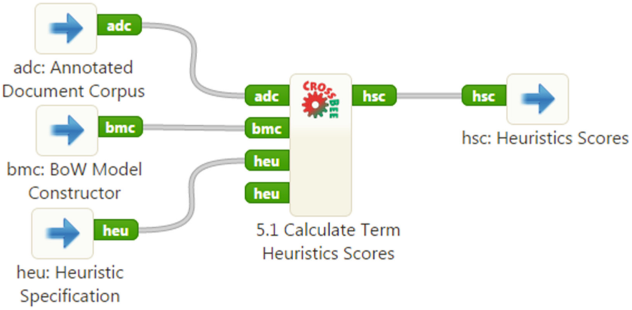

adc
adc: Annotated Document Corpus
bmc
bmc: BoW Model Constructor
heu
heu: Heuristic Specification
adc
bmc
heu
heu
hsc
CROSSBEE
5.1 Calculate Term Heuristics Scores
hsc
hsc: Heuristics Scores

Fig. 9. Heuristic term score calculation.

B-Term Visualization and Exploration Workflow (Step 6). This step of the methodology implements a workflow shown in Fig. 10. It enables visualization and exploration of the ranked list of B-terms. There are four inputs to this step. The first and the most important are the ensemble heuristic scores of the extracted candidate B-terms. Inputs *Annotated Document Corpus* and *BoW Dataset* are used by the online application for cross-context bisociation exploration CrossBee, which needs the exact information about term extraction from documents to be able to align the terms back with the original documents in order to visualize them; while the *BoW Model Constructor* provides the constructed vocabulary. The goals of the created components are the following:

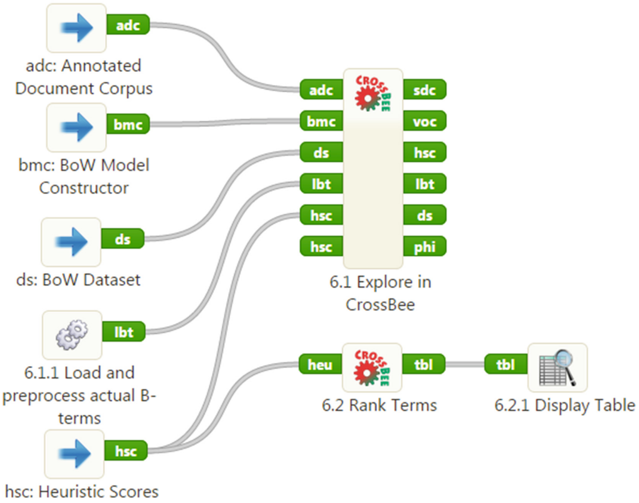


Fig. 10. B-term visualization and exploration.

6.1. explore the final results in a web application CrossBee, which was designed specifically for the purpose of bisociativity exploration (expressed either through terms or through documents),

6.1.1. optional expert specified B-terms may be provided to CrossBee in order to emphasize them in the text and to deliver a feedback about the bisociative quality of the provided ranking. If available, these terms are loaded and preprocessed using the same preprocessing techniques as described in the document preprocessing step,

6.2. rank the terms

6.2.1. display the ranked terms in the form of a table along with their respective scores.

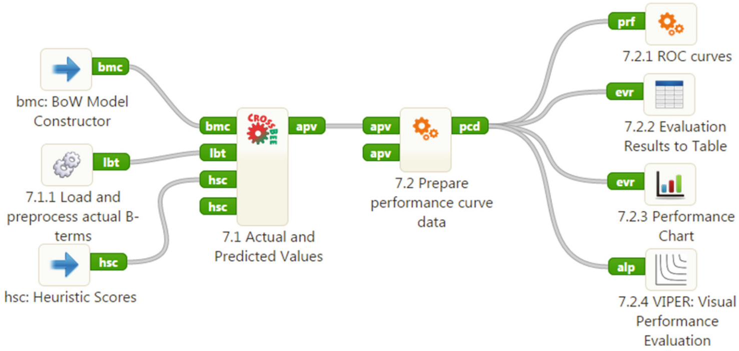

bmc
bmc: BoW Model Constructor
lbt
7.1.1 Load and preprocess actual B-terms
hsc
hsc: Heuristic Scores
bmc
lbt
hsc
hsc
apv
7.1 Actual and Predicted Values
apv
apv
pcd
7.2 Prepare performance curve data
prf
7.2.1 ROC curves
evr
7.2.2 Evaluation Results to Table
evr
7.2.3 Performance Chart
alp
7.2.4 VIPER: Visual Performance Evaluation

Fig. 11. Methodology evaluation.

Methodology Evaluation Workflow (Step 7). The last step of the proposed methodology is the methodology evaluation step, implemented as a workflow shown in Fig. 11. There are three inputs to the process: the heuristic scores of one or more evaluated heuristics (which presents the result of all the preceding methodological steps), the *BowModelContructor* (which contains the corpus vocabulary) and additional information about the actual B-terms (required in order to assess any kind of quality measures). Note that, in order not to overflow the overall methodology workflow of Fig. 4 with additional information, the list of actual bridging terms was not shown as an additional step of the methodology. Instead, it is implemented as a separate subprocess in the methodology evaluation workflow, which is responsible for loading and preprocessing the actual B-terms.

The components of the methodology evaluation workflow perform the following tasks:

7.1. prepare pairs of actual and predicted values, which are used to calculate different information retrieval measures in step 7.2, 7.1.1. if available, load the actual (expert identified) B-terms, which present the gold standard terms used to evaluate the quality of the methodology and preprocess them using same techniques as in document preprocessing step,

7.2. calculate different measures, such as precision, recall, and the F_1-measure, ROC curves and the AUC (Area Under Curve) values,

7.2.1. display ROC curves graphically,

7.2.2. compare information retrieval measures in the form of a table,

7.2.3. compare information retrieval measures in the form of a bar chart,

7.2.4. display and compare the F_1-scores in the advanced VIPER performance evaluation chart [32] component.

The methodology evaluation functionality presented in this section is not part of the actual workflow for cross-domain knowledge discovery; however, it is indispensable when developing a new approach. Description of this step concludes the section presenting the key parts of the methodology.

5.4 Methodology Empowerment with Controlled Vocabulary

This section describes a new ingredient of the methodology: the use of a controlled vocabulary for improving B-term detection and ranking. The motivation for using predefined controlled vocabularies is to reduce the heuristic search space which, consequently, reduces the running times of B-term discovery algorithms. Controlled vocabularies ensure consistency and resolve ambiguity inherent in normal human languages where the same concept can be given different names. In this way, they improve the quality and organization of retrieved knowledge, given that they consist of predefined, authorized terms that have been pre-selected by the designers of the vocabulary that are experts in the subject area. Controlled vocabularies solve problems of homographs and synonyms by a bijection between concepts and authorized terms.

MeSH (Medical Subject Headings) is a controlled vocabulary used for indexing articles for PubMed, designed by The National Library of Medicine (NLM). Figure 12 shows a top-level example of the MeSH structure and hierarchy. The 2015 version of MeSH contains a total of 27,455 subject headings, also known as descriptors. Each descriptor is assigned a unique tree number (shown in square brackets in Fig. 12) that facilitates search and filtering. Most of the descriptors are accompanied by a short description or definition, links to related descriptors, and a list of synonyms or very similar terms (known as entry terms). Because of these synonym lists MeSH can also be viewed as a thesaurus.

We have implemented a vocabulary construction tool called *MeSH filter* as an interactive widget in the TextFlows platform. This implementation uses synonym lists from the MeSH 2015 database, available online[12]. The interface to the developed interaction widget is designed to enable the selection of descriptors of interest from the hierarchy of descriptors. Its final output is a text file

[12] http://www.nlm.nih.gov/mesh/filelist.html.

```
Nervous System Diseases [C10]
  Central Nervous System Diseases [C10.228]
    Brain Diseases [C10.228.140]
      Headache Disorders [C10.228.140.546]
        Headache Disorders, Primary [C10.228.140.546.399]
          Migraine Disorders [C10.228.140.546.399.750]
            Alice in Wonderland Syndrome [C10.228.140.546.399.750.124]
            Migraine with Aura [C10.228.140.546.399.750.250]
            Migraine without Aura [C10.228.140.546.399.750.450]
            Ophthalmoplegic Migraine [C10.228.140.546.399.750.725]
          Tension-Type Headache [C10.228.140.546.399.875]
          Trigeminal Autonomic Cephalalgias [C10.228.140.546.399.937]
```

Fig. 12. Example of MeSH structure and hierarchy.

containing all the terms that belong to the user selected descriptors from the MeSH hierarchy.

This section describes how we have upgraded the proposed methodology with the ability to use a predefined controlled vocabulary for reducing the B-term search space. This not only increases efficiency of the heuristic calculation algorithms, but also tends to improve the relevance of top ranked B-terms due to reduced ambiguities in human languages. The upgraded methodology is shown in Fig. 13. Compared to the initial methodology shown in Fig. 4, the new workflow[13] includes two new steps: vocabulary acquisition and vocabulary preprocessing.

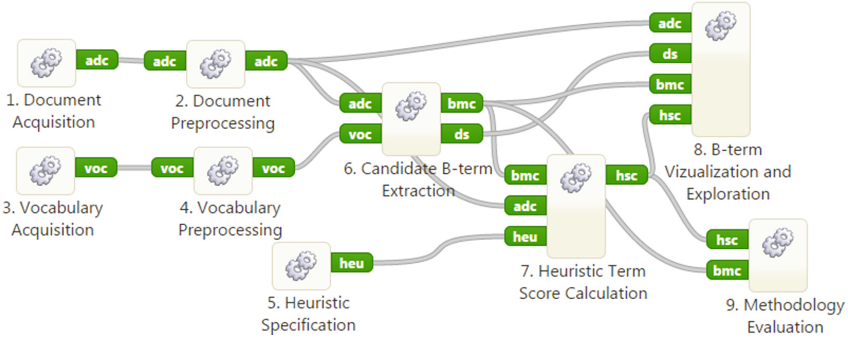

adc
1. Document Acquisition
adc
adc
2. Document Preprocessing
voc
3. Vocabulary Acquisition
voc
voc
4. Vocabulary Preprocessing
heu
5. Heuristic Specification
adc
voc
bmc
ds
6. Candidate B-term Extraction
bmc
adc
heu
hsc
7. Heuristic Term Score Calculation
adc
ds
bmc
hsc
8. B-term Vizualization and Exploration
hsc
bmc
9. Methodology Evaluation

Fig. 13. Methodological steps of the cross-domain literature mining process.

In order to ensure the proper matching between terms from the vocabulary and document corpus, the vocabulary file must be preprocessed using the preprocessing techniques, described in Sect. 5.3, which were also used for preprocessing the document corpus in Step 2. After vocabulary preprocessing in

[13] This workflow is publicly available at http://textflows.org/workflow/497/.

Step 4, the produced vocabulary file is used in Step 5 to filter out terms from the document corpus that do not appear in the vocabulary. A procedural explanation of the new steps of the upgraded workflow of Fig. 13 is presented.

Vocabulary Acquisition (Step 3)

- *One term per line*: Every single line in the text file represents one separate term. Only terms which appear in this file are later used in the heuristic calculation steps of the methodology.
- *Synonym format*: Additionally, term synonyms are listed after the term, separated by commas.

  $$\text{term}_1 \rightarrow \text{synonym}_{1a}, \text{synonym}_{1b}...$$

  Every synonym in the document corpus is then substituted with the term, which appears at the first position in the corresponding line.

Vocabulary Preprocessing (Step 4). This step is responsible for applying the same standard text preprocessing to the predefined vocabulary that is used also to preprocess the document corpus. Similarly, the main components here are tokenization, stopwords labeling and token stemming or lemmatization.

Candidate B-Term Extraction (Step 6). After completing the preprocessing steps, the resulting whitelist output is used in Candidate B-term Extraction step for filtering out terms that are not part of the controlled vocabulary.

6 Experiments and Results

This section presents the evaluation of the presented literature based discovery methodology. We have applied different base and ensemble heuristics on two problems: the standard migraine-magnesium literature mining benchmark problem used in the Swanson's experiments [13], and a more recent example of using literature mining for uncovering the nature of relations that might contribute to better understanding of autism, originated in [19,33]. In both cases, our methodology successfully replicated the results known from the literature.

6.1 Experimental Setting

The evaluation was performed based on two datasets (or two domain pairs, since each dataset consists of two domains)—the migraine-magnesium dataset [13] and the autism-calcineurin [33] dataset—which can be viewed as a training and test dataset, respectively. The training dataset is the dataset we employed when developing the methodology, i.e. for creating a set of base heuristics as well as for creating the ensemble heuristic. The results of the evaluation on the training dataset are important, but need to be interpreted carefully due to a danger of overfitting the dataset, as described in [30]. The test dataset is used for the evaluation of the methodology in a real-life setting. The well-researched migraine-magnesium domain pair [13] was used as a training set. In the literature-based discovery process Swanson managed to find more than 60 pairs of articles connecting the migraine domain with the magnesium deficiency via 43 bridging concepts (B-terms), which are listed in Table 6.[14] In testing the developed methodology we aimed at rediscovering the 43 B-terms by ranking them as high as possible in the ranked list of potential B-terms that include Swanson's B-terms and terms that are not in the Swanson's B-term list.

Table 6. B-terms for the migraine-magnesium dataset identified in [13].

1 5 ht
2 5 hydroxytryptamine
3 5 hydroxytryptamine receptor
4 anti aggregation
5 anti inflammatory
6 anticonvulsant
7 antimigraine
8 arterial spasm
9 brain serotonin
10 calcium antagonist
11 calcium blocker
12 calcium channel
13 calcium channel blocker
14 cerebral vasospasm
15 convulsion
16 convulsive
17 coronary spasm
18 cortical spread depression
19 diltiazem
20 epilepsy
21 epileptic
22 epileptiform
23 hypoxia
24 indomethacin
25 inflammatory
26 nifedipine
27 paroxysmal
28 platelet aggregation
29 platelet function
30 prostacyclin
31 prostaglandin
32 prostaglandin e1
33 prostaglandin synthesis
34 reactivity
35 seizure
36 serotonin
37 spasm
38 spread
39 spread depression
40 stress
41 substance p
42 vasospasm
43 verapamil

Table 7. B-terms for the autism-calcineurin dataset identified in [33].

1 synaptic
2 synaptic plasticity
3 calmodulin
4 radiation
5 working memory
6 bcl 2
7 type 1 diabetes
8 ulcerative colitis
9 asbestos
10 deletion syndrome
11 22q11 2
12 maternal hypothyroxinemia
13 bombesin

For the test dataset we used the autism-calcineurin domain pair [33]. Like Swanson, Petrič et al. also provide B-terms, 13 in total (listed in Table 7), whose importance in connecting autism to calcineurin (a protein phosphatase) is discussed and confirmed by the domain expert. In view of searching for B-terms, this dataset has a relatively different dimensionality compared to the migraine-magnesium dataset. On the one hand it has only about one fourth of the B-terms defined, while on the other hand, it contains more than 40 times

[14] Note that Swanson did not state that this was an exclusive list, hence there may exist other important bridging terms which he did not list.

Table 8. Comparison of some statistical properties of the two datasets used in the experiments.

		migraine-magnesium	autism-calcineurin
Retrieval	Source	PubMed	PubMed
	Query terms	"migraine"-"magnesium"	"autism"-"calcineurin"
	Additional conditions	Year < 1988	/
	Part of paper used	Title	Abstract
Document Statistics	Number	8,058 (2,415–5,633)	15,243 (9,365–5,878)
	Doc. with B-term	394 (4.89%)	1,672 (10.97%)
	Avg. words per doc	11	180
Term statistic	Avg. term per doc.	7	173
	Distinct terms	13,525	322,252
	B-term candidates	1,847	78,805
	Defined B-terms	43	13

as many potential B-term candidates. Therefore, the ratio between the actual B-terms and the candidate terms is substantially lower—approximately by factor 160, i.e. the chance to find a B-term among the candidate terms if picking it at random is 160 times lower in the autism-calcineurin dataset then in the magnesium-migraine dataset. Consequently, finding the actual B-terms in the autism-calcineurin dataset is much more difficult compared to the migraine-magnesium dataset.

Both datasets, retrieved from the PubMed database using the keyword query, are formed of titles or abstracts of scientific papers returned by the query. However, we used an additional filtering condition for selecting the migraine-magnesium dataset. For fair comparison we had to select only the articles published before the year 1988 as this was the year when Swanson published his research about this dataset and consequently making an explicit connection between the migraine and magnesium domains.

Table 8 states some properties for comparing the two datasets used in the evaluation. One of the major differences between the datasets is the length of an average document since only the titles were used in the migraine-magnesium dataset, while the full abstracts were used in the autism-calcineurin case. Consequently, also the number of distinct terms and B-term candidates is much larger in the case of the autism-calcineurin dataset. Nevertheless, the preprocessing of both datasets was the same. We can inspect higher numbers in the migraine-magnesium dataset which points to the problem of harder classification of documents in this dataset, which is also partly due to shorter texts.

6.2 Evaluation Procedure

The key aspect of the evaluation is the assessment of how well the proposed ensemble heuristic performs when ranking the terms. Two evaluation measures were used in the evaluation of the developed methodology: the standard Area under the Receiver Operating Characteristic analysis and the amount of B-terms found among the first 5,10, 20, 100, 500 and 2,000 terms in the heuristics' ranked list of terms.

First, we compared the heuristics using the Area under the Receiver Operating Characteristic (AUROC) analysis [34]. The Receiver Operating Characteristic (ROC) space is defined by two axes, where the horizontal axis scales from zero to the number of non-B-terms, and the vertical axis from zero to the number of B-terms. An individual Receiver Operating Characteristic (ROC) curve, representing a single heuristic, is constructed in the following way:

– Sort all the terms by their descending heuristic score.
– For every term of the term list do the following: if a term is a B-term, then draw one vertical line segment (up) in the ROC space, else draw one horizontal line segment (right) on the ROC space.
– If a heuristic outputs the same score for many terms, we cannot sort them uniquely. In such case, we draw a line from the current point p to the point $p+(nb, b)$, where nb is the number of non-B-terms and b is the number of terms that are B-terms among the terms with the same bisociation score. In this way we may produce slanted lines, if such an equal scoring term set contains both B-terms and non-B-terms.

AUROC is defined as the percentage of the area under ROC curve, i.e. the area under the curve divided by the area of the whole ROC space.[15] Besides AUROC we also list the interval of AUROC which tells how much each heuristic varies among the best and the worst sorting of a possibly existing equal scoring term set. This occurs due to the fact that some heuristics do not produce unambiguous ranking of all the terms. Several heuristics assign the same score to a set of terms—including both the actual B-terms as well as non B-terms—which results in the fact that unique sorting is not possible.[16] In the case of equal scoring term sets, the inner sorting is random (which indeed produces different performance estimates), however the constructed ROC curve corresponds to the average ROC curve over all possible such random inner sortings.

From the expert's point of view, the ROC curves and AUROC statistics are not the most crucial information about the quality of a given heuristic. While in general it still holds that a higher AUROC reflects a better heuristic, we are more interested in the ranking from the perspective of the domain expert (the end-user of the our system) who is usually more interested in questions like:

[15] If a heuristic is perfect (it detects all the B-terms and ranks them at the top of the ordered list), we get a curve that goes first just up and then just right with an AUROC of 100%. The worst possible heuristic sorts all the terms randomly regardless of being a B-term or not and achieves AUROC of 50%. This random heuristic is represented by the diagonal in the ROC space.

[16] In such cases, the AUROC calculation can either maximize the AUROC by sorting all the B-terms in front of all the other terms inside equal scoring sets or minimize it by putting the B-terms at the back. The AUROC calculation can also achieve many AUROC values in between these two extremes by using different (e.g., random) sortings of equal scoring sets. Preferable are the heuristics with a smaller interval which implies that they produce smaller and fewer equal scoring sets.

(a) how many B-terms are likely to be found among the first n terms in a ranked list (where n is a selected number of terms the expert is willing to inspect, e.g., 5, 20 or 100), or (b) how much one can trust a heuristic if a new dataset is explored. Therefore, we also performed an evaluation using an alternative user oriented approach, which evaluates the ranking results adapted to the user's needs. This evaluation estimates how many B-terms can be found among the first 5, 10, 20, 100, 500 and 2,000 terms on the ranked list of terms produced by a heuristic.

6.3 Results on the Migraine-Magnesium Dataset

Table 9 shows the comparison of ranking performance for the ensemble and all the base heuristics on the migraine-magnesium dataset. The heuristics are ordered by their AUROC. The second and third column in the table represent heuristics' average AUROC score[17] and its AUROC interval, respectively. When looking at the ensemble heuristic scores in Table 9, we notice that it achieves higher

Table 9. Comparison of base and ensemble heuristics capacity to rank the B-terms at the very beginning of the term list for the migraine-magnesium dataset.

Heuristic name	AUROC		Number of B-terms among top n ranked terms								
	Average	Interval	5	10	20	50	100	200	500	1,000	2,000
outFreqRelSvm	58.78%	1.26%	0.12	0.24	0.48	1	1.63	5.88	14.44	29	43
outFreqRelSum	58.19%	0.65%	0	0.28	0.83	1.82	3.68	6	15	27	43
freqDomnRatioMin	57.34%	4.71%	0.14	0.28	0.57	1.42	2.83	5.66	14	28	43
outFreqRelRf	56.85%	1.50%	0.24	0.48	0.95	2	4.15	6.94	14	29	43
outFreqSum	55.41%	4.06%	0	0	0	0	0	2.44	15.06	27.16	43
outFreqRf	55.20%	11.07%	0	0	0	0	0.4	5.15	14.86	26.34	43
outFreqSvm	55.19%	9.38%	0	0	0	0	0.35	3	14.14	26.12	43
outFreqRelCs	54.29%	1.50%	0	0	1	1	2.69	5.07	11	27	43
freqDomnProdRel	53.23%	3.08%	0	0	0	0	0	6	14	27	43
outFreqCs	52.34%	10.51%	0	0	0	0	0	1.43	15.62	24.67	43
tfidfDomnSum	52.11%	2.69%	0	0	0	0	1	2	11	26.14	43
tfidfAvg	51.31%	3.63%	0	0	1	1.79	3.11	5.75	11.84	20.9	43
freqDomnProd	51.20%	3.36%	0	0	0	0	1	3	13.17	27.16	43
tfidfDomnProd	51.18%	2.69%	0	0	0	0	1	3	13.5	27	43
freqRatio	50.51%	39.26%	0	0	1	1	4	5	11.65	23.09	43
appearInAllDomains	50.00%	50.00%	0.11	0.23	0.46	1.15	2.3	4.6	11.49	22.98	43
tfidfSum	49.65%	3.63%	0	0	0	0	0	1	9	25.36	43
freqTerm	49.60%	3.78%	0	0	0	0	0	1	8.91	25.49	43
freqDoc	49.55%	3.82%	0	0	0	0	0	1	8.03	24.79	43
ensemble	59.05%	0.26%	1	1	1	5	6	9	18.57	28	43

[17] In contrast to the results reported in [4,5], the AUROC scores presented in this chapter take into account only the terms which appear in both domains. This results in lower AUROC scores, which are thus not directly comparable between the studies. The reason for this approach is in the definition of a bridging term, where the term is required to appear in both domain, as it cannot form a connection otherwise.

AUROC value and lower AUROC interval compared to all the other heuristics. As mentioned in Sect. 5.2, the ensemble was constructed using also two not so well performing heuristics (tfidfDomnSum and freqRatio) in order to avoid overfitting on the training domain. This could have had a negative effect to the ensemble performance, however, the ensemble performance was not seriously affected which gives evidence of right decisions made when designing the ensemble.

As mentioned, such AUROC evaluation does not necessarily aligns well with the methodology evaluation from a user's perspective. Therefore, the right side of Table 9 shows the results of an alternative user oriented evaluation approach, which shows how many B-terms were found among the first 5, 10, 20, 50, 100, 200, 500, 1,000 and 2,000 terms on the ranked list of terms produced by a heuristic. The ensemble heuristic, described in Sect. 5.2, performing ensemble voting of six elementary heuristics, resulted in very favorable results on the training migraine-magnesium domain (as seen in Table 9), where one B-term among the first 5 terms, one B-term (no additional B-terms) among the first 20 terms, 6 B-terms (5 additional) among the first 100 terms, 22 B-terms (16 additional) among first 500 terms and all the 43 B-terms (21 additional) among the first 2,000 terms. Thus, e.g., if the expert limits himself to inspect only the first 100 terms, he will find 6 B-terms in the ensemble ranked term list. These results confirm that the ensemble is the best performing heuristics also from the user's perspective. Even though a strict comparison depends on the threshold of how many terms an expert is willing to inspect, the ensemble is always among the best.

6.4 Results of Using a Controlled Vocabulary on the Migraine-Magnesium Dataset

In this section we demonstrate that by using a predefined controlled vocabulary we can increase the heuristics' capabilities to rank the B-terms at the beginning of the term list. We have repeated the experiments on the migraine-magnesium domain, described in Sect. 6.3, except that we now used a predefined vocabulary constructed from MeSH using the "MeSH filter" widget. As we were particularly interested in the bridging terms between migraine—a disease—and magnesium—a chemical element—as well as the circumstances and processes observed between them, we only selected categories [C] *Diseases*, [D] *Chemicals and drugs* and [G] *Phenomena and Processes*. In the experiment we used the workflow shown in Fig. 13. The generated vocabulary was used in the candidate B-term extraction step as a whitelist filter.

The results of the methodology using a controlled vocabulary on the migraine-magnesium domain are presented in Table 11. The comparison of the heuristics' capabilities to rank the B-terms at the beginning of the term list in the migraine-magnesium domain from Tables 9 and 11 shows an advantage of using the controlled vocabulary. By inspecting the number of B-terms found in the ranked first n terms, we notice that using the controlled vocabulary in the migraine-magnesium domain resulted in a much higher concentration of Swanson's B-terms among the best ranked terms.

Table 10. B-terms for the migraine-magnesium dataset identified in [13]. The 17 terms which are crossed out were not part of the used controlled vocabulary, therefore heuristics were unable to identify them as B-term candidates.

1 5 ht
2 5 hydroxytryptamine
3 5 hydroxytryptamine receptor
~~4 anti aggregation~~
5 anti inflammatory
6 anticonvulsant
~~7 antimigraine~~
~~8 arterial spasm~~
~~9 brain serotonin~~
10 calcium antagonist
~~11 calcium blocker~~
12 calcium channel
13 calcium channel blocker
14 cerebral vasospasm
15 convulsion
~~16 convulsive~~
~~17 coronary spasm~~
18 cortical spread depression
19 diltiazem
20 epilepsy
~~21 epileptic~~
~~22 epileptiform~~
23 hypoxia
24 indomethacin
~~25 inflammatory~~
26 nifedipine
~~27 paroxysmal~~
28 platelet aggregation
~~29 platelet function~~
30 prostacyclin
31 prostaglandin
~~32 prostaglandin e1~~
33 prostaglandin synthesis
~~34 reactivity~~
35 seizure
36 serotonin
37 spasm
~~38 spread~~
39 spread depression
~~40 stress~~
41 substance p
~~42 vasospasm~~
43 verapamil

As explained in Sect. 5.4 a predefined controlled vocabulary can greatly reduce the B-term search space. As a side effect, we were unable to: (a) perform AUROC evaluation comparison due to different number of terms in the vocabulary—As a result, Table 11 provides only evaluation which lists the number of B-terms found in the ranked first n terms, (b) detect all B-terms, identified by Swanson (the crossed out B-terms in Table 10 were not part of the used controlled vocabulary); this could be solved using larger controlled vocabularies, though we must be careful not to overfit the vocabulary to the expected results.

On the other hand, results show that using a predefined controlled vocabulary not only increases the efficiency of the heuristic calculation algorithms, but also tends to improve the relevance of top ranked B-terms. Consequently, the described approach enables the user to perform the exploration task more effectively, potentially leading to new discoveries.

6.5 Results on the Autism-Calcineurin Dataset

In this section we show how our methodology performs on a new independent test dataset—the autism-calcineurin domain—which was not used in the development of the methodology. As discussed, the dimensionality of the autism-calcineurin dataset is considerably different and less favorable compared to the migraine-magnesium dataset.

Table 12 shows that the performance of individual base heuristics significantly changes compared to the migraine magnesium dataset (Table 9), however, the ensemble heuristic is still among the best and exposes small uncertainty. This gives us the final argument for the quality of the ensemble heuristic since it outperforms all the other heuristics (except for the *freqRatio* base heuristic) when comparing the AUROC scores, as well as the numbers of B-terms found in the

Table 11. Comparison of base and ensemble heuristics capacity to rank the B-terms at the very beginning of the term list for the migraine-magnesium dataset using a controlled vocabulary.

Heuristic Name	Number of B-terms among top n ranked terms								
	5	10	20	50	100	200	500	1,000	2,000
freqDomnRatioMin	0.59	1.18	2.37	5.92	13.25	20	26	26	26
outFreqSum	0	1	2.75	5	15.53	17.06	26	26	26
freqDomnProdRel	0	1	2	5.67	9	20	26	26	26
outFreqRf	1	1	2	6.28	12.16	17.5	26	26	26
outFreqSvm	1	1	2.5	5.16	11.74	16.79	26	26	26
outFreqCs	0	0	2.45	5.6	10.22	17.06	26	26	26
tfidfDomnSum	0	1	1	4	10	19	26	26	26
freqDomnProd	0	1	1	4	9	19	26	26	26
tfidfDomnProd	0	1	1	4	9	19	26	26	26
outFreqRelRf	0.67	1.33	2	5	7	14.75	26	26	26
freqDoc	0	0	1	2.5	7.82	17.1	26	26	26
tfidfSum	0	0	1	2.25	7.5	17.35	26	26	26
freqTerm	0	0	1	2.25	7.56	17.43	26	26	26
appearInAllDomains	0.39	0.78	1.56	3.9	7.81	15.62	26	26	26
outFreqRelSum	0.42	0.83	1.29	4	9	15	26	26	26
tfidfAvg	0	1.42	2.47	5.63	7	13	26	26	26
outFreqRelSvm	0.45	0.91	1.82	3.25	10	15	26	26	26
outFreqRelCs	0.31	0.63	1	5	7.06	14	26	26	26
freqRatio	0	1	1	2	5.96	14.56	26	26	26
ensemble	1	3	4	9	13	19	26	26	26

most interesting ranked list lengths (up to 20, 100, 500 terms). The ensemble finds one B-term among 10 ranked terms, 2 among 200 and 3 among 500 ranked terms out of the total of 78,805 candidate terms that the heuristics have to rank. The evidence of the quality of the ensemble can be understood if we compare it to a baseline, i.e. the *appearInAllDomn* heuristic which denotes the performance achievable without developing the methodology presented in this work. The baseline heuristic discovers in average only approximately 0.33 B-terms before position 2,000 in the ranked list while the ensemble discovers 6; not to mention the shorter term lists where the ensemble has even a better ratio compared to the baseline heuristic.

6.6 Results of Using a Controlled Vocabulary on the Autism-Calcineurin Dataset

In this section we replicated the experiments, described in Sect. 6.4, using a predefined controlled vocabulary on the autism-calcineurin dataset. Similarly,

Table 12. Comparison of base and ensemble heuristics capacity to rank the B-terms at the very beginning of the term list for the autism-calcineurin dataset.

Heuristic Name	AUROC		Number of B-terms among top n ranked terms										
	Average	Interval	5	10	20	50	100	200	500	1,000	2,000	5,000	all
freqRatio	95.10%	0.16%	1	1	1	1	1	1	1	3	5	8.99	13
tfidfSum	88.78%	0.05%	0	0	0	0	1	1	1	2	4	5	13
tfidfDomnProd	88.61%	0.05%	0	0	0	0	0	0	1	1	4	6	13
tfidfDomnSum	88.33%	0.02%	0	0	0	0	1	1	2	2	4	5	13
freqTerm	87.80%	0.80%	0	0	0	0	1	1	1	2	3	5	13
freqDomnProd	87.69%	0.73%	0	0	0	0	0	0	0	1	2	6	13
freqDomnProdRel	85.77%	0.69%	0	0	0	0	0	0	0	1	1	6	13
outFreqRf	85.05%	7.91%	0	0	0	0	0	1	1	1.34	4.37	7.4	13
outFreqSum	84.33%	5.80%	0	0	0	0	0	1	1	3	4	8.4	13
outFreqCs	80.50%	10.05%	0	0	0	0	0	1	1	1	4	7.17	13
freqDoc	79.01%	2.53%	0	0	0	0	0	1	1	2	2	5	13
outFreqSvm	75.15%	17.55%	0	0	0	0	1	1	1.46	4	4.67	5.44	13
tfidfAvg	73.56%	0.05%	1	1	1	1	1	1	1	1	3	6	13
outFreqRelRf	72.44%	0.03%	0	0	0	0	1	1	1	1	1	2	13
outFreqRelSum	67.24%	0.03%	0	0	0	0	0	1	1	2	2	2	13
outFreqRelCs	64.40%	0.19%	0	0	0	0	0	0	0	0	0	1.49	13
outFreqRelSvm	58.39%	0.17%	0	0	0	0	0	0	0	0	1.25	2	13
appearInAllDomains	50.00%	50.00%	0	0	0	0.01	0.02	0.03	0.08	0.17	0.33	0.83	13
freqDomnRatioMin	24.93%	1.12%	0	0	0	0	0	0	0	0	0	0	13
ensemble	90.10%	0.00%	0	1	1	1	1	2	3	4	6	8	13

Table 13. B-terms for the autism-calcineurin dataset identified by [33]. The four terms which are crossed out were not part of the used controlled vocabulary, therefore heuristics were unable to identify them as B-term candidates.

~~1 synaptic~~
2 synaptic plasticity
3 calmodulin
4 radiation
~~5 working memory~~
6 bcl 2
7 type 1 diabetes
8 ulcerative colitis
9 asbestos
10 deletion syndrome
~~11 22q11 2~~
~~12 maternal hypothyroxinemia~~
13 bombesin

we wanted to increase the heuristics' capabilities (in the workflow illustrated in Fig. 13) to rank the B-terms at the beginning of the term list. We used the same predefined vocabulary as with the migraine-magnesium domain, which was constructed from MeSH using the following categories: [C] *Diseases*, [D] *Chemicals and drugs* and [G] *Phenomena and Processes* were used for building the controlled vocabulary (Table 13).

Inspecting the heuristics' capabilities to rank the B-terms at the beginning of the term list in the autism-calcineurin domain (Tables 12 and 14) shows the advantage of using a controlled vocabulary. The increase in the number of B-terms found in the ranked first n terms when using the controlled vocabulary is even more significant than in the migraine-magnesium domain. The ensemble

Table 14. Comparison of base and ensemble heuristics capacity to rank the B-terms at the very beginning of the term list for the autism-calcineurin dataset using a controlled vocabulary.

Heuristic Name	Number of B-terms among top n ranked terms									
	5	10	20	50	100	200	500	1,000	2,000	5,000
outFreqSvm	0	0	0	0.5	2	4	4.8	7	8.92	9
outFreqSum	0	0	0	0	0	4	5.56	7	8	9
tfidfDomnProd	0	0	0	0	0	3	4	7	9	9
freqDomnProd	0	0	0	0	1	3	4	7	9	9
freqRatio	1	1	1	1	2	3	3.6	6.01	9	9
freqDomnProdRel	0	0	0	0	0	1	4	7	9	9
outFreqCs	0	0	0	0	0	2	6.59	7	7.82	9
tfidfSum	0	1	1	1	1	2	3	7	9	9
tfidfDomnSum	0	1	1	1	1	2	3	7	9	9
freqTerm	0	1	1	1	1	2	3	6.21	9	9
freqDoc	0	1	1	1	1	2	3	6	8	9
outFreqRf	0	0	0	0	0	1	2.65	5.59	6.99	9
outFreqRelSvm	0	0	1	1	1	1	2	3	9	9
tfidfAvg	1	1	1	1	1	2	2	4	7	9
outFreqRelCs	0	0	0	0	0	0	2	3	7	9
outFreqRelSum	0	0	0	0	0	1	1	3	7	9
appearInAllDomains	0.01	0.03	0.06	0.14	0.28	0.55	1.38	2.76	5.52	9
outFreqRelRf	0	0	0	0	0	0	0	2	6	9
freqDomnRatioMin	0	0	0	0	0	0	1	2	6	9
ensemble	1	1	1	2	2	2	4	6	8	9

heuristic finds the first B-term among the top 5 ranked terms (before only among top 10) and the second B-term among the top 50 ranked terms (before only among 200). These results confirm the findings that controlled vocabularies can increase the heuristics' capacities to rank the B-terms at the beginning of the term list and, thus, provide a more efficient exploration task to the end-user of the platform.

7 Conclusions and Future Outlook

This chapter presents the TextFlows platform together with its cross-context literature mining facility, which in combination with the term exploration engine CrossBee supports the expert in advanced document exploration, aimed at facilitating document retrieval, analysis and visualization. The combination of the two systems forms a creativity support tool, helping experts to uncover not yet discovered relations between seemingly unrelated domains from large textual databases. As estimating which terms have a high bisociative potential is a challenging research question, we proposed a complex methodology which was developed as a pipeline of natural language processing an literature based discovery components in the TextFlows platform. The visual programming user interface of TextFlows not only enables the user to tailor the methodology steps to his own needs but also allows experiment repeatability and methodology reuse by other users and developers.

This chapter contributes also the evaluation of a number of specially designed heuristic functions that provide a bisociation score quality estimate for each term. These base heuristics can be—based on the type of term features they exploit—divided into the following sets: frequency based, TF-IDF based, similarity based, and outlier based. Another contribution is the development of the improved ensemble-based heuristic, which employs a set of base heuristics to ensure robustness and stable performance across the datasets. We evaluated the ensemble based methodology on two domains, migraine-magnesium and autism-calcineurin, showing that the proposed methodology substantially reduces the end-user's burden in terms of the length of the term list that needs to be inspected to find some B-terms. Furthermore, it was shown that by using a predefined vocabulary we can increase the heuristics' capacities to rank the B-terms at the beginning of the term list. Indeed, by applying this approach in the migraine-magnesium and autism-calcineurin domains we got a higher concentration of B-terms among the best ranked terms. Consequently, the user is presented with a simpler exploration task, potentially leading to new discoveries.

In future work we will introduce additional user interface options for data visualization and exploration as well as advance the term ranking methodology by adding new sophisticated heuristics which will take into account also the semantic aspects of the data. Besides, we will apply the system to new domain pairs to exhibit its generality, investigate the need and possibilities of dealing with domain specific background knowledge, and assist researchers in different disciplines in their explorations which may lead to new scientific discoveries.

This research perfectly demonstrated the importance of the HCI-KDD [35] approach of combining the best of two worlds for getting insight into complex data, which is particularly important for health informatics research, where the human expertise (e.g. a doctor-in-the-loop) is of great help in solving hard problems, which cannot be solved by automatic machine learning algorithms otherwise [36]. There is much research in this area necessary in the future.

Acknowledgements. This work was supported by the Slovenian Research Agency and the FP7 European Commission FET projects MUSE (grant no. 296703) and ConCreTe (grant no. 611733).

References

1. Koestler, A.: The Act of Creation, vol. 13 (1964)
2. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I., et al.: Fast discovery of association rules. Adv. Knowl. Discov. Data Min. **12**(1), 307–328 (1996)

3. Dubitzky, W., Kötter, T., Schmidt, O., Berthold, M.R.: Towards creative information exploration based on koestler's concept of bisociation. In: Berthold, M.R. (ed.) Bisociative Knowledge Discovery. LNCS (LNAI), vol. 7250, pp. 11–32. Springer, Heidelberg (2012). doi:10.1007/978-3-642-31830-6_2
4. Juršič, M., Cestnik, B., Urbančič, T., Lavrač, N.: Bisociative literature mining by ensemble heuristics. In: Berthold, M.R. (ed.) Bisociative Knowledge Discovery. LNCS (LNAI), vol. 7250, pp. 338–358. Springer, Heidelberg (2012). doi:10.1007/978-3-642-31830-6_24
5. Juršič, M., Cestnik, B., Urbančič, T., Lavrač, N.: Cross-domain literature mining: finding bridging concepts with CrossBee. In: Proceedings of the 3rd International Conference on Computational Creativity, pp. 33–40 (2012)
6. Berthold, M.R. (ed.): Bisociative Knowledge Discovery. LNCS (LNAI), vol. 7250. Springer, Heidelberg (2012)
7. Swanson, D.R.: Medical literature as a potential source of new knowledge. Bull. Med. Libr. Assoc. **78**(1), 29 (1990)
8. Smalheiser, N., Swanson, D., et al.: Using ARROWSMITH: a computer-assisted approach to formulating and assessing scientific hypotheses. Comput. Methods Programs Biomed. **57**(3), 149–154 (1998)
9. Hristovski, D., Peterlin, B., Mitchell, J.A., Humphrey, S.M.: Using literature-based discovery to identify disease candidate genes. Int. J. Med. Inf. **74**(2), 289–298 (2005)
10. Yetisgen-Yildiz, M., Pratt, W.: Using statistical and knowledge-based approaches for literature-based discovery. J. Biomed. Inform. **39**(6), 600–611 (2006)
11. Holzinger, A., Yildirim, P., Geier, M., Simonic, K.M.: Quality-based knowledge discovery from medical text on the web. In: Pasi, G., Bordogna, G., Jain, L.C. (eds.) Qual. Issues in the Management of Web Information. ISRL, vol. 50, pp. 145–158. Springer, Heidelberg (2013)
12. Kastrin, A., Rindflesch, T.C., Hristovski, D.: Link prediction on the semantic MEDLINE network. In: Džeroski, S., Panov, P., Kocev, D., Todorovski, L. (eds.) DS 2014. LNCS (LNAI), vol. 8777, pp. 135–143. Springer, Heidelberg (2014). doi:10.1007/978-3-319-11812-3_12
13. Swanson, D.R.: Migraine and magnesium: eleven neglected connections. Perspect. Biol. Med. **78**(1), 526–557 (1988)
14. Lindsay, R.K., Gordon, M.D.: Literature-based discovery by lexical statistics. J. Am. Soc. Inform. Sci. Technol. **1**, 574–587 (1999)
15. Srinivasan, P.: Text mining: generating hypotheses from medline. J. Am. Soc. Inform. Sci. Technol. **55**(5), 396–413 (2004)
16. Weeber, M., Klein, H., de Jong-va den Berg, L.T.W.: Using concepts in literature-based discovery: simulating swanson's raynaud-fish oil and migraine-magnesium discoveries. J. Am. Soc. Inform. Sci. Technol. **52**(7), 548–557 (2001)
17. Petrič, I., Cestnik, B., Lavrač, N., Urbančič, T.: Outlier detection in cross-context link discovery for creative literature mining. Comput. J. **55**(1), 47–61 (2012)
18. Sluban, B., Juršič, M., Cestnik, B., Lavrač, N.: Exploring the power of outliers for cross-domain literature mining. In: Berthold, M.R. (ed.) Bisociative Knowledge Discovery. LNCS (LNAI), vol. 7250, pp. 325–337. Springer, Heidelberg (2012). doi:10.1007/978-3-642-31830-6_23
19. Urbančič, T., Petrič, I., Cestnik, B., Macedoni-Lukšič, M.: Literature mining: towards better understanding of Autism. In: Bellazzi, R., Abu-Hanna, A., Hunter, J. (eds.) AIME 2007. LNCS (LNAI), vol. 4594, pp. 217–226. Springer, Heidelberg (2007). doi:10.1007/978-3-540-73599-1_29

20. Aggarwal, C.: Outlier Analysis. Springer, Heidelberg (2013)
21. Kranjc, J., Podpečan, V., Lavrač, N.: ClowdFlows: a cloud based scientific workflow platform. In: Flach, P.A., Bie, T., Cristianini, N. (eds.) ECML PKDD 2012. LNCS (LNAI), vol. 7524, pp. 816–819. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33486-3_54
22. Grčar, M.: Mining text-enriched heterogeneous information networks. Ph.D. thesis, Jožef Stefan International Postgraduate School (2015) (To appear)
23. Bird, S.: Nltk: the natural language toolkit. In: Proceedings of the COLING/ACL on Interactive Presentation Sessions, pp. 69–72. Association for Computational Linguistics (2006)
24. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
25. Feldman, R., Sanger, J.: The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press, New York (2007)
26. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Inf. Process. Manage. **24**(5), 513–523 (1988)
27. Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000). doi:10.1007/3-540-45014-9_1
28. Rokach, L.: Pattern classification using ensemble methods. World Scientific (2009)
29. Hoi, S.C., Jin, R.: Semi-supervised ensemble ranking. In: AAAI, pp. 634–639 (2008)
30. Juršič, M.: Text mining for cross-domain knowledge discovery. Ph.D. thesis, Jožef Stefan International Postgraduate School (2015)
31. Juršič, M., Mozetič, I., Erjavec, T., Lavrač, N.: Lemmagen: multilingual lemmatisation with induced ripple-down rules. J. Univ. Comput. Sci. **16**(9), 1190–1214 (2010)
32. Sluban, B., Gamberger, D., Lavrač, N.: Ensemble-based noise detection: noise ranking and visual performance evaluation. Data Mining Knowl. Discov. **28**, 265–303 (2013)
33. Petrič, I., Urbančič, T., Cestnik, B., Macedoni-Lukšič, M.: Literature mining method rajolink for uncovering relations between biomedical concepts. J. Biomed. Inform. **42**(2), 219–227 (2009)
34. Provost, F.J., Fawcett, T., Kohavi, R.: The case against accuracy estimation for comparing induction algorithms. In: ICML, vol. 98, pp. 445–453 (1998)
35. Holzinger, A.: Human-computer interaction and knowledge discovery (HCI-KDD): what is the benefit of bringing those two fields to work together? In: Cuzzocrea, A., Kittl, C., Simos, D.E., Weippl, E., Xu, L. (eds.) CD-ARES 2013. LNCS, vol. 8127, pp. 319–328. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40511-2_22
36. Holzinger, A.: Interactive machine learning for health informatics: when do we need the human-in-the-loop? Springer Brain Inform. (BRIN) **3**, 1–13 (2016)