applied sciences

MDPI

Article

New Hybrid Techniques for Business Recommender Systems

Charuta Pande *, Hans Friedrich Witschel and Andreas Martin

Intelligent Information Systems Research Group, FHNW University of Applied Sciences and Arts Northwestern Switzerland, Riggenbachstrasse 16, 4600 Olten, Switzerland; hansfriedrich.witschel@fhnw.ch (H.F.W.); andreas.martin@fhnw.ch (A.M.)

* Correspondence: charuta.pande@fhnw.ch

Abstract: Besides the typical applications of recommender systems in B2C scenarios such as movie or shopping platforms, there is a rising interest in transforming the human-driven advice provided, e.g., in consultancy via the use of recommender systems. We explore the special characteristics of such knowledge-based B2B services and propose a process that allows incorporating recommender systems into them. We suggest and compare several recommender techniques that allow incorporating the necessary contextual knowledge (e.g., company demographics). These techniques are evaluated in isolation on a test set of business intelligence consultancy cases. We then identify the respective strengths of the different techniques and propose a new hybridisation strategy to combine these strengths. Our results show that the hybridisation leads to substantial performance improvement over the individual methods.

Keywords: recommender systems; case-based reasoning; hybrid recommenders

check for updates

Citation: Pande, C.; Witschel, H.F.; Martin, A. New Hybrid Techniques for Business Recommender Systems. *Appl. Sci.* **2022**, *12*, 4804. https://doi.org/10.3390/app12104804

Academic Editor: Luis Javier Garcia Villalba

Received: 28 March 2022
Accepted: 2 May 2022
Published: 10 May 2022

1. Introduction

Digitalisation leads to a transformation of internal business processes but also very notably of customer-facing services, while most attention is paid to services in the B2C domain, there is also a rising interest in digitalising knowledge-intensive services in the B2B domain, such as consultancy in general [1] and IT consultancy in particular [2]. Such transformation implies that a digital service (partially) takes over the role of a human consultant and that companies can use that service to help themselves to the required advice.

Obviously, such digital services will be able to give advice only for restricted domains; often, advice will consist of recommending items from a predefined set of solution components. Thus, digital consulting services can be thought of as recommender systems.

As we have laid out in our previous work [3], a recommender that suggests solution components to companies is different in several ways from the typical B2C recommenders that help users in finding, e.g., books, movies or music that fit their preferences (see also [4]):

- Requirement-driven: A consultancy recommender needs to consider business requirements not personal preferences;
- Interdependent items: The recommended items are not simple, atomic and independent products (such as books, movies etc.) but interdependent and sometimes complex components of a larger solution;
- No profiles: While typical B2C recommenders are used repeatedly by the same person, a digital consultancy service has no chance to build up customer profiles through repeated interactions; companies will usually access the service only once. Thus, a profile of the company needs to be acquired within a single session by the recommender; one can regard it as forming a *query* that describes the situation of the company seeking advice.

Despite some of these differences, one can establish a "digital consultancy process" that will make it possible to apply traditional recommender techniques that have been designed for classical preference-based B2C scenarios. Such a process is based on the following considerations (see also Figure 1):

1. Many companies share the same requirements, just like many persons share preferences. The similarity of requirements often depends on the companies' demographics (e.g., size, industry etc.). Thus, a first step in the digital consultancy process may be to capture company demographics and regard them as an initial company profile or **initial query**. This allows establishing a certain similarity between companies from the beginning.
2. Later, the similarity of context and requirements manifests itself in accepting similar suggestions from the recommender. Since solutions will be complex, one may construct a repeated interaction with the recommender in the form of iterations: after entering the company demographics (step 1), the business user receives a first set of recommendations and selects from those some first elements of a solution. These elements are added to the initial company profile to form an **extended query**, and the recommender is invoked again. This process is repeated, each time with a more verbose query. (We will later use the term "query verbosity" to refer to the growing amount of information that the query contains.)

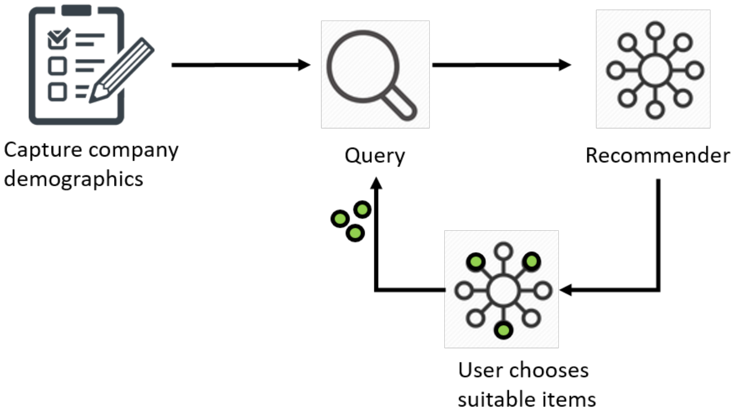

Figure 1. Iterative process for business consultancy recommenders.

Following this iterative process will allow us to assess the *similarity* of company contexts by comparing queries of a company to those of previous users of the service, with an increasing degree of accuracy as the query is iteratively extended. Since similarity is at the heart of both content-based and collaborative filtering approaches [5], being able to assess similarities is an important prerequisite for applying these approaches. In addition, we build up company profiles during the process, which makes it possible to apply content-based filtering. The iterative refinement also makes it possible to take into account the interdependence of solution elements by identifying, in each new step, new elements that fit to the already selected elements.

Although collaborative and content-based filtering become applicable through our iterative process, they may not be the best choice because a) collaborative filtering does not lend itself readily to incorporating company demographics (or other forms of general context) and b) because both do not foresee the use of human-provided knowledge about the business domain, which might be helpful. In fact, previous research has argued for the use of case-based reasoning (CBR) in business recommenders [6] because CBR is a proven way of re-using solutions for business problems. Constraint-based recommenders [4] are another family of algorithms that have been put forward as a good way of satisfying business requirements.

In our previous work [3], we used a graph, as a simple, flexible and easily extensible means of representing both historic user choices and explicit human knowledge about a business domain together with a random walk approach to generate recommendations. We found that explicit knowledge about associations between solution elements improves the recommender performance. On the other hand, taxonomic knowledge, e.g., about relationships between industries, did not help.

While our previous work was able to benefit from a graph's flexibility and ease of incorporating new domain knowledge easily [3,7], it is not perfectly suited to accommodate and make use of all possibly relevant attributes of a company's context. For instance, it does not easily allow representing and comparing numeric attributes (such as company size) or simple string attributes containing longer passages of text. On the other hand, in another work, we demonstrated that the CBR recommender gave good results where past experience plays a significant role in providing recommendations [8]. Thus, the goal of our extended research was to explore options of combining graph-based random walks with other forms of recommenders, above all CBR-based ones.

1.1. Application Scenario

We performed our recommender experiments in the domain of IT consultancy, more precisely business intelligence (BI) consultancy. Typically, before being able to tackle the "technical" elements of a BI solution, companies using a BI consultancy service initially seek advice regarding:

- Suitable *key performance indicators (KPIs)* that can be used to monitor and measure the company's success in achieving its goals, e.g., "sales revenue";
- Adequate *dimensions* to describe the values of KPIs, e.g., to characterise sales by product that was sold, channel through which it was sold and/or date when it was sold;
- Suitable *representations*, e.g., charts or tables that help to analyse KPI values along dimensions (e.g., a chart showing temporal evolution of sales revenue for different products).

Here, we focus on the first two types of solution elements. Obviously, the question of which dimensions should be chosen depends on the KPIs being monitored. The choice of KPIs, in turn, is often determined by the (type of) industry of a company, e.g., companies who produce energy tend to have KPIs that differ substantially from those of, say, architects. KPIs are also usually determined by the business process (e.g., sales) that should be analysed.

1.2. Contribution

Given the application scenario that we just sketched, our contribution is towards knowledge-based recommendations in B2B scenarios. Our main research goal in this work is to construct a new hybrid recommender that optimally supports the requirements of a B2B consultancy service in terms of contextually useful recommendations. We investigate hybridisation because:

1. Despite the existence of some previous work, we do not yet have reliable knowledge about which type of recommender is best suited for the task;
2. We do know that different recommenders have different strengths and weaknesses in general that affect their ability to represent and accommodate certain types of knowledge and/or inputs and their ability to deal with a lack of such knowledge ("cold-start problems").

We will first investigate the performance of algorithms individually and then, through a more detailed analysis of the strengths and weaknesses on the data, propose and evaluate some hybridisation strategies that will lead to superior performance by joining the strengths of the best-suited recommenders.

2. Related Work

In the literature, one finds two main classes of recommender systems: information-filtering-based and knowledge-based systems. The former category selects items from a large collection of items based on user preferences and is further classifed as collaborative-filtering recommenders and content-based filtering recommenders. The knowledge-based recommenders make recommendations by applying constraints or similarities based on domain or contextual knowledge. Common applications are in B2C scenarios such as e-commerce, tourism, news, movie, music, etc. [9]. In the following sections, we provide an overview of these approaches from a perspective of B2B recommendations, particularly consultancy services.

2.1. Digitalisation of Consultancy Services

Digitalising consultancy services has been discussed recently [10]. For the domain of IT consulting, a "computer-executed consulting (CEC) service" is proposed by Werth et al. [2], which replaces, most notably, the two steps of (a) interviewing client representatives and (b) creating a report that summarises the interview results. The digital service is designed by human consultants and consists of (a) a series of questionnaires (replacing the interviews) and (b) an automated report creation module. Obviously, there is a rough correspondence between these components and the step of (a) formulating a query and (b) getting recommendations for that query in Figure 1. The proposed CEC service is general-purpose. Therefore, although it mentions the need for more intelligence in the report creation module and the option of using recommender systems, it does not discuss any details of how to use recommenders.

The application of recommender systems has been discussed for more specific consultancy tasks such as optimisation of product assortments [11], selection of cloud or web services [12–14] or adaptation of conditions in agriculture [15]. In all these cases, the set of possible items that can be recommended is known and well-defined, and the task consists of selecting and possibly orchestrating the items. In its simplest interpretation, the term "orchestration" simply means that the selected services should be well aligned with each other, e.g., for optimal cross-selling opportunities [11] or for obtaining a consistent complex cloud service configuration [14]. This is also the case for our BI consultancy service, see Section 1.1 and [3].

2.2. Business-Oriented Recommendations

With regard to recommender algorithms, business-oriented recommender systems have to deal with **complexity** in terms of company contexts (input) and solutions (output). Attempts to deal with such complexity can be divided into several categories:

- *Augmentations of content-based filtering:*
  Approaches in this category model both the input and output complexities and establish the degree to which both of them match. For instance, constraint-based recommenders [4,16] help model product features and constraints to be expressed about them and then ensure constraint satisfaction. Other approaches use tree-like structures to model items and user preferences [17] or use multiple levels on which queries and items are matched (such as recommending first providers and then actual services in a service recommender [18]).
  In content-based filtering, additional knowledge can be incorporated, e.g., into the function that determines the similarity between an item and the user profile. Often, this is knowledge about user context, item features and/or domain-specific constraints. For instance, refs. [19,20] use ontologies to represent and reason about item features and to apply this knowledge in a sophisticated similarity measure that takes into account "hidden relationships" [20]. Middleton et al. [21] use an ontology to represent user profiles and engage users in correcting the profiles before assessing profile–item similarities. The complexity of business contexts has also been highlighted in [22], where the authors focus on identifying the criteria for recommendations in business processes that will serve as inputs to knowledge-based recommenders.
- *Augmentations of collaborative filtering:*
  Case-based recommenders [6,23] can be seen as a special form of collaborative filtering since they recommend items used in solutions of companies that are similar to the current company. However, instead of only considering already chosen items, case-

based recommenders' similarity measures take into account context variables that describe, e.g., company demographics and other relevant aspects of the company's problem and/or initial situation.
Since case-based reasoning is an approach based on problem-solving from past experience, case-based recommenders have been implemented in domains that most benefit from contextual information coming from past experience. For example, [24] explored case-based recommenders to recommend personalized financial products to the customers of a banking organisation. The authors of [25] argue that case-based recommenders are much more suitable for a complex domain of smart-city intiatives as they can utilize a rich range of domain-specific attributes.

- *Graph-based recommenders:*
  Recommender algorithms based on graph structures [7,26,27] have been put forward because of their ability to accommodate a wide variety of forms of contexts in a flexible way without much effort. Random walks [28–30] are a predominant type of algorithm to provide recommendations based on graph structures. Because of their simplicity, graphs also have limitations, e.g., in modeling and matching simple string-valued attributes of input cases or in modeling certain forms of complex solution structures. The possibility of using graph-based recommenders to "mimick" traditional recommender approaches, such as collaborative or content-based filtering, has been explored by Lee et al. [31]. For this, one needs to assign different weight to different types of graph relations.
  A comprehensive study by [32] has identified the potential of knowledge graphs to improve the explanability of recommendations as well as to provide dynamic recommendations. Wang et al. [33] have combined knowledge graphs with content-based filtering to support the recruitment of consultants with suitable skills for clients by calculating semantic similarity between the graph nodes.

Some studies have handled the complexity in B2B situations by adopting neural approaches in conjunction with one of the techniques described above [33,34].

Obviously, all of these approaches employ and model various types of knowledge. An overview of the different kinds of knowledge that recommenders may use can be found in [3,4]. What distinguishes the business recommenders from most others is the use of *domain knowledge*. Often, this knowledge is obtained from human experts [3,4,35].

2.3. Evaluation of Business-Oriented Recommenders

It is important to evaluate the performance of recommenders, more so when the recommendations are expected to be comparable to those of human experts. A common and popular metric for recommender evaluation is accuracy. Herlocker et al. [36] classify recommender accuracy into three categories: predictive accuracy, classification accuracy and rank accuracy. However, accurate recommendations may not always indicate useful recommendations. Hence, recommenders may be evaluated based on additional metrics such as diversity, novelty, coverage, serenditpity, etc. [37,38]. Some of these metrics are subjective to user preferences and are used to improve user engagement in B2C scenarios.

In the context of B2B recommendations, however, not every metric is relevant. For BI consultancy, for instance, the recommendations are dependent on the domain of the customer, and accuracy in terms of ranking the recommendations is more value-adding for the customers than providing novel or diverse suggestions. Consequently, an item should always be added to the recommendations if it is *relevant*, even if it is not *novel*. The customers expect the recommender to provide recommendations ordered by relevance and usefulness, with the most relevant suggestions at the top. The top recommendations then can be iteratively tuned by adjusting the input to the recommender (query). Thus, for the evaluation of our recommender outputs, we adopted the relevance judgement approach to evaluate *rank accuracy* using the metric Mean Average Precision (MAP) [39]. MAP is commonly used to evaluate the quality of ranking by calculating the average precision at every rank for a query and then computing the mean of all average precisions for all the queries. Metrics such as diversity and coverage may be relevant in occasional cases, e.g., for customers from a new industry (not considered in past consultations) or customers that expect non-standard solutions.

2.4. Hybrid Recommenders

Forming hybrid recommenders [40,41] is an active field of research since combinations of different approaches can often help to combine the strengths and/or avoid the weaknesses of the combined approaches. For instance, content-based filtering can be combined with collaborative filtering, e.g., to mitigate the so-called cold-start problems associated with collaborative filtering, i.e., problems with recommending newly introduced items or serving new users: new items can be recommended immediately by content-based techniques as long as they have a meaningful description that can be matched against user profiles. Besides cold start problems, hybridisation can be used, e.g., to augment similarity in collaborative filtering with the reasons behind user preferences and thus give it a stronger CBR flavour [42]. Another motivation for using hybridisation is to improve the quality of recommendations. For example, Rivas et al. [43] combine CBR recommendations with multi-agent systems to improve the accuracy of recommendations. Further possible complementary strengths and weaknesses of knowledge-based and knowledge-weak recommenders are discussed in [44].

In order to effectively combine the strengths of individual recommendation techniques, Burke [40] has proposed seven different hybrid strategies: weighted, mixed, switching, feature combination, cascade, feature augmentation and meta-level. These strategies are still being successfully applied to address various problems in recommender systems. For instance, Rebelo et al. [45] have used the cascade strategy to improve the *novelty* and *diversity* of recommendations; Alshammari et al. [46] have applied the switching strategy to address the problem of *long tail recommendations*; Hu et al. [47] have combined algorithms in a cascading fashion to improve the *personalization* of recommendations; and Gatzioura et al. [48] have implemented a meta-level hybrid recommender to explore metrics such as *coherence* and *diversity* in music recommendations.

In the context of the problem of this research, we explore hybridisation to improve the *ranking accuracy* of the recommendations by first evaluating different recommender techniques individually and then identifying an appropriate strategy to combine them.

Overall, there is a rather large number of suggestions for enriching recommenders with contextual knowledge. However, as outlined in Section 1.2, we see a gap in exploring which of these suggestions is best suited to support scenarios of business consultancy. We furthermore see a need to gain a deeper understanding of the (complementary) strengths and weaknesses of the mentioned approaches that will lead to successful hybridisation strategies.

3. Methodology

As mentioned in Section 1.2, the main goal of our research is to find a recommender form that optimally supports B2B consultancy services. To study such services, we worked together with a company that provides BI consultancy, as described in Section 1.1.

3.1. Awareness of Current Consultancy Practice

As described in our previous work [3], our research started by interviewing two consultants to understand how they work and which knowledge they require to make the necessary recommendations to their customers. We also obtained some documents that were used to document the outcomes of meetings and workshops with customers.

This was the basis of our definition of the structure of consultancy cases. It gave us an insight into the demographic and contextual variables (attributes) that consultants need to know about each company. It also allowed us to roughly grasp the kind of reasoning that they employed to transfer their experiences to new cases. The corresponding findings are summarised in Section 4.

The details of the past cases and corresponding recommendations were recorded by the BI consultancy as structured data in an excel sheet. We then constructed a case base out of the past experiences of the consultancy and identified cases that represent the business context of customers; each business process that a company wanted to analyse resulted in a separate case. Overall, this resulted in a case base with 82 entries. The data were prepared separately for every recommender configuration, as described in Section 5.

To support our extended research, we performed a second round of interviews to gain further awareness of how consultants currently assess (implicitly or explicitly) the similarity between customer cases. More precisely, we asked them to which degree they take into account each attribute in the case (e.g., the industry, the core business processes to be analysed, the target group, the goal of the consultation), i.e., we elicited the importance they assign to each attribute while deriving recommendations for their customers.

3.2. Recommender Selection

Next, we used the gathered knowledge to configure a selection of recommender algorithms that we wanted to compare:

- Collaborative filtering, using both item-based and user-based k-nearest neighbour algorithms, as provided by the LibRec library [49];
- An implementation of a random walk algorithm based on a "case graph", which has been described in our previous work [3];
- A CBR-based recommender that applies similarity-weighted scoring to the elements contained in similar cases, as provided in the jColibri library [50]. The weights mentioned in Table 1 were used here to define the contribution of the local similarities within the global similarity function in CBR.

A precise description of recommender configurations can be found in Section 5.

Table 1. Local similarities and weights for CBR recommender.

Case Attribute	Local Similarity Measure	Weight
Industry	Taxonomy	0.24
Goal	TF-IDF	0.06
Target Group	Jaccard	0.1
KPIs and dimensions	TF-IDF	0.6

3.3. Experiments

We then designed an experimental setup [51] to compare the initial recommender configurations and our new hybrid recommender strategies.

This setup consists of a *leave-one-case-out* evaluation: for each case C, we used the case base as the training data by *omitting* C. Out of C, we constructed queries Q_C at different verbosity levels: simple queries with no input elements and gradually more verbose queries containing an increasing number of randomly chosen KPIs from the case C.

The random selection of input elements is not realistic as this information is usually provided by the customer. However, we did not have any information about the order in which customers added elements to their solution in the past and thus had to resort to this strategy.

As explained in Section 2.3, we calculated MAP to measure the *ranking accuracy* of all recommenders. Since we want to compare the performance of the recommenders to that of the consultants, MAP lets us evaluate both accuracy and reliability against historical data that were recorded from the past consultations. Thus, we used the knowledge of originally chosen elements in C as a definition of relevance: for a query Q_C, we observed whether a recommender was able to retrieve (and rank highly) the elements in the original case C. That is, for each *ranking* of recommended items that a recommender produced in response to a query Q_C, we computed the MAP of these rankings by treating all elements originally contained in C as relevant and all others as irrelevant.

As mentioned above, this experimental setup was first used to evaluate each recommender in isolation. We then analysed the strengths and weaknesses of each recommender (see Section 6) and formed new hybrid recommender strategies (see Section 7) that we evaluated with the same experimental environment to see whether the hybridisation could bring about an improvement (see Section 8).

4. Interview Findings: Case Structure and Similarity Measure

As mentioned in Section 3.1, we performed two rounds of interviews with consultants to understand their current work and knowledge processing procedures. Here, we summarise the findings from both rounds of interviews (see also [3] for more details on the first round):

- Customers often come to the meetings with some important KPIs and dimensions (i.e., solution elements) already in mind. However, the degree to which customers have initial ideas can vary greatly. We have reflected this variance by creating queries at different verbosity levels.
- In terms of company demographics, consultants consider the industry of a customer as the main criterion for finding similar past cases. Further relevant variables that we elicited were the target group of the solution (e.g., only management or all employees) and the goal of the BI project (expressed in natural language). Finally, consultants use all known customer preferences from initial meetings (see above), i.e., any already known solution elements to remember past cases with similar elements.

  The business process was also mentioned by consultants as an important variable. Because of its importance, we chose not to use it simply as a ranking criterion for the retrieval of similar cases, but as a filter. For a given company, we created separate cases for each business process the company wanted to analyse and retrieved only cases with the same business process. (Analogously, we built separate case base graphs for the graph recommender, see below.)
- In the second round of interviews, we asked the consultants to quantify the relative importance of these types of attributes in terms of percentage weight. This task can be difficult as the weights identified by different consultants can vary. However, since the relative importance of the attributes is based on domain-knowledge and not on personal preferences of the consultants, we were quickly able to arrive at a consensus about the weights. Although quantifying something as abstract as a variable's contribution to a similarity score is a hard task, we were able to verify in some preliminary experiments that the chosen weights gave quite good results compared to other potential weight configurations. The resulting weights are shown in Table 1.
- When talking to a customer from an unknown industry, consultants tried to remember cases of customers from *similar* industries. Since our attempts to use an industry taxonomy for improved similarity assessment in a graph-based recommender were not particularly successful, we did not consider this kind of reasoning in this work. However, we did use the industry taxonomy to define a local similarity measure for industries within a CBR recommender (see Section 5.3).

5. Recommender Configurations

Based on the interview findings, we created suitable configurations of the recommenders to be used in the experiments [51], as described in the following subsections.

5.1. Collaborative Filtering

Since the association between solution elements (which we will call items for simplicity) and cases is binary—an item is either part of the case's solution or not—we can describe this situation as one of "implicit feedback recommendation" [52]. It means that the user–item matrix does not contain true ratings but binary entries; in our case, we replaced users with customers.

However, this does not require changing the way in which collaborative filtering algorithms work on the matrix. In our experiment, we used the user-based *userknn* and the item-based *itemknn* implementations from the LibRec package [49]. Thus, as a part of data preparation, we separated the 82 cases to represent data in a User-Item-Rating format, where case represented users and KPIs in the solution represented items. The rating for all items that were chosen as the case's solution was represented as 1 and the rest as −1.

Since *userknn* and *itemknn* do not allow us to make use of the additional attributes listed in Table 1, "simple" queries that do not contain any items (verbosity level 0) could not be designed. We also expect the collaborative filtering algorithms to have inferior results for low-verbosity queries.

5.2. Random Walk Recommender

The configuration for the graph-based recommender was re-used from [3], where the case graph incorporated the explicit knowledge acquired from the consultants. The data were prepared to represent every case as a delimited collection of key–value pairs. In this technique, the case graph was built by creating a node for each case and connecting it to a node representing the industry as well as to nodes representing solution elements. As mentioned in Section 4, we built a separate graph for each business process to be analysed.

Target group and goal were not represented in this approach: since there are only three possible target groups, the corresponding nodes would have had a very high degree, thus diluting the PageRank scores. Since goals are string attributes, a node representation was not straightforward for them (although future work might consider extracting salient terms and representing them as nodes).

The recommended elements were scored using the PageRank with Priors algorithm [53] on that graph. The scores represent the probability of reaching a node in the case graph (e.g., the elements to be recommended) through a random walk that is biased towards the input elements in the query.

For verbosity level 0, the random walk-based recommender uses only the industry node as a query—we also expect suboptimal results here.

5.3. CBR Recommender

In the case of the CBR recommender, primarily three factors were considered in the configuration:

- Similarity measures depending on attribute type: Based on the taxonomy-tree approach proposed by [54], the industry attribute uses the industry taxonomy derived by [3] that categorizes the customers of the consultancy based on their similarities (e.g., customers that are likely to share KPIs and dimensions). For the attributes goal (free text) and KPI, we could apply the TF-IDF [55] similarity measure by creating a corpus of goals and KPIs, respectively, from the case base for the computation of inverse document frequencies (IDFs). Although KPIs are not free text, applying TF-IDF is appropriate to disregard repeated terms such as "Number of" or "Amount" since they do not add significant value to the recommendations. Lastly, for the attribute target group, we calculated the Jaccard coefficient [55] as a case may have more than one target audience from the possible values "employees"/"middle management"/"top management".
- The number n of the most relevant (top) cases retrieved: The number of the retrieved cases played a significant role in calculating the scores of the recommended elements, which in turn determine the ranking. For an element appearing in any of the retrieved cases $R(Q_C)$ for a query Q_C, the score of that element is the sum of the scores of all the retrieved cases in which the element occurs:

$$score(i) = \sum_{C_j \in R(Q_C): i \in C_j} sim(C, C_j) \tag{1}$$

Obviously, the larger the case base, the larger we can choose n, i.e., the maximum size of $R(Q_C)$. For a rather small case base like ours, we expect that smaller values of n will work better since larger values will likely imply a "topic drift" by including rather dissimilar cases. The score of the case $sim(C, C_j)$ was generated by the CBR recommender using the global similarity function, which is the weighted average of the local similarity measures [56]: $sim(C, C_j) = \sum w_k sim_k(C, C_j)$.

- For that weighted average, we used the weights assigned to the local similarity measures sim_k shown in Table 1.

The retrieved ranking of matching cases was first filtered by business process such as to return only cases with matching process before applying the local similarity measures.

6. Experiment 1: Strengths and Weaknesses of Recommenders

The goal of our first experiment was to identify a recommendation technique that performs well for different query verbosity values [51]. The results of Experiment 1 are shown in Table 2. Note that the verbosity refers to the absolute number of solution elements that the query contained.

Table 2. Experiment 1: MAP values for individual recommendation techniques for different configurations.

Query Verbosity	User-Knn	Item-Knn	Graph-Based	CBR			
				Top 1	Top 2	Top 3	Top 5
0	-	-	0.408	0.773	0.783	0.773	0.747
5	0.487	0.420	0.566	0.777	0.805	0.774	0.714
10	0.497	0.416	0.646	0.785	0.805	0.772	0.719
15	0.498	0.413	0.689	0.787	0.807	0.766	0.709
20	0.501	0.411	0.713	0.787	0.807	0.776	0.713
30	0.498	0.411	0.733	0.787	0.812	0.780	0.716
40	0.499	0.411	0.742	0.787	0.812	0.783	0.719
100	0.499	0.409	0.746	0.787	0.812	0.781	0.718

From the results, we can see very clearly that the collaborative filtering algorithms obviously suffer too much from their inability to accommodate contextual knowledge. Their performance is substantially worse than that of the other recommenders.

Regarding those, we observed that the performance of the CBR recommender is better than the graph-based recommender; however, there is no improvement in the performance of the CBR recommender above a certain query verbosity. Thus, one can see that retrieving a single case is restrictive for the recommendations since only a limited number of elements are available which in turn creates a recall problem. The performance of the graph-based recommender, on the other hand, steadily improves as more elements are added to the query. In order to enable the CBR recommender to stretch its (better) performance to any size of the query, we repeated the leave-one-case-out evaluation by retrieving more number of most relevant cases. With the top two retrieved cases, the performance of the CBR recommender improved further, however, again only up to a certain query verbosity. By retrieving more and more cases, it was possible to overcome the recall problem and achieve a steady improvement in the performance of the CBR recommender, similar to the graph-based recommender. Nonetheless, one can observe that retrieving more cases also introduces more noise, consequently decreasing the overall performance of the CBR recommender. Thus, increasing the number of retrieved cases seems to be neither the optimal nor a generic solution to the recall problem of the CBR recommender because of its severe precision-degrading effect.

Overall, to achieve an optimum performance, the CBR recommender needs to be configured to retrieve a low number of the most-relevant cases. However, if a customer needs a solution with more elements than are available in the (small number of) retrieved cases, the CBR recommender fails to expand its range of recommendations. The graph-based recommender, on the other hand, can leverage the whole range of elements available in the case base and hence seems to be a better solution for increasing recall without adding too much noise. We, therefore, see a benefit in combining the graph-based and CBR recommendation techniques using a hybrid strategy.

7. A New Hybrid Recommender Design

In Section 2, we saw that hybrid recommender systems are commonly used to overcome the weaknesses of individual recommendation techniques. Of the seven hybrid recommender strategies described by [41], strategies such as *switching*, *cascade* or *mixed* are not ideal (and the others are not applicable), as the results show that the CBR recommender is clearly the better performer. Since we would like the graph-based recommender to contribute by adding more relevant elements where CBR is limited, we adopted the weighted combination method [41] because it allows us to "overrule" the decisions of the CBR by adjusting the importance (weight) given to either CBR or graph-based recommender. A representation of the weighted hybrid strategy designed by us is shown in Figure 2.

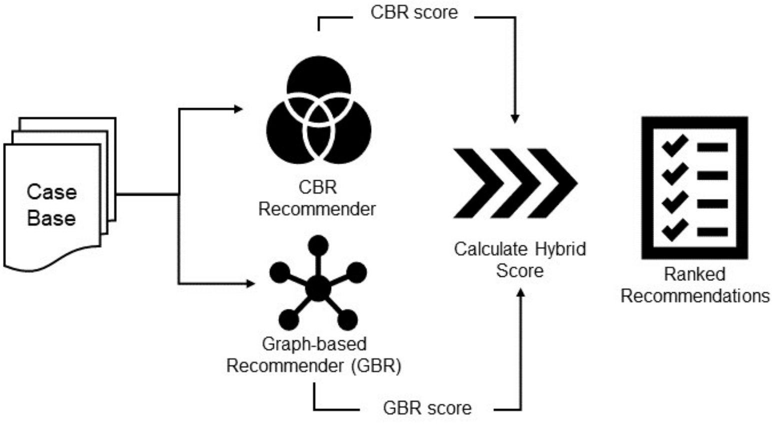

Figure 2. Design of weighted hybrid recommendation strategy.

For designing the hybrid strategy, we built upon the CBR configuration to retrieve the most relevant two cases, as this configuration achieved the optimum performance in the previous experiment. We now explore if the recall issue of CBR can be resolved by adding some component of the graph-based recommendations (GBR). We first normalised the scores of the individual recommendation techniques using min-max normalisation since the graph-based and CBR recommenders have their own (different) scoring mechanisms, as described in Section 5. We then combined the normalised scores of both recommenders and calculated the hybrid weighted score using Equation (2).

$$hybrid\_score(item) = \alpha \cdot |CBR(item)| + (1 - \alpha) \cdot |GBR(item)| \tag{2}$$

where $|\cdot|$ refers to min-max score normalisation.

Because of the CBR recommender's strength in dealing with sparse, i.e., low-verbosity queries and the relative strength of the graph-based recommender in handling high-verbosity queries, we made the mixture parameter α dependent on the query verbosity, i.e., the number of referred elements $|q|$ in the query q:

$$\alpha = \begin{cases} 1 - \frac{(1-\beta)\cdot|q|}{\bar{c}} & \text{if } |q| \leq \bar{c} \\ \beta & \text{otherwise} \end{cases}, 0 < \beta < 1 \tag{3}$$

Here, $\bar{c}$ refers to half the average size of all cases in the case base in terms of their number of referred elements (KPIs) and serves as the "verbosity threshold", whereas β is a parameter that controls the contribution of the CBR recommender.

Since CBR was the better performer of the two recommendation techniques, we designed Equation (3) such that the weight of the CBR recommender (α) is never 0 in Equation (2). On the other hand, we do not set β to 1 as this would always give a full weight to CBR, which we already know has limitations performing as a "pure" recommender. Additionally, from the results of Experiment 1, we concluded that a CBR-dominated hybrid recommender would perform better for queries below the verbosity threshold and vice versa, also taken care of in Equation (3). Figure 3 shows the dependency between α and query verbosity $|q|$ graphically for different values of β and $\bar{c} = 14$. We can see that α is always 1 when $|q|$ is 0. We can also see how β acts as the "minimum CBR contribution" and that below the verbosity threshold, less and less weight is given to CBR as the verbosity increases.

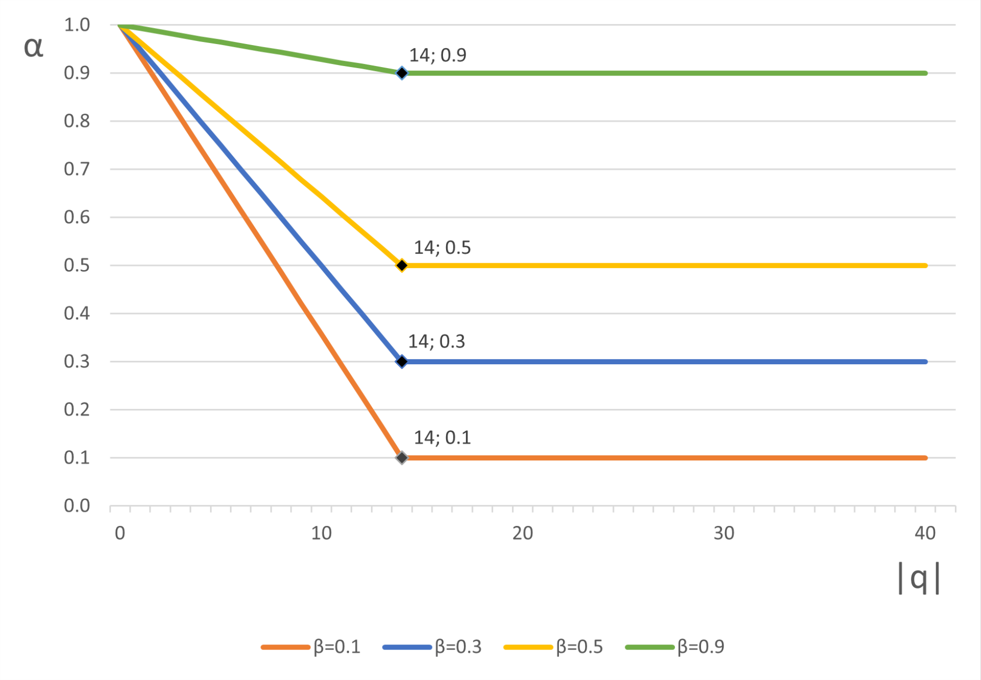

Figure 3. Mixture parameter α as a function of query verbosity q, β controlling minimum CBR contribution.

With this setup, we carried out the second experiment—leave-one-case-out evaluation for different query verbosity using the hybrid strategy. Our goal, now, was to find the appropriate combination of weights that could overcome the recall issue of CBR without impacting its performance. After every run, we compared the MAP values for each recommendation technique with that of the hybrid strategy, as seen in Table 3.

Table 3. Experiment 2: MAP values for individual recommendation techniques and hybrid strategy.

Query Size (Verbosity)	Graph-Based	CBR Top-2 Cases	Hybrid $\beta = 0.1$	Hybrid $\beta = 0.3$	Hybrid $\beta = 0.5$	Hybrid $\beta = 0.9$
0	0.408	0.783	0.801	0.801	0.801	0.801
5	0.566	0.805	0.836	0.835	0.836	0.835
10	0.646	0.805	0.843	0.845	0.839	0.840
15	0.689	0.807	0.804	0.856	0.849	0.849
20	0.713	0.807	0.817	0.862	0.853	0.851
30	0.733	0.812	0.821	0.864	0.856	0.852
40	0.742	0.812	0.825	0.866	0.858	0.854
100	0.746	0.812	0.827	0.866	0.858	0.854

8. Experiment 2: Performance of Hybrid Recommender

To find the right combination of the graph-based and CBR recommender, we experimented with different values of β, starting with a very low value. The lower values of β indicate a higher weight to the graph-based recommender. As can be seen in Table 3, the performance of the hybrid recommender appears to be better than either of the individual recommendation techniques; however, the precision issue of the graph recommender still shows its negative impact for very low values of β. The verbosity threshold for our experiments was at 14, and it can be observed that the performance suddenly dips at 15 input elements for $\beta = 0.1$ (where MAP = 0.843 for a verbosity of 10 and MAP = 0.804 for verbosity 15).

On the other hand, although a high β resolves the precision problem, the performance is not optimum because for, e.g., $\beta = 0.9$, the graph recommender's ability to provide more recall is not sufficiently leveraged.

From the results for $\beta = 0.3$, one can conclude that the right value of β can cure both the recall problem of the CBR recommender and the precision problem of the graph recommender and thus can give an optimum performance among the individual recommendation techniques and the various configurations of the hybrid strategy put together.

Finally, we take a look at the time complexity of the hybrid recommender strategy, which is dependent on the complexities of the recommender systems that we combined.

$$O_{hyb} = max(O_{GBR}(k_{GBR} * N_{GBR}), O_{CBR}(k_{CBR} * N_{CBR})) \tag{4}$$

where:

k_{GBR} = no. of iterations over the graph;
N_{GBR} = no. of edges in the graph;
k_{CBR} = no. of local similarity measures;
N_{CBR} = no. of cases in the case base.

The time complexity of the graph-based recommender is similar to that of PageRank with Priors [53], i.e., depending on the no. of iterations and the no. of edges in the graph, whereas the complexity of the CBR recommender depends on the no. of cases in the casebase and the no. of local similarity measures, which are roughly equal to the no. of attributes used to describe the cases. Thus, the overall complexity of the hybrid recommender is the maximum from either of these recommendation techniques, as depicted in Equation (4).

9. Conclusions and Future Work

In this work, we considered the application of recommender systems to business consultancy. We have argued how certain consultancy tasks can be formulated as recommendation problems, especially in the domain of IT consultancy, e.g., selection and orchestration of web services or selection of Key Performance Indicators and dimensions for BI solutions. Since such problems are in several respects different from the typical, purely preference-based B2C recommenders, we have addressed the question which (combinations of) recommendation techniques are most suitable for these new B2B scenarios.

We worked with data from the BI consultancy domain and performed experiments with a range of known recommender techniques. These techniques offer a varying degree of possibility to feed—besides the item choices that a company makes—contextual knowledge, such as company demographics, into the algorithm. This ranges from none (collaborative filtering) over limited (graph-based random walks) to full coverage (CBR-based recommender).

Our initial comparison showed that—as one might expect—the CBR-based recommendation benefits from its ability to accommodate more contextual knowledge and provides the best results. However, we also recognised a limitation: CBR-based recommenders have a free parameter, namely n, the number of most similar cases to use for the identification of possible solution elements. We found that, for the rather small case base in our experiments, small values of n performed better. Obviously, a larger n implies more noise coming from more dissimilar cases. In our previous work [3], we already observed that including cases from different but similar industries can be dangerous.

On the other hand, limiting n also limits the potential recall of the recommender, i.e., some useful items from less similar cases are excluded. Obviously, a graph-based approach—although less precise—offers a natural way to include more items from the more dissimilar cases.

We, therefore, explored the combination of CBR-based recommendation with a graph-based recommender in order to combine its strengths in terms of precision with the graph-based recommender's strength in providing more relevant items in the lower ranks. We followed a weighted hybridisation strategy. The weight was dynamic, giving more and more importance to the graph recommender with the growing size of the query. This makes sense since contextual knowledge becomes less important as we know more and more about already chosen items. Because of the superior performance of the CBR recommender, we also designed the weighting so as to ensure that there is always a certain minimum weight given to it. It turned out that, indeed, this minimum weight should not be 0.

We found that the weighted hybrid performed better than any of the individual recommenders at all levels of query verbosity. Although we have only tested the hybrid strategy on one particular data set, we believe that we can carefully conclude from this that a CBR recommender's problems in balancing between precision and recall can be overcome by combining it with another recommender that is less limited by case boundaries and can contribute better recall at lower ranks. The graph-based recommender was able to achieve that in our experiments.

In future work, we plan to apply our approach to different domains and data sets. In that context, it will also be important to study the relationship between the size and characteristics of the case base and the optimal choice of the parameter n of the case-based recommender more closely. Further investigation also involves recommendations to customers who expect non-standard solutions or who belong to a domain very different from the past cases. However, this involves additional measures such as maintaining a catalogue of solutions from different customer domains—added knowledge engineering effort is a known drawback for knowledge-based recommenders. For such non-standard recommendations, additional metrics such as coverage and diversity can be evaluated to deliver more value to the customers.

Author Contributions: Conceptualization, C.P. and H.F.W.; Data curation, C.P.; Funding acquisition, H.F.W.; Investigation, C.P.; Methodology, C.P. and H.F.W.; Software, C.P.; Supervision, H.F.W. and A.M.; Validation, C.P.; Writing—original draft, C.P. and H.F.W.; Writing—review and editing, H.F.W. and A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Innosuisse-Swiss Innovation Agency grant number 25665.1 PFES-ES.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nissen, V. Digital transformation of the consulting industry—Introduction and overview. In *Digital Transformation of the Consulting Industry*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 1–58.
2. Werth, D.; Zimmermann, P.; Greff, T. Self-service consulting: Conceiving customer-operated digital IT consulting services. In Proceedings of the AMCIS 2016, San Diego, CA, USA, 11–14 August 2016.
3. Witschel, H.; Martin, A. Random Walks on Human Knowledge: Incorporating Human Knowledge into Data-Driven Recommender. In Proceedings of the 10th International Conference on Knowledge Management and Information Sharing (KMIS), Seville, Spain, 18–20 September 2018.
4. Felfernig, A.; Burke, R. Constraint-based recommender systems: Technologies and research issues. In Proceedings of the 10th International Conference on Electronic Commerce, Innsbruck, Austria, 19–22 August 2008; p. 3.
5. Bobadilla, J.; Ortega, F.; Hernando, A.; Gutiérrez, A. Recommender systems survey. *Knowl.-Based Syst.* **2013**, *46*, 109–132. [CrossRef]
6. Bridge, D.; Göker, M.H.; McGinty, L.; Smyth, B. Case-based recommender systems. *Knowl. Eng. Rev.* **2005**, *20*, 315–320. [CrossRef]
7. Minkov, E.; Kahanov, K.; Kuflik, T. Graph-based recommendation integrating rating history and domain knowledge: Application to on-site guidance of museum visitors. *J. Assoc. Inf. Sci. Technol.* **2017**, *68*, 1911–1924. [CrossRef]
8. Witschel, H.F.; Peter, M.; Seiler, L.; Parlar, S.; Grivas, S.G. Case Model for the RoboInnoCase Recommender System for Cases of Digital Business Transformation: Structuring Information for a Case of Digital Change. In Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2019), Vienna, Austria, 17–19 September 2019; SciTePress: Setubal, Portugal, 2019; pp. 62–73.
9. Burke, R.; Ramezani, M. Matching recommendation technologies and domains. In *Recommender Systems Handbook*; Springer: Boston, MA, USA, 2011; pp. 367–386.
10. Deelmann, T. Does digitization matter? Reflections on a possible transformation of the consulting business. In *Digital Transformation of the Consulting Industry*; Springer: Cham, Switzeland, 2018; pp. 75–99.
11. Witschel, H.F.; Galie, E.; Riesen, K. A Graph-Based Recommender for Enhancing the Assortment of Web Shops. In Proceedings of the Workshop on Data Mining in Marketing DMM'2015, Hamburg, Germany, 11–14 July 2015.
12. Zhang, M.; Ranjan, R.; Nepal, S.; Menzel, M.; Haller, A. A declarative recommender system for cloud infrastructure services selection. In *International Conference on Grid Economics and Business Models*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 102–113.
13. Kritikos, K.; Laurenzi, E.; Hinkelmann, K. Towards business-to-IT alignment in the cloud. In *Advances in Service-Oriented and Cloud Computing. ESOCC 2017*; Springer: Cham, Switzerland, 2017; pp. 35–52.
14. Yao, L.; Sheng, Q.Z.; Ngu, A.H.; Yu, J.; Segev, A. Unified collaborative and content-based web service recommendation. *IEEE Trans. Serv. Comput.* **2015**, *8*, 453–466. [CrossRef]
15. Laliwala, Z.; Sorathia, V.; Chaudhary, S. Semantic and rule based event-driven services-oriented agricultural recommendation system. In Proceedings of the 26th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'06), Lisboa, Portugal, 4–7 July 2006; pp. 24.
16. Felfernig, A.; Friedrich, G.; Jannach, D.; Zanker, M. Constraint-based recommender systems. In *Recommender Systems Handbook*; Springer: Boston, MA, USA, 2015; pp. 161–190.
17. Wu, D.; Zhang, G.; Lu, J. A fuzzy preference tree-based recommender system for personalized business-to-business e-services. *IEEE Trans. Fuzzy Syst.* **2015**, *23*, 29–43. [CrossRef]
18. Mohamed, B.; Abdelkader, B.; M'hamed, B.F. A multi-level approach for mobile recommendation of services. In Proceedings of the International Conference on Internet of things and Cloud Computing, Cambridge, UK, 22–23 March 2016; p. 40.
19. Carrer-Neto, W.; Hernández-Alcaraz, M.L.; Valencia-García, R.; García-Sánchez, F. Social knowledge-based recommender system. Application to the movies domain. *Expert Syst. Appl.* **2012**, *39*, 10990–11000. [CrossRef]
20. Blanco-Fernández, Y.; Pazos-Arias, J.J.; Gil-Solla, A.; Ramos-Cabrer, M.; López-Nores, M.; García-Duque, J.; Fernández-Vilas, A.; Díaz-Redondo, R.P.; Bermejo-Muñoz, J. A flexible semantic inference methodology to reason about user preferences in knowledge-based recommender systems. *Knowl.-Based Syst.* **2008**, *21*, 305–320. [CrossRef]
21. Middleton, S.E.; Shadbolt, N.R.; De Roure, D.C. Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst. (TOIS)* **2004**, *22*, 54–88. [CrossRef]
22. Revina, A.; Rizun, N. Multi-Criteria Knowledge-Based Recommender System for Decision Support in Complex Business Processes. In Proceedings of the Workshop on Recommendation in Complex Scenarios co-located with 13th ACM Conference on Recommender Systems (RecSys 2019), Copenhagen, Denmark, 20 September 2019; pp. 16–22.
23. Bousbahi, F.; Chorfi, H. MOOC-Rec: A case based recommender system for MOOCs. *Procedia-Soc. Behav. Sci.* **2015**, *195*, 1813–1822. [CrossRef]
24. Hernandez-Nieves, E.; Hernández, G.; Gil-González, A.B.; Rodríguez-González, S.; Corchado, J.M. CEBRA: A CasE-Based Reasoning Application to recommend banking products. *Eng. Appl. Artif. Intell.* **2021**, *104*, 104327. [CrossRef]
25. Anthony Jnr, B. A case-based reasoning recommender system for sustainable smart city development. *AI Soc.* **2021**, *36*, 159–183. [CrossRef]
26. Bogers, T. Movie recommendation using random walks over the contextual graph. In Proceedings of the 2nd International Workshop on Context-Aware Recommender Systems, Barcelona, Spain, 26 September 2010.

27. Zhang, Z.; Zeng, D.D.; Abbasi, A.; Peng, J.; Zheng, X. A random walk model for item recommendation in social tagging systems. *ACM Trans. Manag. Inf. Syst. (TMIS)* **2013**, *4*, 8. [CrossRef]
28. Fouss, F.; Pirotte, A.; Renders, J.M.; Saerens, M. Random-Walk Computation of Similarities Between Nodes of a Graph with Application to Collaborative Recommendation. *IEEE Trans. Knowl. Data Eng.* **2007**, *19*, 355–369. [CrossRef]
29. Huang, Z.; Chung, W.; Ong, T.H.; Chen, H. A Graph-based Recommender System for Digital Library. In Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries, Portland, OR, USA, 14–18 July 2002; pp. 65–73.
30. Liu, Y.; Ma, H.; Jiang, Y.; Li, Z. Learning to recommend via random walk with profile of loan and lender in P2P lending. *Expert Syst. Appl.* **2021**, *174*, 114763. [CrossRef]
31. Lee, S.; Park, S.; Kahng, M.; Lee, S.G. PathRank: Ranking nodes on a heterogeneous graph for flexible hybrid recommender systems. *Expert Syst. Appl.* **2013**, *40*, 684–697. [CrossRef]
32. Chicaiza, J.; Valdiviezo-Diaz, P. A comprehensive survey of knowledge graph-based recommender systems: Technologies, development, and contributions. *Information* **2021**, *12*, 232. [CrossRef]
33. Wang, Y.; Allouache, Y.; Joubert, C. A Staffing Recommender System based on Domain-Specific Knowledge Graph. In Proceedings of the 2021 Eighth International Conference on Social Network Analysis, Management and Security (SNAMS), Gandia, Spain, 6–9 December 2021; pp. 1–6.
34. Nia, A.G.; Lu, J.; Zhang, Q.; Ribeiro, M. A Framework for a Large-Scale B2B Recommender System. In Proceedings of the 2019 IEEE 14th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), Dalian, China, 14–16 November 2019; pp. 337–343.
35. Tarus, J.K.; Niu, Z.; Mustafa, G. Knowledge-based recommendation: A review of ontology-based recommender systems for e-learning. *Artif. Intell. Rev.* **2018**, *50*, 21–48. [CrossRef]
36. Herlocker, J.L.; Konstan, J.A.; Terveen, L.G.; Riedl, J.T. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst. (TOIS)* **2004**, *22*, 5–53. [CrossRef]
37. Silveira, T.; Zhang, M.; Lin, X.; Liu, Y.; Ma, S. How good your recommender system is? A survey on evaluations in recommendation. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 813–831. [CrossRef]
38. Ge, M.; Delgado-Battenfeld, C.; Jannach, D. Beyond accuracy: Evaluating recommender systems by coverage and serendipity. In Proceedings of the Fourth ACM Conference on Recommender Systems, Barcelona, Spain, 26–30 September 2010; pp. 257–260.
39. Voorhees, E.M.; Harman, D.K. *TREC—Experiment and Evaluation in Information Retrieval*; The MIT Press: Cambridge, MA, USA, 2006.
40. Burke, R. Hybrid recommender systems: Survey and experiments. *User Model. User-Adapt. Interact.* **2002**, *12*, 331–370. [CrossRef]
41. Burke, R. Hybrid web recommender systems. In *The Adaptive Web*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 377–408.
42. Burke, R. A case-based reasoning approach to collaborative filtering. In European Workshop on Advances in Case-Based Reasoning, Trento, Italy, 6–9 September 2000; Springer: Berlin/Heidelberg, Germany, 2000; pp. 370–379.
43. Rivas, A.; Chamoso, P.; González-Briones, A.; Casado-Vara, R.; Corchado, J.M. Hybrid job offer recommender system in a social network. *Expert Syst.* **2019**, *36*, e12416. [CrossRef]
44. Burke, R. Knowledge-based recommender systems. *Encycl. Libr. Inf. Syst.* **2000**, *69*, 175–186.
45. Rebelo, M.Â.; Coelho, D.; Pereira, I.; Fernandes, F. A New Cascade-Hybrid Recommender System Approach for the Retail Market. In *International Conference on Innovations in Bio-Inspired Computing and Applications*; Springer: Cham, Switzerland, 2021; pp. 371–380.
46. Alshammari, G.; Jorro-Aragoneses, J.L.; Polatidis, N.; Kapetanakis, S.; Pimenidis, E.; Petridis, M. A switching multi-level method for the long tail recommendation problem. *J. Intell. Fuzzy Syst.* **2019**, *37*, 7189–7198. [CrossRef]
47. Hu, J.; Liu, L.; Zhang, C.; He, J.; Hu, C. Hybrid recommendation algorithm based on latent factor model and PersonalRank. *J. Internet Technol.* **2018**, *19*, 919–926.
48. Gatzioura, A.; Vinagre, J.; Jorge, A.M.; Sanchez-Marre, M. A hybrid recommender system for improving automatic playlist continuation. *IEEE Trans. Knowl. Data Eng.* **2019**, *33*, 1819–1830. [CrossRef]
49. Guo, G.; Zhang, J.; Sun, Z.; Yorke-Smith, N. LibRec: A Java Library for Recommender Systems. In Proceedings of the UMAP Workshops, Dublin, Ireland, 29 June–3 July 2015; Volume 1388.
50. Bello-Tomás, J.J.; González-Calero, P.A.; Díaz-Agudo, B. Jcolibri: An object-oriented framework for building cbr systems. In Proceedings of the European Conference on Case-Based Reasoning, Madrid, Spain, 30 August–2 September 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 32–46.
51. Pande, C. Benchmarking Recommender Algorithms for Business Intelligence Consultancy. Master's Thesis, FHNW University of Applied Sciences and Arts Northwestern Switzerland, Olten, Switzerland, 2019.
52. Zhang, Y.; Zuo, W.; Shi, Z.; Yue, L.; Liang, S. Social Bayesian Personal Ranking for Missing Data in Implicit Feedback Recommendation. In Proceedings of the International Conference on Knowledge Science, Engineering and Management, Changchun, China, 17–19 August 2018; Springer: Cham, Switzerland, 2018; pp. 299–310.
53. White, S.; Smyth, P. Algorithms for Estimating Relative Importance in Networks. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–27 August 2003; pp. 266–275.
54. Bergmann, R. On the Use of Taxonomies for Representing Case Features and Local Similarity Measures. In Proceedings of the 6th German Workshop on CBR, Berlin, Germany, 6–8 March 1998; Gierl, L., Lenz, M., Eds.; Springer: Berlin/Heidelberg, Germany, 1998.

55. Huang, A. Similarity measures for text document clustering. In Proceedings of the Sixth New Zealand Computer Science Research Student Conference (NZCSRSC2008), Christchurch, New Zealand, 14–18 April 2008; Volume 4, pp. 9–56.
56. Richter, M.M. Introduction. In *Case-Based Reasoning Technology SE-1*; Lecture Notes in Computer Science; Lenz, M., Burkhard, H.D., Bartsch-Spörl, B., Wess, S., Eds.; Springer: Berlin/Heidelberg, Germany, 1998; Volume 1400, pp. 1–15.