Bisociative Literature Mining by Ensemble Heuristics

Matjaž Juršič[1], Bojan Cestnik[1,2], Tanja Urbančič[1,3], and Nada Lavrač[1,3]

[1] Jožef Stefan Institute, Ljubljana, Slovenia
[2] Temida d.o.o., Ljubljana, Slovenia
[3] University of Nova Gorica, Nova Gorica, Slovenia
{matjaz.jursic,bojan.cestnik,tanja.urbancic,nada.lavrac}@ijs.si

Abstract. In literature mining, the identification of bridging concepts that link two diverse domains has been shown to be a promising approach for finding bisociations as distinct, yet unexplored cross-domain connections which could lead to new scientific discoveries. This chapter introduces the system CrossBee (on-line Cross-Context Bisociation Explorer) which implements a methodology that supports the search for hidden links connecting two different domains. The methodology is based on an ensemble of specially tailored text mining heuristics which assign the candidate bridging concepts a bisociation score. Using this score, the user of the system can primarily explore only the most promising concepts with high bisociation scores. Besides improved bridging concept identification and ranking, CrossBee also provides various content presentations which further speed up the process of bisociation hypotheses examination. These presentations include side-by-side document inspection, emphasizing of interesting text fragments, and uncovering similar documents. The methodology is evaluated on two problems: the standard migraine-magnesium problem well-known in literature mining, and a more recent autism-calcineurin literature mining problem.

Keywords: Bisociative Literature Mining, Term Ranking, Ensemble Heuristics, Bisociation Score.

1 Introduction

One of the prevailing trends in research and development is professional over-specialization, resulting in islands of deep, but relatively isolated knowledge. On the other hand, many complex problems require knowledge from different domains to be combined. Due to huge amounts of information available on-line it has become difficult to follow even specific literature limited to a single specialization. Searching for cross-domain scientific connections is even harder, as also scientific literature all too often remains closed and cited only in professional sub-communities. As a promising solution to this problem, literature mining offers methods and software tools which support the experts in their knowledge discovery process, especially in searching for yet unexplored connections between different domains. The notion of such connections is closely related to bisociations as defined by Koestler [8] and further refined by Dubitzky et al. [3].

M.R. Berthold (Ed.): Bisociative Knowledge Discovery, LNAI 7250, pp. 338–358, 2012.
© The Author(s). This article is published with open access at SpringerLink.com

A specific type of knowledge discovery problems, addressed in this chapter, is *closed discovery* introduced by Weeber et al., [21] which has been explored previously in literature mining. In closed discovery we start with a hypothesis that two particular concepts usually investigated in separate literatures are connected. We search for supportive evidence for this by investigating available literatures about these two concepts. As suggested already by Swanson [16], this can be done by identifying interesting bridging terms (b-terms) appearing in both literatures and bearing a potential of indirectly connecting the two concepts under investigation. Although being time-consuming, searching for terms appearing in both literatures is not the main problem. The main issue which also motivated the research presented in this chapter is the fact that a list of terms shared by the two literatures can be very long. Estimating which of the terms have higher potential for interesting discoveries is an interesting research question, important for practical applications.

Narrowing the list of candidate bridging terms can be done in different ways. For example, in the RaJoLink methodology presented by Petrič et al. [11] the list of interesting terms is effectively filtered according to MeSH (Medical Subject Headings) categories; in the next step the expert checks which of the remaining terms seem to be promising. In spite of MeSH filtering, the list of interesting terms can still be long and estimating the potential of a particular bridging term candidate to lead to useful bisociations is based on the expert's knowledge and intuition. The expert's involvement assures that the search is guided towards promising bridging concepts which are meaningful and interesting for the expert [11]. Therefore, we believe that experts' involvement should remain an important part of the process. However, in order to ensure that the expert's inspection of the list of candidate bridging terms is made easier, our main motivation was to automatically estimate the bisociation potential of term candidates and rank the terms.

In the methodology proposed in this work, we estimate the bisociation potential of a term by calculating its *bisociation score*. To this end, different heuristics were developed (see [7]), which are summarized in this chapter. As the experiments described in this chapter show the choice of the right heuristic for a particular domain is far from being trivial. A solution, proposed in this work, is to combine multiple heuristics into an ensemble heuristic which is less sensitive to the variability of domain characteristics.

Ensemble learning is a known approach used in machine learning for combining predictions of multiple models into one final prediction. It is well known [2] 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, e.g., Bagging, Boosting, Majority voting, Random forest, Naïve bayes, etc. (see [14]). However, these approaches are usually used for the problem of classification while the core problem presented in this work is ranking. Nevertheless, as information retrieval and especially ranking of web pages by search engines are becoming more and more popular also the ensemble ranking is gaining research attention, e.g., [4, 6].

To evaluate the proposed methodology, implemented in the on-line CrossBee (Cross-Context Bisociation Explorer) system, we applied it to two problems. The first

one is the well-known migraine-magnesium example [16, 17] which represents a gold standard in literature mining and served as a testing dataset also in more recent studies [20]. To prevent overfitting the given literature pair and to show the performance in a more complex case, we performed the evaluation of the proposed methodology in the autism-calcineurin problem introduced in [19, 12, 13].

This chapter is structured as follows. In Section 2 the problem of ranking potential b-terms according to their bisociation potential is defined in detail. Section 3 describes the newly introduced ranking methodology and deals with heuristics, the main emphasis being on the proposed ensemble heuristic. In Section 4, the proposed methodology is evaluated through the migraine-magnesium and autism-calcineurin experiments. Section 5 presents a new on-line software tool CrossBee which implements the methodology and provides additional functionalities, making expert's knowledge discovery process easier and more efficient. The chapter concludes with a discussion and plans for further work.

2 Problem Description

The problem addressed in this work is to help the domain expert to effectively find bisociations between two domains presented by two sets of text documents. The main inputs to this task are two sets of documents – one for each of the examined domains. The top-level problem is split, mainly for the reason of evaluation, in two subproblems as follows:

- Develop a *methodology for identifying bisociations* which (among various patterns of bisociation, identified by Dubitzky et al. in [3]) identifies and ranks the key bridging concepts (also named bridging terms or b-terms) that provide the expert with clues about the potential bisociations. The evaluation of this subproblem is based on defining quality values of different solutions which can then be compared to each other. In this way one is able to evaluate the improvements made over the previously existing solutions.
- Create a *system which can support the expert* not only by providing results of the b-term identification methodology but also by adding multiple layers of information to plain data (documents). The added information can be used for human exploration and judgment whether the connections suggested by b-terms are indeed bisociations. The evaluation of this subproblem is slightly less clear, however, by setting an experiment and observing the effectiveness of the expert using the system, one can approximately estimate the quality of different solutions.

3 Methodology for Bridging Concept Identification and Ranking

This section describes the methodology for identifying and ranking of terms according to their potential for being b-terms. The basics of our methodology was developed with the purpose of using potential bridging concepts in the construction of information networks from text documents (see [7] for details), as well as for b-term

identification and ranking in our new CrossBee system described in more detail in Section 5 below.

The input to the procedure for b-term identification and ranking consists of two sets of documents – one for each domain. Input documents can be either in the standard form of running text, e.g., titles and abstracts of scientific documents or in the form of partly preprocessed text, e.g., text with already recognized named entities. The output of the procedure is a ranked list of all identified interesting terms. The output list of terms is ordered according to terms' *bisociation score* which is the estimate of a potential that the evaluated term is indeed a b-term which can trigger a bisociation. Our solution to the presented problem of b-term identification and ranking is based on the following three procedural steps:

1. *Preprocess input documents:* Employ state of the art approaches for text preprocessing to extract the most of useful information present in raw texts. Documents are transformed into the bag-of-words [5] feature vector representation, where features represent the terms or concepts. The extracted concepts are *identified as candidate b-terms* and ranked in the next step. More details on text preprocessing are presented in [7].
2. Score *candidate b-terms:* Take the list of candidate b-terms generated in the document preprocessing step and evaluate their b-term potential by calculating the *bisociation score* for each term from the list. This is performed in two steps:
   a. *Employ the base heuristics*: Based on the feature vector representation and some other properties of documents and terms, use specially designed base heuristic functions to score the terms. The output of a base heuristic (the term's score) evaluates the term's potential of being a b-term (see [7] for details).
   b. *Employ the ensemble heuristic*: Scores of base heuristics are integrated into one ensemble heuristic score which represents the final output of the scoring candidate b-terms step and is used as the estimate of the term's bisociation potential. The exact procedure for calculating the ensemble bisociation score is explained in more detail below in this section.
3. *Output the ranked list of b-terms:* Order the list of terms according to the descending order of the calculated bisociation score and return the ranked list of terms with their bisociation scores. This step is elementary and does not need to be presented in detail.

The rest of this section deals with the second step sketched above.

3.1 Base Heuristics

We use the term "heuristic" or "heuristic function" to name a function that numerically evaluates term's quality in the view of its bisociation potential. Ranking all the terms using the scores calculated by an ideal heuristic should result in finding all the b-terms together at the top of such a sorted list. This ideal scenario is generally

not realistic; however, ranking by heuristic scores (either ascending or descending) should still increase the proportion of b-terms at the top of the term list.[1]

In [7] we defined a heuristic as a function with two inputs: (a) a set of documents labeled with two domain labels and (b) a term t appearing in these documents; and one output, i.e., a score that estimates the term's bisociation potential. We here list the heuristics with short descriptions only, while the detailed heuristics definition along with their equations are provided in [7].

BoW Heuristics

The heuristics in the group of BoW (bag-of-words) work in a similar way – they manipulate the data present in document vectors to derive the terms' bisociation score. They can be divided into three subgroups:

Term frequency based

(1) $\boldsymbol{freqTerm(t)}$**:** dataset term frequency,
(2) $\boldsymbol{freqDoc(t)}$**:** dataset document frequency,
(3) $\boldsymbol{freqRatio(t)}$**:** dataset term to document frequency ratio,
(4) $\boldsymbol{freqDomnRatioMin(t)}$**:** minimum of domain term frequencies ratio,
(5) $\boldsymbol{freqDomnProd(t)}$**:** product of domain term frequencies,
(6) $\boldsymbol{freqDomnProdRel(t)}$**:** product of domain term frequencies relative to a dataset term frequency.

Tf-idf based

(7) $\boldsymbol{tfidfSum(t)}$**:** sum of document tf-idf weights of a term in a dataset,
(8) $\boldsymbol{tfidfAvg(t)}$**:** average of document tf-idf weight of a term in a dataset,
(9) $\boldsymbol{tfidfDomnProd(t)}$**:** product of domain centroid tf-idf weights of a term,
(10) $\boldsymbol{tfidfDomnSum(t)}$**:** sum of domain centroid tf-idf weights of a term.

Similarity based

(11) $\boldsymbol{simAvgTerm(t)}$**:** similarity of a term to an average dataset document – the distance of a term to the dataset centroid,
(12) $\boldsymbol{simDomnProd(t)}$**:** product of similarities of a term to domain centroids,
(13) $\boldsymbol{simDomnRatioMin(t)}$**:** min of similarities of a term to domain centroids.

Outlier Heuristics

The outlier heuristics focus on outlier documents since they frequently embody new information that is often hard to explain in the context of existing knowledge. 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. In the definition of outlier heuristics we used three outlier sets of documents corresponding to the three different underlying document

[1] Note that regardless of the choice, all the heuristics give score 0 to all the terms which appear only in one of the two domains, as these terms have zero potential for bisociation between the two domains.

classification algorithms used for outlier detection: Centroid Similarity classifier (CS), Random Forest classifier (RF), and Support Vector Machine classifier (SVM). Research focused in detecting the outlier documents was performed in [15] and two of the sets, namely RF and SVM were provided by that research. The detection of CS outlier documents was implemented directly in CrossBee using the principles described in [15] but using the Centroid Similarity classifier. The resulting heuristics are:

Based on absolute term frequency in outlier sets

(14) $\boldsymbol{outFreqCS(t)}$**:** term frequency in CS outlier set,
(15) $\boldsymbol{outFreqRF(t)}$**:** term frequency in RF outlier set,
(16) $\boldsymbol{outFreqSVM(t)}$**:** term frequency in SVM outlier set,
(17) $\boldsymbol{outFreqSum(t)}$**:** sum of term frequencies in all three outlier sets.

Based on relative term frequency in outlier sets

(18) $\boldsymbol{outFreqRelCS(t)}$**:** relative frequency in CS outlier set,
(19) $\boldsymbol{outFreqRelRF(t)}$**:** relative frequency in RF outlier set,
(20) $\boldsymbol{outFreqRelSVM(t)}$**:** relative frequency in SVM outlier set,
(21) $\boldsymbol{outFreqRelSum(t)}$**:** sum of relative term frequencies in all three outlier sets.

Baseline Heuristics

We defined two heuristics which are supplementary and serve as baselines:

(22) $\boldsymbol{random(t)}$**:** random number in the interval [0,1),
(23) $\boldsymbol{appearInAllDomn(t)}$**:** a better baseline heuristic which separates two classes of terms, the ones that appear in both domains and the ones that appear in one domain only. The terms that appear in one domain only have a strictly lower heuristic score that those appearing in both. The inner scores of terms inside these two classes are still random numbers.

3.2 Ensemble Heuristic

An ensemble heuristic is a heuristic which combines results of multiple base heuristics into one aggregated result. This work extends the methodology presented in our previous work [7] with an ensemble heuristic due to identified problematic aspect of using a single heuristic for final ranking. The problem arises from the fact that the process of selection of a single heuristic is prone to overfitting the training dataset which results in heuristics' performance instability across other datasets. As long as our experiments were performed only on a single dataset, i.e., the migraine-magnesium dataset, the results of the selected single heuristic, i.e., the (21)outFreqRelSum which proved to be the best heuristic on that dataset were stable, even if we used various modifications of data preprocessing, removed random documents from the set, randomly deleted words from documents or did some other data perturbations.

One possible 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 such approach; however, the results were even more overfitted to the training domain used in our study. Consequently, we decided to manually – based on experiences 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 only describe the final solution, presented in the following subsections.

Ensemble Construction

The ensemble heuristic results in the *ensemble score*, constructed from two parts: the *ensemble voting score* and the *ensemble position score* which are summed together to give the final ensemble score.

- 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[2] in its ranked list of terms and zero votes to all the other terms ($s_{t_j,h_i}^{vote} = 0$). Formally, the ensemble voting score of a 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:
$$s_{t_j}^{vote} = \sum_{i=1}^{k} s_{t_j,h_i}^{vote} = \sum_{i=1}^{k} \begin{cases} 1: p_j < n/3, \\ 0: otherwise \end{cases}.$$
Therefore, each term can get a score $s_{t_j}^{vote} \in \{0, 1, 2, \ldots, k\}$, 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_{t_j,h_i}^{pos} is calculated as $(n - p_j)/n$, 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:
$$s_{t_j}^{pos} = \frac{1}{k}\sum_{i=1}^{k} s_{t_j,h_i}^{pos} = \frac{1}{k}\sum_{i=1}^{k} \frac{(n - \mathrm{p}_j)}{n}.$$
- The final *ensemble score* is computed as:
$$s_t = s_t^{vote} + s_t^{pos}.$$

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

[2] The voting threshold is one third (1/3) of the terms which appear in both domains (not one third of all the terms). It was set empirically based on the evaluation of the ensemble heuristic on the migraine-magnesium domain.

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.

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.

The described method for ensemble score calculation is illustrated in Example 1. In the upper left table the base heuristics scores are shown for each term. The next table 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 the upper right table. For example, all terms at position 2, i.e., t_1, t_6, and t_6, get voting score 1 and position score 4/6. The central table below shows the exact equation how these individual base heuristics' voting and position scores are combined for each term. The table at the bottom displays the list of terms ranked by the calculated ensemble scores.

	Base scores		
Term	h_1	h_2	h_3
t_1	0.93	0.46	0.33
t_2	0.26	0.15	0.10
t_3	0.51	0.22	0.79
t_4	0.45	0.84	0.73
t_5	0.41	0.15	0.11
t_6	0.99	0.64	0.74

Base heuristic scores

	Base ranking		
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

Terms ranked by base heuristics

	Voting score	Position score
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

Voting and position scores based on positions in the ranked lists

	Voting score sum	+	Pos. score average	=	Ensemble 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

Calculation of ensemble heuristic score

t_6 (3.72), [t_2, t_3] (1.56), t_1 (1.50), t_5 (0.11), t_2 (0.06)

Ranked list of terms produced by the ensemble

Example 1. Ensemble construction illustrated on a simple example with six terms and three heuristics. The last table states the result – the ranked list of terms.

Selecting Base Heuristics for the Ensemble

Another important decision when constructing the ensemble is the selection of base heuristics. Table 1 shows the results that influenced our decision which base heuristics to select. The measure used for heuristic performance comparison is the AUC (area under ROC) presented and discussed already in [7]. Our final set of heuristics included in the ensemble is the following:

- [19]outFreqRelRF
- [18]outFreqRelCS
- [10]tfidfDomnSum
- [20]outFreqRelSVM
- [17]outFreqSum
- [3]freqRatio

Table 1. Comparison of the results (presented and discussed already in [7]) for the base heuristics ordered by the quality – AUC. The first column states the name of the heuristic; the second displays the AUC. The heuristics chosen for the ensemble are shown in italics.

Heuristic	AUC				
[21]outFreqRelSum	95,33%	[16]outFreqSVM	94,70%	[5]freqDomnProd	93,42%
[19]*outFreqRelRF*	95,24%	[14]outFreqCS	94,67%	[3]*freqRatio*	93,35%
[20]*outFreqRelSVM*	95,06%	[4]freqDomnRatioMin	94,36%	[23]appearInAllDomn	93,31%
[18]*outFreqRelCS*	94,96%	[10]*tfidfDomnSum*	93,85%	[12]simDomnProd	93,27%
[17]*outFreqSum*	94,96%	[6]freqDomnProdRel	93,71%	[1]freqTerm	93,20%
[8]tfidfAvg	94,87%	[13]simDomnRatioMin	93,58%	[2]freqDoc	93,19%
[15]outFreqRF	94,73%	[7]tfidfSum	93,58%	[11]simAvgTerm	92,71%
		[9]tfidfDomnProd	93,47%	[22]random	50,00%

Our initial idea was to choose one (possibly the best performing) heuristic form each set. The rationale behind this idea was to include the top performing heuristics that are as independent as possible. In such a way, the combined information provided by the constructed ensemble was expected to be higher than the information contributed by the individual heuristics. However, certain additional decisions were made to maximize ensemble performance on the migraine-magnesium dataset as well as due to trying not to overfit this dataset:

- The first observation (see Table 1) is that all outlier heuristics based on relative term frequency, i.e., [19]outFreqRelRF, [20]outFreqRelSVM, and, [18]outFreqRelCS perform very well. Actually the only heuristic that is better is the [21]outFreqRelSum which is the combination of all these three. As we want to emphasize the power of this best performing set, we include all three heuristics into the ensemble instead of only [21]outFreqRelSum. So they get more votes and a chance to over-vote some other – not so well performing – heuristics.
- A representative heuristic of the second outlier heuristic set, based on absolute term frequency, is [17]outFreqSum which is not only the best performing of this set, but also integrates the votes of other three heuristics from this set and is therefore the best candidate.
- Representatives of BoW heuristics based on frequency and tf-idf were chosen in a way which tries to avoid overfitting the migraine-magnesium dataset. We chose [3]freqRatio and [10]tfidfDomnSum with the reasoning that they are not among the best performing on the training dataset (but we expect them to

perform better on other datasets) and will therefore act as a counterweight to prevent overfitting.

– We completely discarded all the heuristics of the type similarity, as their performance is in the range of the baseline heuristic $^{(23)}$appearInAllDomn.

Table 2. B-terms for the autism-calcineurin dataset identified by Petrič et al. [11]

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

4 Evaluation of the Methodology

This section presents the evaluation of the presented base and ensemble heuristics. The key result of this evaluation is the assessment how well the proposed ensemble heuristic performs when ranking the terms from the perspective of the domain expert who acts as the end-user of the CrossBee system. From the expert's point of view, the ROC curves and AUC statistics (as used and described in [7]) are not the most crucial information about the quality of a single heuristic – even though, in general, a better ROC curve reflects a better heuristic. Usually the user is interested in questions like: (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. This section provides the evaluation of the heuristics in terms of their performance on a training dataset as well as on a new experimental dataset.

4.1 Experimental Setting

The experimental setting is related to the one in [7] and [15]. The evaluation was performed based on two datasets (or two domain pairs, since each dataset consists of two domains), which can be viewed as a training and test dataset. The training dataset is the dataset we employed when developing the methodology, i.e., for creating a set of base heuristics in [7], as well as for creating the ensemble heuristic presented in this work. The results of the evaluation on the training dataset are important, but needs to be interpreted carefully due to a danger of overfitting the dataset. The test dataset is used for the evaluation of the methodology in a broader (non-dataset biased) scenario.

As the training data we used the well-researched *migraine-magnesium* domain pair which was introduced by Swanson in [16] and was later explored in [17, 18, 20, 11] and others. 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). Using the developed methodology we tried to rank these 43 b-terms (listed in Table 1 in [7]) as high as possible among other terms which are not marked as b-terms. Since Swanson does not state that this is an

exclusive list, there may be also other important bridging terms which he did not list. Consequently, there are two obvious reasons for our results not showing 43 b-terms as the first 43 terms on the ensemble's ranked list. The first reason is a non-optimal ensemble performance and the second reason is that some other terms – not listed by Swanson – may be equally important for bridging the two domains.

For the training dataset we used the *autism-calcineurin* domain pair which was introduced and initially researched by Urbančič et al. [19] and later also in [11, 12]. Like Swanson, Petrič et al. [11] also provide b-terms, 13 in total (listed in Table 2), whose importance in connecting autism to calcineurin (a protein phosphatase) is discussed and confirmed by the domain expert. In the 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 approximately one fourth of the b-terms defined, while on the other hand, it contains more than 40 times as many potential b-term candidates. Therefore, the ratio between b-terms and 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. We needed 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 3 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 – due to matching the properties of experiments of original research [16, 19] on these two datasets. Consequently, also the number of distinct terms and b-term candidates is much larger in

Table 3. Comparison of 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
Docum. Statistics	Number	8,058 (2,415-5,633)	15,243 (9,365-5,878)
	Doc. with b-term	394 (4.89%)	1672 (10.97%)
	Avg. words per doc.	11	180
	Outliers (CS-SVM-RF)	(505 - 362 - 896)	(377 - 292 - 142)
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

the case of the autism-calcineurin dataset. Nevertheless, the preprocessing of both datasets was the same with the exception of outlier document identification. For the needs of RF and SVM outlier based heuristics we used the outlier documents identified by Sluban et al. [15] since we did not implement RF and SVM classifiers ourselves. Thus, our outlier heuristics results are completely aligned with the results provided in [15] for both datasets; however, Sluban et al. used slightly different document preprocessing for each of the two datasets. Table 3 also shows the exact number of outliers identified in each dataset. We can inspect higher numbers in the migraine-magnesium dataset which points to the problem of harder classification of documents in this dataset – this is also partly due to shorter texts.

4.2 Results in the Migraine-Magnesium Dataset

Fig. 1 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 AUC. Black dots along with percentages show the heuristic's AUC performance. Gray bars around AUC central point shows the interval of a heuristics' AUC result, explained below.

The property of heuristics having AUC on the interval and not as a fixed value is 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

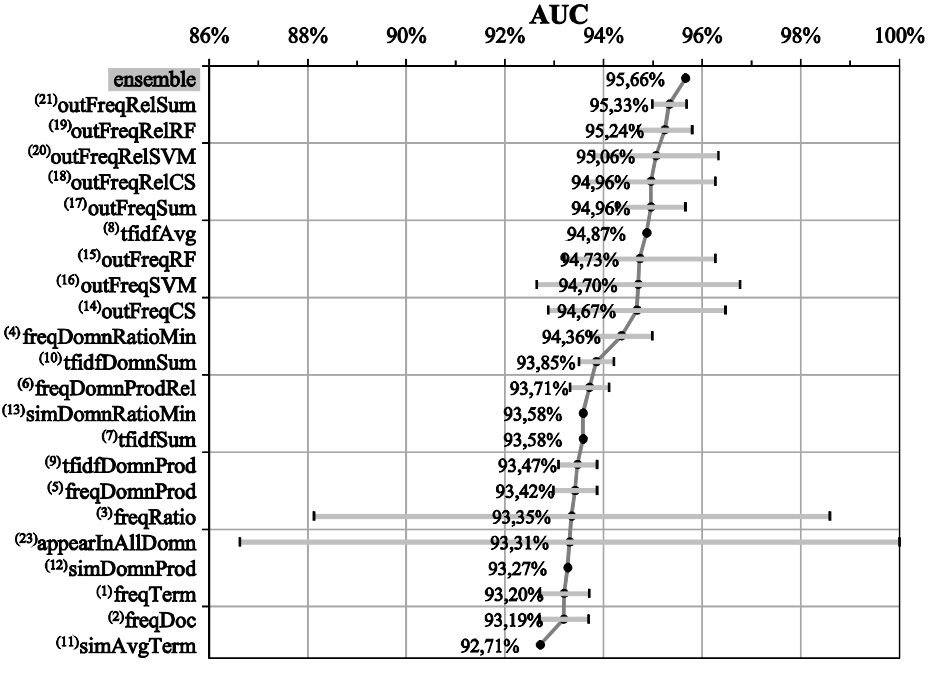


Fig. 1. Graphical representation of the AUC measure for all the individual heuristics and the ensemble heuristic on the migraine-magnesium dataset

b-terms as well as non b-terms – which results in a fact that unique sorting is not possible (i.e., see equal ensemble scores for terms t_2 and t_3 in Example 1). In such cases, the AUC calculation can either maximize the AUC 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 AUC calculation can also achieve many AUC values in between these two extremes by using different (e.g., random) sorts of equal scoring sets. Therefore, an interval bar of AUC shows the interval which contains all the possible AUC values and a black dot shows the interval's middle point which represents the average AUC over a large number of random sorts of equal scoring sets.

Fig. 1 shows no surprises among the base heuristics, since the results are equal to those presented in our previous work (see [7]), however, when focusing on the ensemble heuristic, we notice that it is better in both, higher AUC value and lower AUC interval compared to all the other heuristics. We constructed the ensemble using also two not so well performing heuristics ($^{(10)}$tfidfDomnSum and $^{(3)}$freqRatio) in order to avoid overfitting on the training domain. This could have a negative effect to the ensemble performance, however, the ensemble performance was not seriously affected which signals an evidence on the right decisions when designing the ensemble.

As stated in the introduction of this section, we are mostly interested in the heuristics quality from the end user's perspective. Such evaluation of heuristics quality is shown in Fig. 2, where the length of colored bars tells how many b-terms were found among the first 5, 20, 100, 500 and 2000 terms on the ranked list of terms

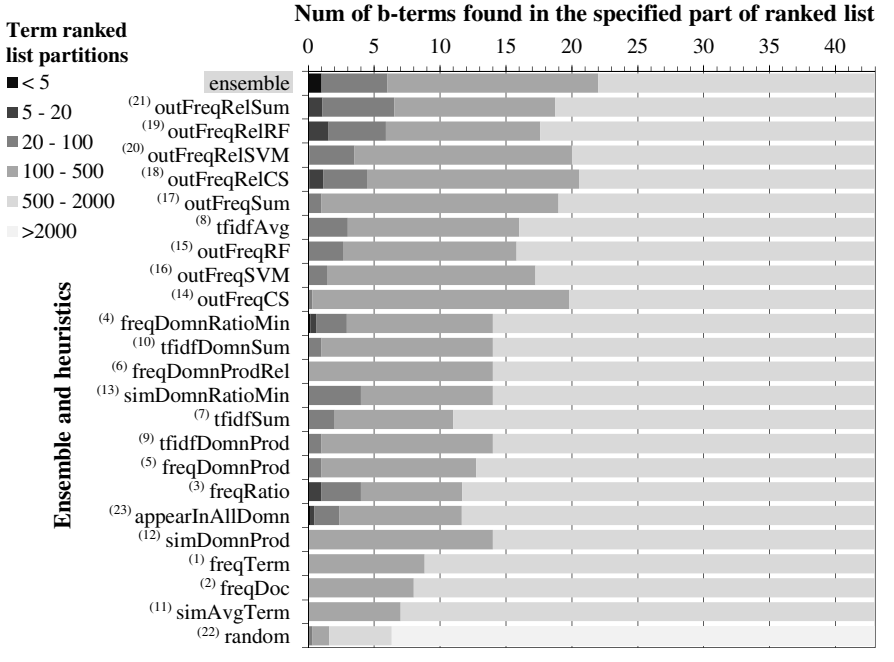


Fig. 2. Comparison of the ensemble and base heuristics capacity to rank the b-terms at the very beginning of the terms list for the migraine-magnesium dataset

produced by a heuristic. We can see that the ensemble finds one b-term among the first 5 terms (the darkest gray bar), one b-term – no additional b-terms – among the first 20 terms (no bar), 6 b-terms – 5 additional – among the first 100 terms (lighter gray bar), 22 b-terms – 16 additional – among first 500 terms (even lighter gray bar) and all the 43 b-terms – 21 additional – among the first 2000 terms (the lightest gray bar). Thus, if the expert limits himself to inspect only the first 100 terms, he will find 6 b-terms in the ensemble list, slightly more than 6 in the [21]outFreqRelSum list, 6 in the [19]outFreqRelRF, and so on. Results in Fig. 2 also give us the confirmation that the ensemble is among the best performing heuristics also from the user's perspective. Even though a strict comparison depends also on the threshold of how many terms an expert is willing to inspect, the ensemble is always among the best.

4.3 Results in Autism-Calcineurin Dataset

Fig. 3 shows how our methodology works on a new independent test dataset which was not used in the development of our methodology. As discussed, the dimensionality of the autism-calcineurin dataset is considerably different and less favorable compared to the migraine-magnesium dataset. This is evident also when observing Fig. 3, since the performance of individual base heuristics significantly changes. Some of the originally best performing heuristics, e.g., based on relative frequency in outlier sets are now among the worst and the other types, e.g., tf-idf based that were not performing well before, are now among the best. The most important observation is

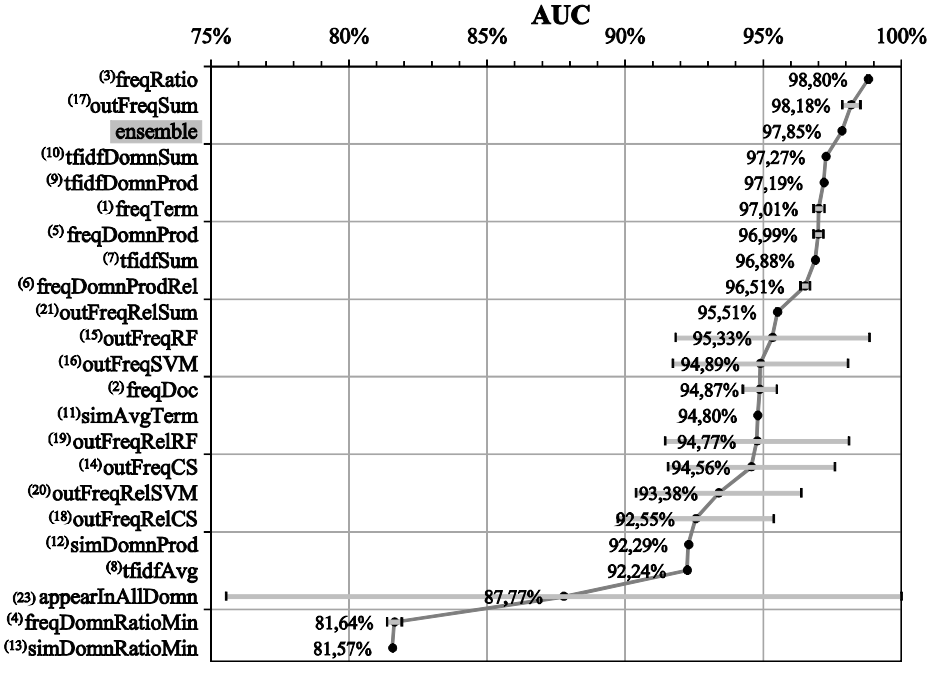


Fig. 3. Graphical representation of the AUC measure for all the individual heuristics and the ensemble heuristic on the autism-calcineurin dataset

that the ensemble heuristic is still among the best (placed after [3]freqRatio and [17]outFreqSum) and preserves a zero AUC interval. Otherwise, we can notice a slight AUC increase of the best performing heuristics which is very positive since the candidate term list is much longer now and we expect we will find the same number of b-terms much later in the candidate term list compared to the migraine-magnesium dataset.

The last result in this section is the user oriented visualization of heuristics performance shown in Fig. 4. This gives us the final argument for the quality of the ensemble heuristic since it outperforms or at least equals to all the other heuristics on the most interesting ranked list lengths (up to 20, 100, 500 terms). The ensemble finds one b-term among 20 ranked terms, 2 among 100 and 3 among 500 ranked terms. At a first sight, this may seem a bad performance, but, note that there are 78,805 candidate terms which the heuristics have to rank. The evidence of the quality of the ensemble can be understood if we compare it to the [23]appearInAllDomn heuristic which is the baseline heuristic and represents the performance which is achievable without developing the methodology presented in this work. The [23]appearInAllDomn heuristic discovers in average only approximately 0.33 b-terms before position 2000 in the ranked list while the ensemble discovers 5 – not to mention the shorter term lists where the ensemble is relatively even better compared to the [23]appearInAllDomn heurisitc.

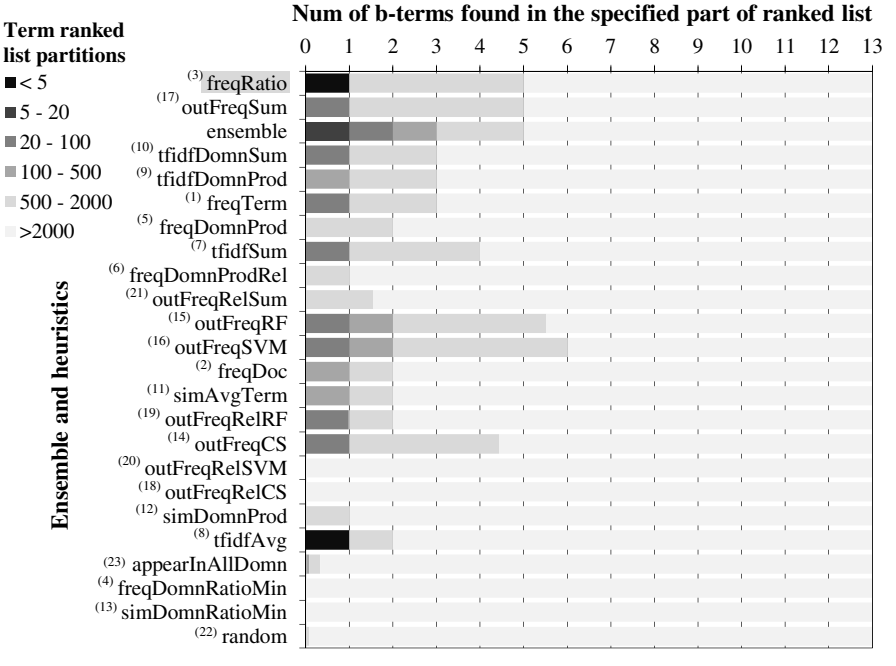


Fig. 4. Comparison of the ensemble and base heuristics capacity to rank the b-terms at the very beginning of the terms list for the autism-calcineurin dataset. The longer the dark part, the more b-terms a heuristic ranks at the specified partition of the ranked list.

5 The CrossBee System

Besides introducing the methodology for identifying bisociations, our aim was to create a system which helps the experts when searching for hidden links that connect two seemingly unrelated domains. We designed and implemented an online system CrossBee – a *Cross-Context Bisociation Explorer*. Initially, the system was designed as an online implementation of the ensemble ranking methodology presented in this chapter. To this core functionality, we have added various supplementary functionalities and content presentations which effectively turn CrossBee into a user-friendly tool for ranking and exploration of perspective cross-context links. This enables the user not only to spot but also to efficiently investigate cross-domain links pointed out by our ensemble ranking methodology.

This section presents CrossBee by describing its most important functionality and a typical use case example. The CrossBee system is built on top of the TexAs (Text Assistant) library created for the needs of our previous work [7]. From this perspective CrossBee is firstly, a functional enhancement (ensemble functionality) of TexAs and secondly, a wrapping of this functionality into a practical web user interface especially designed for the requirements of bisociation discovery (available online at the web address http://crossbee.ijs.si). Note that future versions of CrossBee might not be visually identical as the current version presented here. Nevertheless, the core ensemble ranking algorithm, the two datasets and the results presented in this work will remain available in the future by providing a link to the application, data and settings compatible with this chapter in order to ensure the repeatability of the experiments.

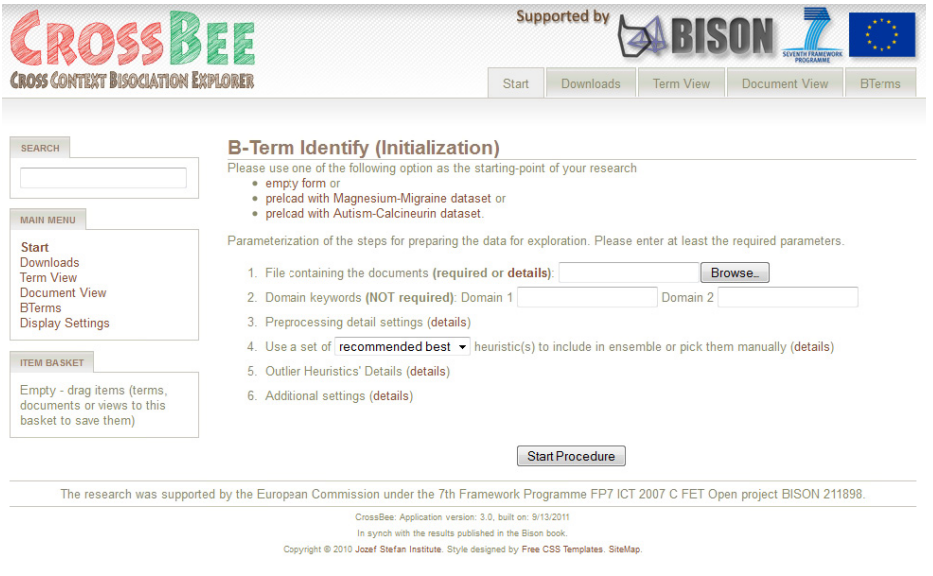
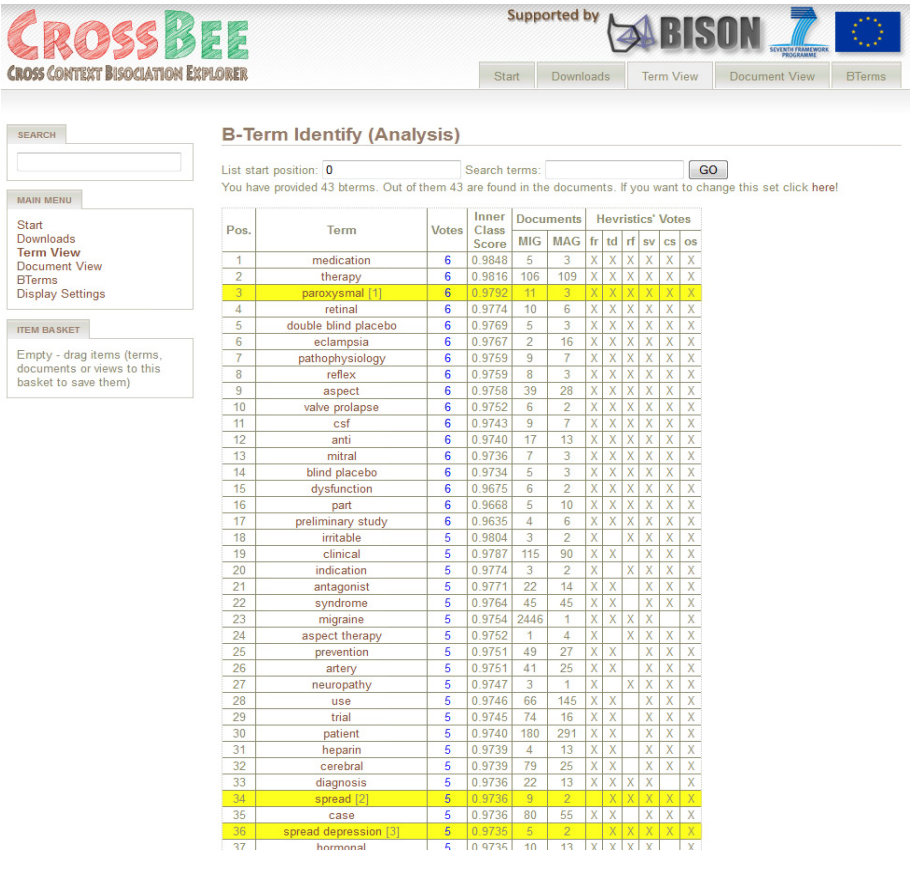


Fig. 5. Home page of the CrossBee system. The user starts an exploration at this point by inputting documents of interest and by tuning the parameters of the system.

5.1 A Typical Use Case

The most standard use case – as envisioned by CrossBee authors – is the following:

1. Prior to starting the process of bisociation exploration, the user needs to prepare the input file of documents. The prescribed format of the input file is kept simple to enable all users, regardless of their computing skills, to prepare the file of documents of their interest. Each line of this file contains exactly three tab-separated entries: (a) the document identification number, (b) the domain acronym, and (c) the document text.
2. The user starts at the initialization page (see Fig. 5) which is the main entry point to the system. The minimal required user's input at this point is a file of documents from two domains. The other options available to the user at this point include specifying the exact preprocessing options, specifying the base heuristics to be used in the ensemble, specifying outlier documents identified by an external outlier detection software (e.g., [15]), defining the already known



Pos.	Term	Votes	Inner Class Score	Documents		Hevristics' Votes					
				MIG	MAG	fr	td	rf	sv	cs	os
1	medication	6	0.9848	5	3	X	X	X	X	X	X
2	therapy	6	0.9816	106	109	X	X	X	X	X	X
3	paroxysmal [1]	6	0.9792	11	3	X	X	X	X	X	X
4	retinal	6	0.9774	10	6	X	X	X	X	X	X
5	double blind placebo	6	0.9769	5	3	X	X	X	X	X	X
6	eclampsia	6	0.9767	2	16	X	X	X	X	X	X
7	pathophysiology	6	0.9759	9	7	X	X	X	X	X	X
8	reflex	6	0.9759	8	3	X	X	X	X	X	X
9	aspect	6	0.9758	39	28	X	X	X	X	X	X
10	valve prolapse	6	0.9752	6	2	X	X	X	X	X	X
11	csf	6	0.9743	9	7	X	X	X	X	X	X
12	anti	6	0.9740	17	13	X	X	X	X	X	X
13	mitral	6	0.9736	7	3	X	X	X	X	X	X
14	blind placebo	6	0.9734	5	3	X	X	X	X	X	X
15	dysfunction	6	0.9675	6	2	X	X	X	X	X	X
16	part	6	0.9668	5	10	X	X	X	X	X	X
17	preliminary study	6	0.9635	4	6	X	X	X	X	X	X
18	irritable	5	0.9804	3	2	X		X	X	X	X
19	clinical	5	0.9787	115	90	X	X		X	X	X
20	indication	5	0.9774	3	2	X		X	X	X	X
21	antagonist	5	0.9771	22	14	X	X		X	X	X
22	syndrome	5	0.9764	45	45	X	X		X	X	X
23	migraine	5	0.9754	2446	1	X	X	X	X		X
24	aspect therapy	5	0.9752	1	4	X		X	X	X	X
25	prevention	5	0.9751	49	27	X	X		X	X	X
26	artery	5	0.9751	41	25	X	X		X	X	X
27	neuropathy	5	0.9747	3	1	X		X	X	X	X
28	use	5	0.9746	66	145	X	X		X	X	X
29	trial	5	0.9745	74	16	X	X		X	X	X
30	patient	5	0.9740	180	291	X	X		X	X	X
31	heparin	5	0.9739	4	13	X	X		X	X	X
32	cerebral	5	0.9739	79	25	X	X		X	X	X
33	diagnosis	5	0.9736	22	13	X	X	X	X		X
34	spread [2]	5	0.9736	9	2		X	X	X	X	X
35	case	5	0.9736	80	55	X	X		X	X	X
36	spread depression [3]	5	0.9735	5	2		X	X	X	X	X
37	hormonal	5	0.9735	10	13	X	X	X	X		X

Fig. 6. Candidate b-term ranking page as displayed by CrossBee after the preprocessing is done. This example shows CrossBee's term list output as ranked by the ensemble heuristic in the migraine-magnesium dataset. The terms marked yellow are the Swanson's b-terms.

b-terms, and others. When the user selects all the desired options he proceeds to the next step.

3. CrossBee starts a computationally very intensive step in which it prepares all the data needed for the fast subsequent exploration phase. During this step the actual text preprocessing, base heuristics, ensemble, bisociation scores and rankings are computed in the way presented in Section 3. This step does not require any user's intervention.
4. After computation, the user is presented with a ranked list of b-term candidates as seen in Fig. 6. The list provides the user some additional information including the ensemble's individual base heuristics votes (columns 7-12) and term's domain occurrence statistics in both domains (columns 5 and 6). If the user defines the actual b-terms during the initialization (which is not a realistic scenario when exploring a new domain for the first time) then these b-terms are marked throughout the whole CrossBee session, as seen in Fig. 6 and Fig. 7. The user then browses through the list and chooses the term he believes to be promising for finding meaningful connections between domains.

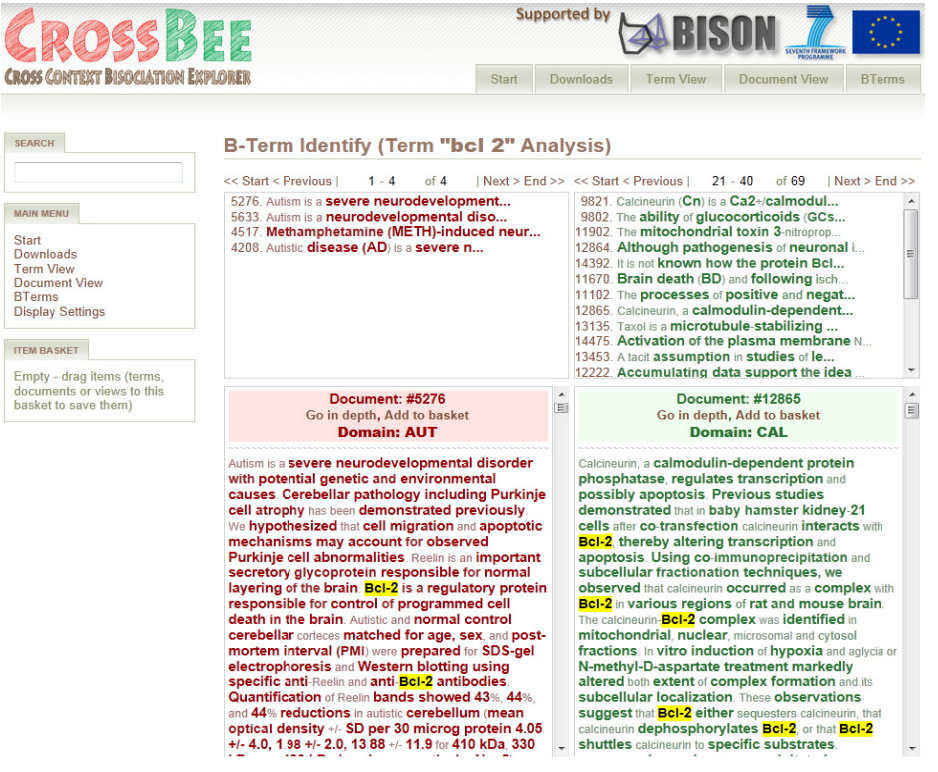


Fig. 7. CrossBee supports the inspection of potential cross-domain links by a side-by-side view of documents from the two domains under investigation. The figure presents an example from the autism-calcineurin dataset, showing the analysis of the *bcl 2* term. The presented view enables efficient comparison of documents, the left one from the autism and the right one from the calcineurin domain. The actual displayed documents were reported by Macedoni-Lukšič et al. [10] as relevant for exploring the relationship between autism and calcineurin.

5. At this point, the user inspects the actual appearances of the selected term in both domains, using the side-by-side document inspection as shown in Fig. 7. In this way, he can verify whether his rationale behind selecting this term as a bridging term can be justified based on the contents of the inspected documents.
6. Afterwards, the user continues with the exploration by returning to step 3 or by choosing another term in step 4, or concludes the session.

The most important result of the exploration procedure is a proof for a chosen term to be an actual bridge between the two domains, based on supporting facts from the documents. As experienced in sessions with the experts, the identified documents are an important result as well, as they usually turn out to be a valuable source of information providing a deeper insight into the discovered cross-domain relations.

5.2 Other CrossBee Functionalities

Below we list the most important additional functionalities of the CrossBee system:

- *Document focused exploration* empowers the user to filter and order the documents by various criteria. The user can find it more pleasing to start exploring the domains by reading documents and not browsing through the term lists. The ensemble ranking can be used to propose the user which documents to read by suggesting those with the highest proportion of highly ranked terms.
- *Detailed document view* provides a more detailed presentation of a single document including various term statistics and a similarity graph showing the similarity between this document and other documents from the dataset.
- *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 making them easier to spot (note different font sizes in Fig. 7).
- *b-term emphasis* marks the terms defined as b-terms by the user (note yellow terms in Fig. 7).
- *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 (note different colors in Fig. 7).
- *UI customization* enables the user to decrease or increase the intensity of the following features: high-ranked term emphasis, b-term emphasis and domain separation. In cooperation with the experts, we discovered that some of them do like the emphasizing features while the others do not. Therefore, we introduced the UI customization where everybody can set the intensity of these features by their preferences.

6 Discussion and Further Work

This work presents a methodology and a system for bisociative literature mining focusing on b-term identification and ranking by using an ensemble heuristic. First, a

detailed description of the proposed methodology and its experimental evaluation are provided, followed by the overview of the implemented system CrossBee.

In the experimental evaluation we tested a set of base heuristics and the proposed ensemble heuristic on two datasets: migraine-magnesium and autism-calcineurin. While the first dataset was used to develop the b-term ranking methodology, the second dataset was used as an independent test to validate the findings.

The comparison of the results on both datasets has shown that the performances of individual heuristics vary substantially. This indicates that there are differences between datasets which influence the performance of individual heuristics; while some base heuristic can be more adapted to one dataset, the others might be better suited to another. The proposed ensemble heuristic, which is among the best performing heuristics in both datasets, is therefore suggested as a dataset independent methodology.

The results of the heuristics were evaluated from two perspectives: (a) using the AUC measure, and (b) by counting the number of b-terms found in the first n term candidates. While the first measure (a) is used to estimate the quality of heuristics as a single number, which is good for ranking the heuristics, the second measure (b) is used to illustrate the heuristics quality from the end-user's perspective. In a typical scenario, the end-user appreciates reducing the burden of exploration by browsing through as few b-term candidates as possible to find the b-terms bridging the two domains. The comparison of baseline heuristics results with the constructed ensemble heuristic results confirms that the proposed methodology substantially reduces the end-user burden in this respect.

The CrossBee System has proved to be a user-friendly implementation of the presented methodology. Its visualization functionalities, in particular its presentation of pairs of documents which can be inspected in more detail for meaningful relations, is very helpful. An obvious extension planned for the near future is automatic download of documents from a selected bibliographic database, such as MEDLINE.

Investigation of more general connections between properties of domains and the best choice of selected heuristics combined into ensemble heuristic remains an important issue for further work, together with a more systematic study and comparison with other ensemble approaches known from the literature.

Acknowledgements. The work presented in this chapter was supported by the European Commission under the 7th Framework Programme FP7-ICT-2007-C FET-Open project BISON-211898, and the Slovenian Research Agency grant Knowledge Technologies (P2-0103).

Open Access. This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

1. Albert, R., Barabasi, A.L.: Statistical mechanics of complex networks. Rev. Mod. Phys. 74(1), 47–97 (2002)
2. 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)

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)
4. Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank Aggregation Methods for the Web. In: Proc. of the 10th int. Conference on World Wide Web, pp. 613–622 (2001)
5. Feldman, R., Sanger, J.: The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press (2007)
6. Hoi, S.C.H., Jin, R.: Semi-Supervised Ensemble Ranking. In: Proc. of the 23rd National Conference on Artificial Intelligence, vol. 2. AAAI Press (2008)
7. Juršič, M., Sluban, B., Cestnik, B., Grčar, M., Lavrač, N.: Bridging Concept Identification for Constructing Information Networks from Text Documents. In: Berthold, M.R. (ed.) Bisociative Knowledge Discovery. LNCS (LNAI), vol. 7250, pp. 66–90. Springer, Heidelberg (2012)
8. Koestler, A.: The Act of Creation. The Macmillan Co. (1964)
9. Li, D., Wang, Y., Ni, W., Huang, Y., Xie, M.: An Ensemble Approach to Learning to Rank. In: 5th Int. Conf. on Fuzzy Systems and Knowledge Discovery, pp. 101–105 (2008)
10. Macedoni Lukšič, M., Petrič, I., Cestnik, B., Urbančič, T.: Developing a Deeper Understanding of Autism: Connecting Knowledge through Literature Mining. In: Autism Research and Treatment (2011)
11. Petric, I., Urbancic, T., Cestnik, B., Macedoni-Luksic, M.: Literature mining method RaJoLink for uncovering relations between biomedical concepts. Journal of Biomedical Informatics 42(2), 219–227 (2009)
12. Petrič, I., Cestnik, B., Lavrač, N., Urbančič, T.: Outlier Detection in Cross-Context Link Discovery for Creative Literature Mining. Comput. J., November 2 (2010)
13. Petrič, I., Cestnik, B., Lavrač, N., Urbančič, T.: Bisociative Knowledge Discovery by Literature Outlier Detection. In: Berthold, M.R. (ed.) Bisociative Knowledge Discovery. LNCS (LNAI), vol. 7250, pp. 313–324. Springer, Heidelberg (2012)
14. Rokach, L.: Ensemble-based classifiers. Art. Int. Review 33(1-2), 1–39 (2010)
15. 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)
16. Swanson, D.R.: Migraine and magnesium: Eleven neglected connections. Perspectives in Biology and Medicine 31(4), 526–557 (1988)
17. Swanson, D.R.: Medical literature as a potential source of new knowledge. Bull. Med. Libr. Assoc. 78(1), 29–37 (1990)
18. Swanson, D.R., Smalheiser, N.R., Torvik, V.I.: Ranking Indirect Connections in Literature-Based Discovery: The Role of Medical Subject Headings (MeSH). Journal of the American Society for Inf. Science and Technology 57, 1427–1439 (2006)
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)
20. Urbančič, T., Petrič, I., Cestnik, B., Macedoni Lukšič, M.: RaJoLink: A Method for Finding Seeds of Future Discoveries in Nowadays. In: Proceedings of the 18th Symposium on Methodologies for Intelligent Systems, Prague, pp. 129–138 (2009)
21. Weeber, M., Vos, R., Klein, H., de Jong-van den Berg, L.T.W.: Using concepts in literature-based discovery: Simulating Swanson's Raynaud–fish oil and migraine-magnesium discoveries. J. Am. Soc. Inf. Sci. Tech. 52(7), 548–557 (2001)