A Hybrid Intelligent Approach for the Support of Higher Education Students in Literature Discovery

Ryan Prater, Emanuele Laurenzi

School of Business FHNW University of Applied Sciences and Arts Northwestern Switzerland, CH-4600 Olten, Switzerland

Abstract

In this paper, we present a hybrid intelligent approach that combines knowledge engineering, machine learning, and human intervention to automatically recommend literature resources relevant for a high quality of literature discovery. The primary target group that we aim to support is higher education students in their first experiences with research works. The approach builds a knowledge graph by leveraging a logistic regression algorithm which is first parameterized and then influenced by the interventions of a supervisor and a student, respectively. Both interventions allow continuous learning based on both the supervisor's preferences (e.g. high score of H-index) and the student's feedback to the resulting literature resources. The creation of the hybrid intelligent approach followed the Design-Science Research methodology and is instantiated in a working prototype named PaperZen. The evaluation was conducted in two complementary ways: (1) by showing how the design requirements manifest in the prototype, and (2) with an illustrative scenario in which a corpus of a research project was taken as a source of truth. A small subset from the corpus was entered into the PaperZen and Google Scholar, independently. The resulting literature resources were compared with the corpus of a research project and show that PaperZen outperforms Google Scholar.

Keywords

literature discovery, hybrid intelligence, knowledge graph, logistic regression

1. Introduction

The review of literature in research works like master's theses serve as the foundation upon which that research is built [1]. The presence of a research-worthy problem, for example, is established through the literature [2]. The discovery of content covering existing research, theories and evidence (i.e., literature discovery) is prerequisite for a literature review and creates the conditions for critically evaluating and discussing this content [2].

Conducting literature discovery is an intense learning process for higher education students, especially for those novel to research. The initial guidance of a supervisor is gradually weaned over time by the increasingly acquired self-regulated learning ability of the student, until the discovery can be conducted autonomously. Self-regulated learning refers to students' cognitive

In A. Martin, K. Hinkelmann, H.-G. Fill, A. Gerber, D. Lenat, R. Stolle, F. van Harmelen (Eds.), Proceedings of the AAAI 2022 Spring Symposium on Machine Learning and Knowledge Engineering for Hybrid Intelligence (AAAI-MAKE 2022), Stanford University, Palo Alto, California, USA, March 21–23, 2022.

ryan.prater@students.fhnw.ch (R. Prater); emanuele.laurenzi@fhnw.ch (E. Laurenzi)
0000-0003-3698-0111 (R. Prater); 0000-0001-9142-7488 (E. Laurenzi)

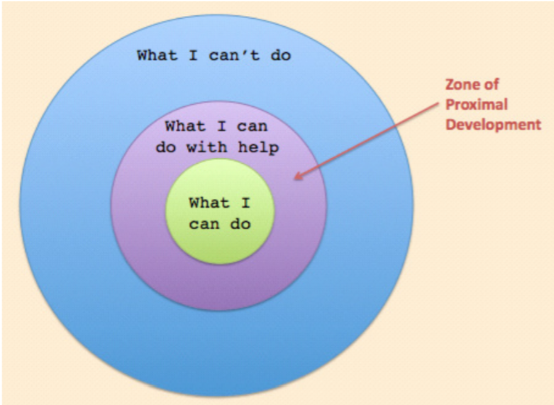

Figure 1: Zone of Proximal Development (ZPD) [6]

and motivational strategies for learning [3], and shall be supported for the benefit of the student's learning process, e.g., see [4].

Poor support on student's literature discovery might bring frustration such as sense of lost time or inability to make progress. This is problematic as it can lead to serious consequences such as the student abandoning the thesis topic or dropping the module [5].

In response to this problem, we present a hybrid intelligent approach that aims to foster self-regulated learning in literature discovery for research works like, but not limited to, master's theses. To accomplish this, the approach implements the Zone of Proximal Development (ZPD) [6]. That is, a student in the ZPD of a particular task can accomplish it if appropriate assistance is provided (see the middle layer in Figure 1).

In our approach, the assistance is provided by incorporating:

1. the supervisor's preferences about paper discovery, e.g., a high H-index of authors or Journal paper over conference papers. This refers to the "what I can't do" layer for a student (see Fig. 1);
2. the student's feedback over the discovered research resources. This refers to the "what I can do" layer for a student (see Fig. 1);
3. the automatic suggestion of new resources, based on the combination of both (1) and (2). This refers to the "what I can do with help" layer for a student (see Fig. 1).

While the supervisors' preferences can be engineered, discovery of research resources is tacit knowledge which we aim to learn with the help of the student's feedback. Therefore, a hybrid intelligent approach is implemented that combines Knowledge Engineering and Machine Learning approaches with human interventions.

This paper is structured as follows. First, in Section 2 background concepts are explored, including a brief overview of learning theory, imparting of knowledge, as well as related work and design patterns for hybrid intelligent approaches. The methodology of the approach is then explained in Section 3. The approach is then described in Section 4.2 in the form of a boxology. The approach is instantiated into a working prototype as illustrated in Section 4.3. Section

5 discusses how the value delivered to students and supervisors was measured by existence-matching through comparison of the corpus of a research project. Finally, a conclusion and suggestions for future research are made in Section 6.

2. Background and Related Work

This section introduces the theoretical background for learning and imparting of knowledge within the learning context. Next, approaches for literature discovery are discussed. Finally, design patterns for hybrid intelligence are introduced.

2.1. Learning Theory and Imparting of Knowledge

Learning theory strives to explain how students receive, process and retain knowledge during learning [7]. Prior knowledge or experiences play a fundamental part in how understanding is acquired or changed and knowledge and skills retained.

Vygotsky [6] introduced the notion of a Zone of Proximal Development (ZPD) as "the distance between the actual developmental level as determined by independent problem-solving and the level of potential development as determined through problem-solving under adult guidance or in collaboration with more capable peers". According to Vygotsky, when a learner is in the ZPD for a particular task, she/he can achieve it if appropriate assistance is provided.

Similarly, connectivism [8] postulates that learning occurs when connections are made between nodes in a learner's network - where a node can be anything from knowledge in the learner's mind, to a digital artifact, or another person. This implies that new knowledge must be connected to existing knowledge or experiences which can be understood as a concretization of the ZPD and that such connection can be mediated by human interaction in the digital environment.

2.2. Approaches for Literature Discovery

This section elaborates on existing approaches for literature discovery which follow a subset of the system outlines and characteristics of existing literature discovery systems by Beel et al. [9]. The relevant systems are categorized in approaches that first rely on knowledge graphs, then machine learning, and finally their hybrid combination.

2.2.1. Knowledge Graphs in Literature Discovery

Knowledge graphs (KG) use labeled schemas to enable semantical reasoning of their contained data [10, 11], e.g., searching papers which are cited from a specific author. Knowledge Graphs are often stored in graphs databases, which are designed to facilitate knowledge retrieval and logical inferencing. Elsevier[1] and SemanticScholar[2] are examples of scholarly knowledge graphs which aim to structure properties and information of resources related to literature such as publication date, journal ISSN, and publishing authors.

[1]https://www.elsevier.com/connect/how-ai-and-knowledge-graphs-can-make-your-research-easier
[2]https://www.semanticscholar.org/about

Knowledge graphs are also used in scholarly literature for approaches to literature discovery. Ammar et al. [12] described the construction of SemanticScholar through the use of labeled nodes and edges, and how they can be used for discovery. Liang et al. [13] constructed a citation graph with a citation link schema, e.g., *Comparable*, *Based-On*, and *General*, and developed literature recommendations based on schema weights and data distance. Auer et al. [14] demonstrated the human-machine readability of knowledge graph literature discovery via three use-cases: computer science, COVID-19, and materials science, and highlight the importance of semantic structuring.

Pure knowledge graph-based approaches however, require contextual experience for semantical understanding which novice researchers may lack. An approach which learns student behaviors via machine learning may provide a better medium for self-regulated learning.

2.2.2. Machine Learning in Literature Discovery

Some literature discovery systems leverage machine learning to make predictions based on scoring of literature properties without the use of knowledge graphs. The LitSuggest system [15] used a logistic regression classifier to analyze a flat-list of PubMed articles based on various properties, e.g., text analysis of abstracts, and achieved competitive results compared to similar systems. CiteSeer [16], one of the earliest literature recommendation systems, parsed related citations into tabular format, evaluated various methods of intelligent string comparison, and noted difficulties with "identification and disambiguation of authors".

A drawback of pure machine learning approaches is the lack of contextual knowledge and generalizability, which hinder the identification of resources that are relevant to a research problem, especially those targeting different application domains. A hybrid combination of knowledge engineering and machine learning can be used to overcome the drawback of both approaches.

2.2.3. Hybrid Approaches in Literature Discovery

Literature recommendation systems exist which use both knowledge graphs and machine learning. For example, Middleton et al. [17] used a multi-class nearest-neighbor algorithm and compared a flat-list data representation against an instantiated is-a ontology of hierarchical literature topics and found the ontology approach resulted in improved accuracy and "rounder profiles".

Such multi-faceted systems may also include a human component which can satisfy or supplement the recommendation task. Beel et al. [9] explored studies using *collaborative filtering*, which leverage social systems for improved performance. Pennock et al. [18] categorized users according to personality type and applied it to the CiteSeer [16] system. Tang and McCalla [19] applied pedagogical features, e.g., learner background knowledge and interest, to literature discovery and emphasized the their importance for expanding suggestions outside of localized results.

Although such hybrid approaches provide greater support in literature discovery, they lack human intervention, which could be helpful for the students to achieve an adequate Zone of Proximal Development.

By integrating student-supervisor interaction into a hybrid approach, we aim to support students who are novel to research to achieve an adequate Zone of Proximal Development. That is, literature resources are offered to students by taking into account both their feedback and supervisors' preferences. The proposed hybrid intelligent approach also contribute to address the research call made by Auer et al. [14], who asked "How can we increasingly involve specialist scientists in the curation process?".

2.3. Patterns for Hybrid Approaches

van Harmelen and ten Teije [20] proposed an abstraction mechanism for all hybrid approaches that combine learning and knowledge engineering which led to a set of reusable and compositional design patterns. Witschel et al. [21] builds on their work to abstract systems with a human in the loop. For example, the output of a machine learning may be in a human-interpretable form which could be used by a human expert to create or improve a knowledge representation, which could then be used for deductive reasoning. We leverage these research findings on design patterns for hybrid (intelligent) approaches to create our approach (see Section 4.2).

3. Methodology

The hybrid intelligent approach was developed by following Design Science Research [22] which is a problem-solving methodology via the process of awareness of problem, approach, development, evaluation, and conclusion. This methodology was used by other studies on literature discovery [23] and was selected because instantiation of an innovative artifact would enable evaluation of our approach to self-regulated learning of literature discovery.

In the *awareness of problem phase* (Section 4.1), semi-structured interviews were conducted to supplement literature findings on approaches for literature discovery. A set of requirements were then derived and addressed with the hybrid intelligent boxology approach by the designed artifact (Section 4.2 - *approach phase*). These requirements were incorporated in the approach instantiation, which comes in the form of a running prototype (Section 4.3 - *development phase*). In the *evaluation phase* (Section 5), the hybrid intelligent approach was evaluated through the prototype, first with respect to the requirements and then by performing test runs to evaluate on the breadth of the approach. The *conclusion phase* then takes places in Section 6.

4. The Hybrid Intelligent Approach

This section describes the proposed hybrid intelligent approach to support higher education students in literature discovery. An awareness of the problem is derived from stakeholder interviews which are translated into a set of requirements. These requirements are manifested in the components of the approach. The latter is then instantiated in the technical implementation called PaperZen.

4.1. Awareness of Problem

For an in-depth understanding of the problem addressed in this work, literature findings were supplemented with primary data from semi-structured interviews. The latter were conducted face-to-face physically or via calls, with duration of one hour each. The selection criteria (C1-4) for interviewees were as follows:

C1 a university research supervisor and reviewer with at least 5 years of experience in both supervising research works and reviewing literature discovery guided by other supervisors.

C2 a university research supervisor with at least 5 years of experience in supervising research works.

C3 a master student engaged in writing a master's thesis, thus being supervised.

C4 a master student in the last year of studies having at least one experience with a supervised research work but not yet started their theses.

Respectively, interviews were conducted with: (C1) one research supervisor and reviewer - a university professor with 10 years experience in both supervising and reviewing. (C2) one research supervisor - a university lecturer and researcher with 5 years experience in supervising, (C3) two master's students writing theses, (C4) and 17 masters students at their last year of studies with 1 to 3 research works under supervision. The higher number in category (C4) is justified by less experience in research works.

To elicit as much insights as possible, open-ended questions were asked, e.g.,

- As a supervisor to thesis writers, what pain points do you experience when guiding literature discovery? Please underpin the answers by referring to concrete cases from your past supervision experience (C1 and C2).
- As a master's student writing your thesis, how do you discern literature quality? Please underpin the answers by referring to concrete cases from your past experience (C3).
- As a master's student with previous experience in research works, what constraints or oversight does your supervisor have on your literature discovery? Please underpin the answers by referring to concrete cases from your past experience (C4).

Interviews results were analysed and the main findings (F1-4) were consolidated.

F1 : No clear guidance makes students insecure about their direction (C3 and C4).

F2 : There is no scientific method to convert research questions into literature searches (C1 and C2). This prompts supervisors to create varying and sometimes inconsistent constraints for their students (C4).

F3 : Students must review the results of their literature discovery, many of which might not be relevant to the topic (C3 and C4).

F4 : If a student loses direction, supervisors only have the opportunity for correction at the next review of the students work (C1 and C2).

F5 : From the perspective of (C1), neglect of proper supervision from (C2), e.g. of lack of time, impacts the quality of literature discovery from (C3).

Finally, findings were mapped into three main requirements (R1-3), which set the basis for designing the novel hybrid intelligent approach.

R1 : the approach shall enable the students to be guided when selecting literature as well as being empowered to decide on which resource to review (F1, F3).

R2 : the approach shall enable the supervisors to influence student literature discovery outside of the classroom (F4, F5).

R3 : the approach shall enable to adequately consider the knowledge of both students and supervisors in the literature discovery result (F2, F4).

4.2. The Suggested Boxology for the Hybrid Intelligent Approach

Our approach is illustrated in Figure 2 using the boxology approach proposed by van Harmelen and ten Teije [20] and later extended by Witschel et al. [21]. Human represents a human component, sym a model-based input or output, data a non-model-based input or output, ML an inductive machine learning inference, and KR a deductive knowledge representation inference. Numbers are for reference purposes.

Three boxology design patterns influenced this approach. The *Feedback-Based Learning Pattern* of Witschel et al. [21] was used to address R1 and manifested in boxes 1-6 to reuse a student's input (box 2) to incorporate human judgement into the the machine learning weights of box 3 which feeds back to box 1. These weights are then used in box 5 in combination with the supervisor scores (box 12) to influence the suggestions (box 6), which addresses R2 and follows the *Explainable learning systems with background knowledge* pattern of van Harmelen and ten Teije [20]. The combination of the student and supervisor scores (boxes 4, 12) to produce a combined output (box 6) follows the *Learning an intermediate abstraction for reasoning* pattern, which addresses R3.

4.3. Instantiation and Steps of the Hybrid Intelligent Approach

The hybrid intelligent approach was instantiated in a working prototype called PaperZen. Neo4j[3] was chosen as the knowledge graph database for storage. The literature resources were obtained via API of SemanticScholar. The *LogisticRegression* model provided by the Python library *scikit-learn* with default parameters was used to implement machine learning with logistic regression (LR).

The instantiation consists of several steps and components which were derived from the boxology pattern of Figure 2. A student (box 1) begins their discovery journey by adding literature papers to their library (*student corpus*, box 2) which are saved to a graph database. During or before this time, a supervisor (box 9) sets their search preferences for the student by adjusting multiple feature weights (box 10). The user then initiates a search. The approach gathers (box 7) a number of related papers or authors (*Potentials*, box 8) which are trained on a logistic regression model (box 3) of previously saved or rejected search suggestions from the student (box 8). The Potentials are then evaluated against the trained model and a vector of probabilities for the saved class are outputted (vector-y-student, box 4). The normalized

[3]https://neo4j.com/

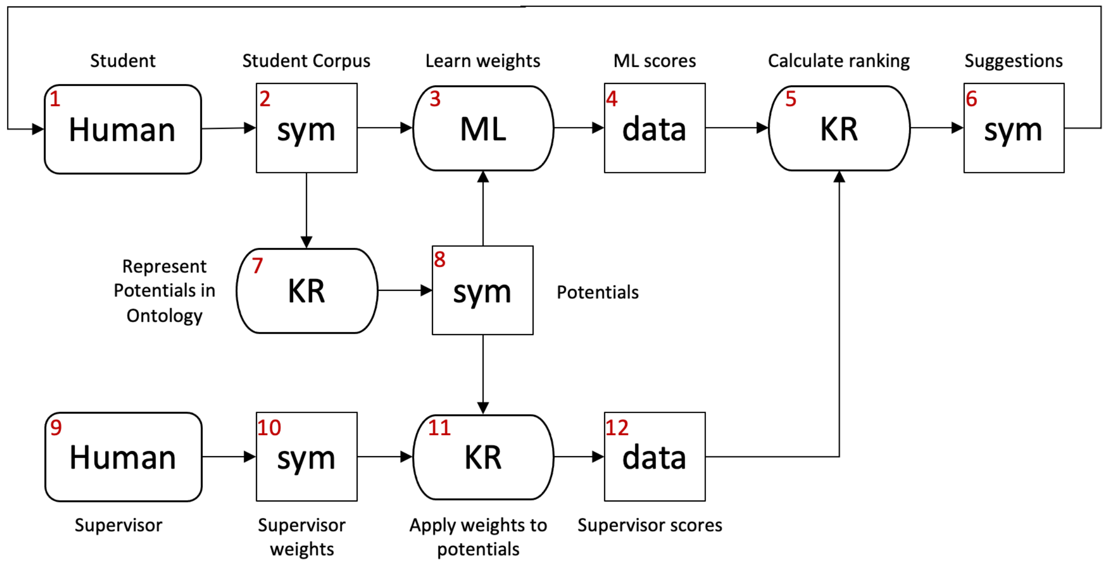

Figure 2: Boxology of the hybrid intelligent approach

feature columns of the test set are multiplied by the supervisor preferences (box 11) and each row is then mean-averaged into a vector of probabilities (`vector-y-supervisor`, box 12). `vector-y-student` and `vector-y-supervisor` are then mean-averaged together (box 5) to produce the final per-Potential prediction `vector-y-prediction` (box 6) which is then ordered by prediction value and presented in groups of three to the student (back to box 1). The student can choose to save or reject each suggestion individually (box 2). Based on these decisions, the LR model (box 3) is further trained and the process begins again.

Boxes 7, 11, 3, 5, and 1 are further detailed below.

4.3.1. Knowledge Graph Component

To seed (start) their corpus, a student may search for and add a paper by DOI, paper title, or BibTeX upload. To represent the various literature resource types while supporting efficient storage and effective human and machine readability, the following labelling schema was used:

- `Corpus - Saved`: Nodes which have been saved to the student's corpus.
- `Corpus - Rejected`: Nodes which have been rejected by the student during search.
- `Resource - Paper`: Nodes which are papers, e.g., journal article, conference paper, book, etc. This is predetermined by SemanticScholar.
- `Resource - Author`: Nodes which are authors of papers.
- `Resource - Citation`: Nodes (implicitly papers) listed as a citations to papers.
- `Resource - Reference`: Nodes (implicitly papers) listed as a references from papers.

The labels are added at different stages of student interaction, and multiple labels may be applied to a single node, e.g., a student may be suggested a `Resource - Paper` which is both a

Resource - Reference and a Resource - Citation, and upon saving the suggestion the node receives the Corpus - Saved label. Currently, rejected suggestions are never presented again unless the student manually adds them to their corpus.

Upon addition of resources to the student's library, PaperZen preemptively discovers and hydrates (represents) Potential resources (box 7). To ensure a distributed approach, each node saved in the student's corpus is iterated following the process of Listing 1. A configuration number, NUM_LOOKUP_PER_NODE, controls how many Potentials of each resource can be extracted per node, and another, NUM_TARGET_PER_RESOURCE, defines the target amount of Potentials per resource type. To prevent redundancy, no lookup is done if a node has already been hydrated by the API.

Listing 1: Pseudocode of aggregating Potentials

```
def aggregate_potentials(students_saved_corpus):
    NUM_TARGET_PER_RESOURCE = 100
    NUM_LOOKUP_PER_NODE = 3
    for node in students_saved_corpus:
        for label in ['author', 'citation', 'reference']:
            existing_potentials = get_hydrated_nodes(corpus, label
                )
            potentials = get_connections_to_node(node)
            new_potentials = existing_potentials - potentials
            new_nodes = []
            for potential in new_potentials:
                if len(existing_potentials) + len(new_nodes) >
                    NUM_LOOKUP_PER_NODE:
                     break
                new_node = hydrate_and_store_node(potential)
                new_nodes += new_node
            if (len(exiting_potentials) + len(new_nodes)) >
                NUM_TARGET_PER_RESOURCE:
                break
```

4.3.2. Human Preference Component

Box 11 symbolizes the encoding of Potentials' various resource characteristics as features to be preferenced by the supervisor. To do this, the following properties are used as feature weights:

- L: Labels indicating resource type. This is one-hot-encoded into L-A (Resource - Author), L-C (Resource - Citation), and L-R (Resource - Reference).
- HI: H-Index of the author
- CC: Citation count of the paper
- RC: Reference count of the paper
- PY: Publication year of the paper
- RI: Number of relationships *inbound* to a node
- RO: Number of relationships *outbound* from a node

These weights are available to supervisors via a user interface which allows customization of preferences (*supervisor weights*) for the prevalence of such features in search suggestions and addresses `R2` (See Figure 3 (2)). For example, by setting the weight for `L-A` to 100 and the weight of `L-C` and `L-R` both to zero, the supervisor effectively influences the search to only return authors. These weights are also used as features to the LR model (box 3). This also addresses `R1` by quantifying various features of the literature discovery to learn the search preferences of students.

4.3.3. Machine Learning Component

Upon initialization of PaperZen, the LR (box 3) training dataset is empty, so results are compensated by the supervisor weights which are defaulted to `0.5`. If not empty, existing decisions (saved or rejected) of a student are used for training the LR model. `HI`, `CC`, `RC`, and `PY` are commonly used in literature recommendation systems [16]. These values are set to zero when not applicable to certain resource types, e.g., a paper will not have an H-Index, but publication year is set to the mean of the publication years.

To ensure balanced weighting across features, each feature is individually normalized using a min-max scalar with values in the range of [0,1]. The resulting training set matrix is referred to as `matrix_X_train_student`. The Potentials are then prepared and normalized using the same process for training data to produce `matrix_X_test_student`. The output of evaluation is a vector of prediction probabilities (`vector_y_student`) for `Corpus - Saved` with values in the range of [0,1] which represents the vectorization of `R1`.

4.3.4. Recommendation Component

Box 5 represents the combination of LR and supervisor weights to form literature resource suggestions. Following the LR of the Potentials and the output of the probabilities vector, the loaded model coefficients for each feature are extracted. This is a vector which represent the LR weights the trained model assigns to each feature in order to calculate the prediction for each sample. The supervisor weights are then matrix-multiplied across `matrix_X_test_student` to produce a matrix of human-influenced features `matrix_X_test_supervisor` and is then mean-averaged row-wise to produce the prediction vector `vector_y_supervisor` which represents the supervisor preference of each resource. `vector_y_student` and `vector_y_supervisor` are then concatenated and mean-averaged row-wise to produce `vector_y_prediction` which represents the hybrid student and supervisor preference, addressing `R2`.

4.3.5. Feedback Loop Component

The results of `matrix_y_hybrid` are index-matched and sliced from `matrix_X_train_supervisor`, ordered by descending value, and returned to the calling search function. The result is a list of Potentials which are ordered by hybrid student-supervisor probability of the student saving the resource. This list is then displayed to the student (box 1) in batches of three.

When presented the suggestions, the student may choose to save or reject each suggestion individually. This represents the guided choice of `R1`. If rejected, the resource (node) is labeled with `Corpus - Rejected` and the resource is never again suggested to the student. If saved, the label `Corpus - Saved` is applied to the resource, and the resource will be added to the student's corpus. These resources are later included in the next search as training data, which means the LR model dataset grows proportional to use and `R3` is iteratively addressed.

5. Evaluation

The hybrid intelligent approach was evaluated in two manners: (1) by showing how the requirements manifest in the technical prototype and (2) by proving the utility of the proposed approach through tests. The former is described in Section 5.1 and the latter in Section 5.2.

5.1. Requirements Traceability

Figure 3 depicts the three main user interfaces of PaperZen with numbers for reference. The student may upload resources which are added to their library (1). The supervisor may then adjust their preference weights (2) which influence the suggestions in the student's search (3). The three main requirements for the approach manifest as follows:

- `R1` : the approach shall enable the students to be guided when selecting literature as well as being empowered to decide on which resource to review. Fig. 3 (3) shows how PaperZen suggests resources for review to the student.
- `R2` : the approach shall enable the supervisors to influence student literature discovery outside of the classroom. Fig. 3 (2) shows how PaperZen allows the supervisor to adapt their preference weights.
- `R3` : the approach shall enable to adequately consider the knowledge of both students and supervisors in the literature discovery result. This requirement manifests in terms of displayed resources (see Fig. 3 (1)) after both the supervisor and the student provide the preferences for and the feedback to the resources, respectively.

5.2. Utility of the PaperZen Approach

This section describes how the approach was evaluated with respect to its utility. This refers to the ability of the PaperZen approach to return satisfying results. For this, the research literature of the Swiss research project "Digital Self-Study Assistant" (DSSA) [24] was used as a ground truth dataset (*truth corpus*). Hence, five test runs were performed covering various configurations to explore the breadth of PaperZen's features. The results were then compared to Google Scholar, to which the title of the paper(s) in each of the test runs were entered in the search bar. Details are reported in the below sub-sections 5.2.1 to 5.2.3. Finally, a discussion about the evaluation results is provided in sub-section 5.2.4.

5.2.1. The Truth Corpus

The truth corpus[4] contained 43 items, one of which was a website, therefore the sample size n was 42. The 42 articles were searched in SemanticScholar and four articles could not be found,

[4] https://github.com/manulaur/TruthCorpus4Evaluation/commit/e440f42128caed9151968a82496a8c1e49c1496c

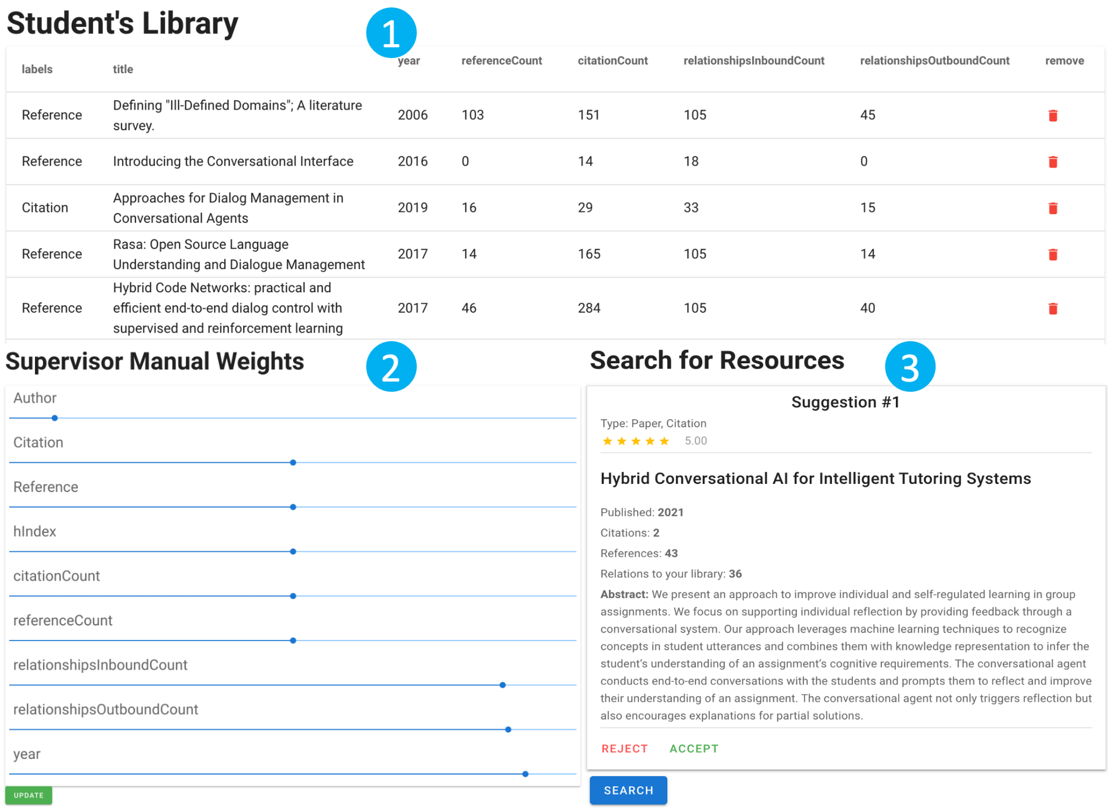

Figure 3: Screenshots of the PaperZen prototype showing (1) the student's library (corpus), (2) the supervisor's assignment of preference weights, and (3) search suggestions for the research corpus of the DSSA research project [24]

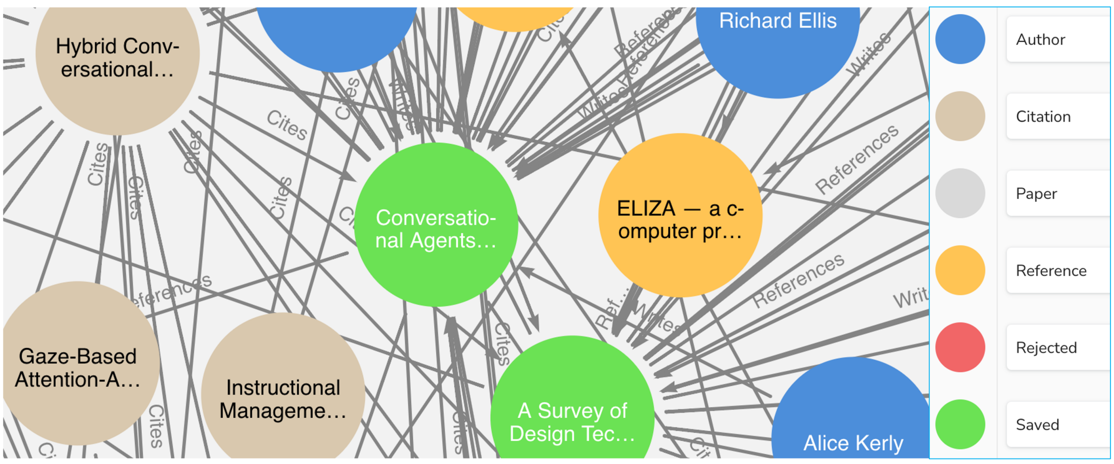

Figure 4: A subset of PaperZen's knowledge graph for the corpus of the DSSA research project [24]

meaning the best accuracy possible for PaperZen would be 90.48%. Additional characteristics of the truth corpus were: 22 of the literature were journal articles, 12 were conference proceedings, and eight were books or book excerpts. The mean publication year of the literature was 2011 and the median was 2014. The total amount of authors (separated from their papers) was 107. Figure 4 depicts an excerpt of the knowledge graph from the corpus of the DSSA research project [24].

Five evaluations were run on the truth corpus:

- `Eval0`: One random article from the truth corpus was selected, LR weights were set at default, supervisor weights were set at default
- `Eval1`: The same article from `Eval0` was selected, LR weights were reused from `Eval0`, supervisor weights were set at default, except for `Author`=1.0
- `Eval2`: Three random articles from the truth corpus were selected, LR weights were set at default, supervisor weights were set at default
- `Eval3`: The same articles from `Eval2` were selected, LR weights were reused from `Eval2`, supervisor weights were set at default, except for `Publication Year`=1.0
- `Eval4`: Nine random articles from the truth corpus were selected, LR weights were set at default, supervisor weights were set at default, except for `Author`=0.0

Given the truth corpus of size `n` and because PaperZen provides recommendations in groups of 3, `n/3` iterations of PaperZen search took place for each evaluation. Accuracy was measured (`true positives/n`), which is a common evaluation method of recommendation systems [9].

The seed article(s) for matching the truth corpus were chosen by random integer between 0 and `n` and used to index-select the truth corpus literature sorted by ascending publication year. When indicated, the LR weights were reset to default, meaning the training began with the seed article(s). When indicated, the supervisor weights were reset to default values of 0.5 per feature. After every evaluation, the database was reset, allowing suggestions to reappear in subsequent tests.

5.2.2. Evaluation Results

The results of the five evaluations are show in Table 1 and the resulting LR weights in Table 2.

In `Eval0`, three suggestions matched the truth corpus, with mean and median publication years of 2011 and 2014 respectively. From 42 suggestions, two were duplicates, but were not matches to the truth corpus. All three matches were literature articles with publication years of 2018, 1999, and 2019 and appeared in the 7th, 9th, and 14th search iterations respectively.

In `Eval1`, the LR weights were kept from `Eval0` and `L-A` was increased to 1 to test the effect on resource types returned. There were six matches to the truth corpus - two literature articles and four authors. The literature article publication years were both 2017 and appeared in the 4th and 5th search iterations respectively.

`Eval2` evaluated the effect of using multiple seed articles. The three articles did not include the article used in `Eval0` or `Eval1`. The LR weights and supervisor weights were reset to default (0.5). Out of 42 suggestions, there were zero matches. Two suggestions were duplicates. Both the mean and median publication years for `Eval2` were 1993.

Table 1
Results of Supervisor Truth Corpus Similarity

Evaluation	Matches	% Accuracy	% Authors	Mean Pub Year	Median Pub Year
Eval0	3	07.07	0	2010	2010
Eval1	6	14.29	67	2016	2020
Eval2	0	00.00	0	1993	1993
Eval3	9	21.43	100	2021	2019
Eval4	0	00.00	0	2015	2018

Table 2
Resulting Logistic Regression Weights

Evaluation	LA	LC	LR	H	CC	RC	RI	RO	PY
Eval0	0	0	0	0	0	0	0.72	0.71	0.14
Eval1	1	0	0	0.64	0.04	0	0.59	0.60	0
Eval2	0	0	0	0	0	0.03	0.76	0.96	0.29
Eval3	1	0	0	0.89	0	0	0.59	0.75	0
Eval4	0	0	0	0	0	0	1	1	0

`Eval3` used the same seed articles as `Eval2`, but set `PY` to 1. The LR weights and supervisor weights were reset to default. 33 literature articles were returned (9 authors) with mean and median publication years of 2021 and 2019 respectively. There were nine matches, all authors of the seed papers. The author matches were spread among the 5th-12th search iterations.

`Eval4` used nine seed articles to evaluate the effect of using 20% of the truth corpus for seeding. The LR and supervisor weights were reset to default, except for `L-A` set to 0 to avoid author-resource-centralization. There were zero matches to the truth corpus.

5.2.3. Comparison to Google Scholar

For a comparison to Google Scholar, the titles of the selected articles of the evaluations were concatenated together into the search of Google Scholar, and the top `n` results were recorded, with `n` being equal to the number of articles in the truth corpus. This comparison was done for `Eval0`, `Eval2`, and `Eval4` because `Eval2` and `Eval3` reused the articles from `Eval0` and `Eval2` respectively.

For `Eval0`, Google Scholar returned 29,300 results in 0.05 seconds, of which the first 42 articles were compared to the truth corpus. One article which was present in the truth corpus, resulting in an accuracy of 0.0238. For `Eval2`, Google Scholar returned 25 results in 0.06 seconds, of which all results were compared to the truth corpus. Zero articles were present in the truth corpus, resulting in an accuracy of 0. For `Eval4`, Google Scholar returned 0 results. This was most likely due to the concatenation of the nine article titles.

5.2.4. Discussion on Evaluation Results

The evaluation of matching accuracy was generally low, with a 7% literature match in `Eval0`, 5% literature match in `Eval1`, and 0% in other evaluations. However, PaperZen outperformed

Google Scholar which matched 2% of the literature in Eval0 and 0% in other evaluations. Further, many of the returned articles showed characteristics applicable to the research topic of DSSA (e.g., titles such as "Simulating Instructional Roles through Pedagogical Agents") but were not in the truth corpus, perhaps indicating a small sample size. Additionally, the first suggestions of Eval4 were unrelated in title to DSSA, but had a large number of citations, possibly indicating min-max stratification of the CC weight and a local search maximum.

The feature weights in Table 2 appeared to be susceptible to clustering, such as RI and RO. Given the low matching accuracy, we are hesitant to conclude that RI and RO are actually as important as indicated by the data. However, Eval1 confirmed the ability of L-A to influence suggested resource types and Eval3 confirmed the ability of PY to influence the publication year of the literature. This indicates feasibility of supervisor influence on student literature suggestions during discovery and addresses.

Four authors in Eval1 and nine in Eval3 were suggested, all authors of the seed literature. The relation of these resources to the seed literature indicates the connectivity of related resource types and verifies the integrity of the knowledge graph architecture.

6. Conclusion

The presented hybrid intelligent approach aims to support higher education students who are novel to research to achieve an adequate Zone of Proximal Development in literature discovery. That is, literature resources are offered to students by taking into account both their feedback and supervisors' preferences. Specifically, the suggestions derive from a combination of literature discovery techniques of knowledge graphs, machine learning, and collaborative filtering and allow the research-supervisor to influence what is presented to the student while the student's self-regulated learning is supported by learning their personal capabilities. The approach is instantiated in a working prototype called PaperZen, which incorporates techniques of knowledge engineering, machine learning, and human interventions. By matching output suggestions to a known research corpus, PaperZen showed matches could be made and displayed comparative and better performance to Google Scholar.

Limitations included the small sample size of the 42 resources in the single truth corpus, the suggestion of author resources without inclusion in the seed corpus, and the difference in search inputs for Google Scholar (keyword text input) compared to PaperZen (resource based). In the future, additional features will be explored over a greater evaluation sample to ensure reproducible results, text analysis will be included to reduce local search maximums, and both students and supervisors will be involved in the evaluation for a deeper analysis of learning.

References

[1] Y. Levy, T. Ellis, A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research, Informing Science: The International Journal of an Emerging Transdiscipline 9 (2006) 181–212. doi:10.28945/479.

[2] T. Ellis, Y. Levy, Framework of Problem-Based Research: A Guide for Novice Researchers on the Development of a Research-Worthy Problem, Informing Science: The International Journal of an Emerging Transdiscipline 11 (2008) 17–33. doi:10.28945/438.

[3] A. Friedrich, K. Jonkmann, B. Nagengast, B. Schmitz, U. Trautwein, Teachers' and students' perceptions of self-regulated learning and math competence: Differentiation and agreement, Learning and Individual Differences 27 (2013) 26–34. doi:10.1016/j.lindif.2013.06.005.

[4] M. English, A. Kitsantas, Supporting student self-regulated learning in problem-and project-based learning, Interdisciplinary Journal of Problem-based Learning 7 (2013) 128–150. doi:10.7771/1541-5015.1339.

[5] D. Hallberg, U. Olsson, Self-regulated learning in students' thesis writing, International Journal of Teaching and Education 1 (2017) 13–24. doi:10.20472/TE.2017.5.1.002.

[6] L. S. Vygotsky, Mind in Society: Development of Higher Psychological Processes, Harvard University Press, Cambridge, MA, 1978. doi:10.2307/j.ctvjf9vz4.

[7] O. J. Ellis, Human Learning, 6th ed., Pearson, 2012. URL: https://www.pearson.com/us/higher-education/product/Ormrod-Human-Learning-6th-Edition/9780132595186.html.

[8] G. Siemens, Connectivism: A learning theory for the digital age, International Journal of Instructional Technology and Distance Learning 2 (2004).

[9] J. Beel, B. Gipp, S. Langer, C. Breitinger, Research-paper recommender systems: A literature survey, International Journal on Digital Libraries 17 (2016) 305–338. doi:10.1007/s00799-015-0156-0.

[10] P. A. Bonatti, S. Decker, A. Polleres, V. Presutti, Knowledge Graphs: New Directions for Knowledge Representation on the Semantic Web (Dagstuhl Seminar 18371), Dagstuhl Reports 8 (2019) 29–111. doi:10.4230/DagRep.8.9.29.

[11] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge Graphs, ACM Computing Surveys 54 (2021) 1–37. doi:10.1145/3447772.

[12] W. Ammar, D. Groeneveld, C. Bhagavatula, I. Beltagy, M. Crawford, D. Downey, J. Dunkelberger, A. Elgohary, S. Feldman, V. Ha, R. Kinney, S. Kohlmeier, K. Lo, T. Murray, H.-H. Ooi, M. Peters, J. Power, S. Skjonsberg, L. Wang, C. Wilhelm, Z. Yuan, M. van Zuylen, O. Etzioni, Construction of the Literature Graph in Semantic Scholar, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers), Association for Computational Linguistics, New Orleans - Louisiana, 2018, pp. 84–91. doi:10.18653/v1/N18-3011.

[13] Y. Liang, Q. Li, T. Qian, Finding relevant papers based on citation relations, in: H. Wang, S. Li, S. Oyama, X. Hu, T. Qian (Eds.), Web-Age Information Management, volume 6897 LNCS, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 403–414. doi:10.1007/

978-3-642-23535-1_35.
[14] S. Auer, A. Oelen, M. Haris, M. Stocker, J. D'Souza, K. E. Farfar, L. Vogt, M. Prinz, V. Wiens, M. Y. Jaradeh, Improving Access to Scientific Literature with Knowledge Graphs, Bibliothek Forschung und Praxis 44 (2020) 516–529. doi:10.1515/bfp-2020-2042.
[15] A. Allot, K. Lee, Q. Chen, L. Luo, Z. lu, LitSuggest: A web-based system for literature recommendation and curation using machine learning, Nucleic Acids Research 49 (2021) W352–W358. doi:10.1093/nar/gkab326.
[16] C. L. Giles, K. D. Bollacker, S. Lawrence, Citeseer: An automatic citation indexing system, in: Proceedings of the Third ACM Conference on Digital Libraries, DL '98, Association for Computing Machinery, New York, NY, USA, 1998, pp. 89–98. doi:10.1145/276675.276685.
[17] S. E. Middleton, D. C. De Roure, N. R. Shadbolt, Capturing knowledge of user preferences: Ontologies in recommender systems, in: Proceedings of the 1st International Conference on Knowledge Capture, number January in K-CAP '01, Association for Computing Machinery, New York, NY, USA, 2001, p. 100–107. doi:10.1145/500737.500755.
[18] D. M. Pennock, E. Horvitz, S. Lawrence, C. L. Giles, Collaborative filtering by personality diagnosis: A hybrid memory and model-based approach, in: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, UAI '00, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000, p. 473–480. URL: https://dl.acm.org/doi/10.5555/647234.720062.
[19] T. Y. Tang, G. I. McCalla, The pedagogical value of papers: a collaborative-filtering based paper recommender, J. Digit. Inf. 10 (2009). URL: http://dblp.uni-trier.de/db/journals/jodi/jodi10.html#TangM09.
[20] F. van Harmelen, A. ten Teije, A boxology of design patterns for hybrid learning and reasoning systems, Journal of Web Engineering 18 (2019) 97–124. doi:10.13052/jwe1540-9589.18133, abstract in CEUR vol. 2491.
[21] H. F. Witschel, C. Pande, A. Martin, E. Laurenzi, K. Hinkelmann, Visualization of patterns for hybrid learning and reasoning with human involvement, in: R. Dornberger (Ed.), New Trends in Business Information Systems and Technology: Studies in Systems, Decision and Control, volume 294, Springer International Publishing, Cham, 2021, pp. 193–204. doi:10.1007/978-3-030-48332-6_13.
[22] V. K. Vaishnavi, Design Science Research Methods and Patterns: Innovating Information and Communication Technology, 1st ed., Auerbach Publications, New York, 2007. doi:10.1201/9781420059335.
[23] B. Sturm, A. Sunyaev, Design Principles for Systematic Search Systems: A Holistic Synthesis of a Rigorous Multi-cycle Design Science Research Journey, Business Information Systems Engineering 61 (2019) 91–111. doi:10.1007/s12599-018-0569-6.
[24] C. Pande, H. F. Witschel, A. Martin, D. Montecchiari, Hybrid conversational ai for intelligent tutoring systems, in: A. G. D. L. R. S. F. v. H. A. M. K. Hinkelmann, H.-G. Fill (Ed.), Proceedings of the AAAI 2021 Spring Symposium on Combining Machine Learning and Knowledge Engineering (AAAI-MAKE 2021), Stanford University, Palo Alto, California, USA, March 22-24, 2021, volume 2846 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021. URL: http://ceur-ws.org.