Towards a Text-based Recommender System for Data Mining Method Selection

Completed Research

Patrick Zschech
TU Dresden
patrick.zschech@tu-dresden.de

Kai Heinrich
TU Dresden
kai.heinrich@tu-dresden.de

Richard Horn
TU Dresden
richard.horn@mailbox.tu-dresden.de

Daniel Höschele
TU Dresden
daniel.hoeschele@mailbox.tu-dresden.de

Abstract

The task of mapping a domain-specific problem space to an adequate set of data mining (DM) methods is a crucial step in data science projects. While there have been several efforts for automated method selection in general, only few approaches consider the particularities of problem contexts expressed in domain-specific language. Therefore, we propose the concept of a text-based recommender system (TBRS) which takes problem descriptions articulated in domain language as inputs and then recommends the best suitable class of DM methods. Following a design science research methodology, the current focus is on the initial steps of motivating the problem and conducting a requirements analysis. In particular, we outline the problem setting using an exemplary scenario from industrial practice and derive requirements towards an adequate solution artifact. Subsequently, we discuss potential TBRS methods with regard to requirement fulfillment while organizing both methods and requirements in a structured framework. Finally, we conclude the paper, discuss limitations and draw an outlook.

Keywords

Intelligent Information System, Recommendation System, Automated Method Selection, Data Mining, Text Mining, Natural Language Processing, Domain Knowledge, Design Science Research.

Introduction

Data science and analytics (DSA) projects are usually multidisciplinary in their nature and therefore require combined expertise from multiple fields, such as solid domain knowledge about the problem of interest, analytical modeling skills and experiences with the acquisition and preparation of data assets generated by different IT systems (Mikalef et al. 2018; Zschech 2018). For this reason, there have been several initiatives to support the execution of data-driven projects in a step-wise manner, such as CRISP-DM (cross-industry standard process for data mining), by providing instructions for all relevant tasks from data pre-processing to analytical method selection and evaluation with the purpose to give guidance and structure the overall implementation process (Kurgan & Musilek 2006). At the same time, leading software providers offer more and more standardized functionalities along this process-oriented view within their DSA platforms like SAS, SPSS, RapidMiner or KNIME to make it easier for users with different background knowledge to carry out DSA projects in a systematic and repeatable way (Serban et al. 2013).

However, despite existing tool support, one critical step often remains a challenging task throughout the DSA process, in which the multidisciplinary character shows its remarkable impact, while no tool offers adequate support so far. This step concerns the particular mapping between i) the problem space expressed by concepts and language elements of the domain-specific problem environment, and ii) the class of domain-independent data mining (DM) methods providing an algorithmic solution for data-driven decision support (Choiński & Chudziak 2009; Eckert & Ehmke 2017). In this step, a translation is required that defines the character of the remaining DSA process and thus determines the success of the overall
project (Hogl 2003). Usually this step is carried out by well qualified data scientists, who bring the necessary skillset to combine both contexts, i.e., the methodical skills along a typical data lifecycle and the required business understanding to comprehend underlying domain characteristics and the desired output towards economic goals (Debortoli et al. 2014; Schumann et al. 2016). In practice, however, such fully equipped data scientists are often still a rare species (Mikalef et al. 2018; Zschech et al. 2018) and thus, the mapping task often remains to be carried out by multiple stakeholders in terms of modeling experts and domain experts, making the DSA process an iterative and time-consuming endeavor.

Against this background, we aim to investigate the design of an adequate information system (IS) that is able to support this type of mapping problem. While there have been many efforts for the automated selection of DM methods in general, only a few of the existing approaches take into account the particularities of the problem context expressed in a domain-specific language; which could help domain experts to stay in their familiar surrounding without the need of acquiring deeper DSA knowledge for starting to implement DSA projects (Eckert & Ehmeke 2017; Hogl 2003). In particular, we aim to address this issue by proposing a text-based recommender system for DM method selection (TBRS-DMMS), which takes problem descriptions articulated in natural domain language as inputs and then recommends not only the best suitable class of DM methods but also extracts semantically necessary entities and translates them into the necessary data scheme for the analytical processing.

Following a design science research (DSR) methodology (Gregor & Hevner 2013), it is intended to learn iteratively about the nature of the problem space and thus derive prescriptive knowledge about the resulting design principles of a TBRS-DMMS. Therefore, the paper at hand can be considered as part of a bigger research project, where we currently start with the initial DSR steps of motivating the problem and conducting a fundamental requirements analysis. The remainder of this paper is organized as follows: In the next section, we provide a brief overview of related work and the conceptual background, before outlining our research method in detail. We then continue to define the problem space using an exemplary scenario and extract requirements towards an adequate solution approach. Subsequently, we discuss several recommender methods with regard to requirement fulfillment and finally, conclude the paper by summarizing our findings, discussing limitations, and deriving an outlook for subsequent steps.

Related Work and Conceptual Background

The problem of automated method selection has received increasing attention in recent years in the DM and machine learning (ML) communities. In several survey papers, prior work is summarized from slightly different perspectives, such as meta-learning (e.g., Lemke et al. 2015), per-instance algorithm selection (e.g., Kerschke et al. 2018) or intelligent assistants for data analysis (e.g., Serban et al. 2013). However, the majority of related studies is either limited to a specific type of DM method or it does not consider the particularities of problem contexts expressed in a domain-specific language. For example, Daba et al. (2018) and Danubianu (2008) propose rule-based recommendation frameworks in which DM methods are determined by a variety of selection criteria, such as the number and type of data observations or the handling of missing values. While the first approach is particularly limited to classification problems, the latter allows a broader scope by considering additional methods like cluster analysis, deviation detection and association rule mining. Nevertheless, both approaches stay at a generic level and do not incorporate natural language input as they only rely on features derived from method and dataset characteristics. The approach of Vainshtein et al. (2018), on the other hand, also includes textual features based on word-embeddings from problem descriptions obtained through a large number of academic publications, but is still limited to classification problems. Another notable approach was proposed by Hogl (2003). The author developed a knowledge-based system in which domain experts communicate with the system in natural language via questions and answers (Q&A) and directly receive DM results derived from connected databases. The actual selection of DM methods is done “invisibly” in the background depending on the type and content of the questions. On the downside, however, the approach requires the modeling of a complex and sophisticated knowledge base, including predefined Q&A elements as well as a strong formalization of conceptual and methodical knowledge. For this purpose, we pursue the idea of a more flexible approach using a recommender system (RS) trained on the basis of textual problem descriptions.

In general, RSs support users in finding and selecting suitable items for a particular problem of interest by retrieving the most relevant information from a huge amount of data to avoid information overload. Therefore, they have already been applied successfully in numerous areas like e-commerce, e-learning or e-
tourism (Lu et al. 2015). The type of RS to be developed primarily depends on the availability of data, where either information about the item, the user or the item-user interaction can be used to derive recommendations. These data can take all conceivable types and formats such as structured attributes, images, videos or even texts (Bobadilla et al. 2013). Moreover, each type of data is processed with specific recommender methods, while there are often alternative technologies to choose from with different functionalities and limitations. In the current case of textual contents, for example, multiple approaches have been evolved over time in sub-areas like natural language processing (NLP), artificial intelligence or semantic computing. For this reason, it is important to carry out a precise investigation of the problem setting in order to derive requirements for the specific recommender methods to be implemented.

**Research Method**

To carry out our research, we apply a DSR approach. Design science is a fundamental paradigm in IS research as it is concerned with the construction of socio-technical artifacts to solve organizational problems and derive prescriptive design knowledge (Gregor & Hevner 2013). As such, it is intended to learn about the nature of the problem space and systematically capture design principles and features for a TBRS-DMMS considering multiple facets, such as the design of processing pipelines, the specification and combinability of recommender methods, the shape of input data or the type and degree of user interaction.

In particular, we follow the DSR methodology proposed by Peffers et al. (2007) consisting of the six process steps 1) problem identification and motivation, 2) definition of the objectives for a solution, 3) design and development, 4) demonstration, 5) evaluation, and 6) communication. Focusing in the current paper on the first three steps, we start with a detailed description of the mapping problem in DSA projects (step 1), using an illustrative example scenario based on experiences obtained from industrial practice. Simultaneously, the problem is translated into requirements for a solution artifact in terms of a supporting RS (step 2). Afterwards, we set up the preliminaries for designing the artifact (step 3) by identifying different methods that are suitable for text-based recommendations to compare them with the previously derived requirements. Thus, the result of this paper is an overview of functionalities and limitations of different recommender methods or components, which can be used to establish adequate processing pipelines of a TBRS-DMMS that are either built directly on the investigated approaches or combine them in an integrated manner to support the sketched DSA mapping problem. However, the actual step of designing the artifact is subject to further research together with the remaining steps 4-6.

For the identification of suitable text-based recommender methods, we conducted a literature review (Webster & Watson 2002) using the databases ScienceDirect, IEEE Explore and ArXiv as well as Google Scholar as academic search engine. Specifically, we applied a hermeneutic approach consisting of two intertwined phases – i) a search and acquisition circle and ii) an analysis and interpretation circle (Boell & Cecez-Kecmanovic 2014). For the initial search, we started with two combined keyword groups, where the first group included synonyms for RSs (e.g., recommender, recommendation system) and the second group focused on textual data (e.g., natural language, linguistic). Subsequently, after screening and analyzing initially retrieved articles, it was possible, on the one hand, to identify some key contributions dealing with suitable methods in the given context, and on the other hand, to determine further concepts that led to the creation of new search terms to expand the scope of the literature review. Thus, as a result of conducting several iterations until the identification of previously unconsidered concepts appeared to be exhausted, a large number of diverse approaches could be identified, ranging from general umbrella concepts (e.g., NLP, ML, ontology learning) to more specific method classes (e.g., text classification, topic modeling) to concrete models and algorithms (e.g., latent Dirichlet allocation, TextRank). Although many of these approaches partially overlap and extend across different levels of abstraction, it was nevertheless possible to identify six superior groups of methods towards the implementation of an adequate solution artifact. Therefore, they were subject to a more detailed investigation in Section 5 with respect to the requirements derived from the problem definition.

**Problem Description and Extraction of Requirements**

To define the particularities of the DSA mapping problem, we describe the typical initiation of a DSA project as obtained in industrial practice. Usually, it starts with a certain business case where data-driven decision support is desired (e.g., configuration management of machines). The domain experts (e.g., technical
engineers) with their respective understanding about the setting of interest describe the problem at hand (e.g., machine operations) in their own domain-specific language and provide the relevant context. Then, the modeling experts (e.g., either internal specialists or external advisors) are consulted to abstract the specific problem instance to a more generic class in order to realize a mapping with a certain class of DM methods that could possibly address the issue. Thereby, the modeling experts organize the problem space in a way that not only the output of the DM method is specified but also the input of the DM method. The input specification is necessary for the communication with a third party involved, i.e., the IT system experts, who are responsible for providing all relevant data assets in their required type and structure from the IT source systems (e.g., manufacturing execution systems). Consequently, the DSA project in its initial phase can be described in terms of domain entities of interest, the DM method to be implemented and the data assets to be used to create a blueprint for project realization (Zschech 2018).

To assist this type of mapping problem in an automated manner, the intended solution artifact should provide the following recommender functionality as depicted in Figure 1. It receives the textual problem description from the domain experts and recognizes all relevant entities and relationships of interest. The domain problem is then translated into a more abstract/generic problem class that can be mapped to a certain class of DM methods (e.g., classification, cluster analysis, association rule mining, etc.). The mapping itself is based on a textual knowledge base consisting of previous problem descriptions as well as generic DM method descriptions with characterizing attributes, from which the TBRS-DMMS is able to infer which class of methods is most likely to address the articulated problem. As a result, a recommendation is derived by determining the DM method with the highest degree of suitability. Moreover, the artifact provides two further results to enrich the context of the recommendation. First, the entities identified in the problem description are converted into an exemplary data scheme in which the input is required for the algorithmic processing of the recommended DM method (e.g., machine units as row elements and configuration variables as column elements within a relational table scheme). This helps to simplify the identification of relevant datasets and data structures when communicating with IT experts. Second, an exemplary output of the recommended method is visualized in a comprehensive manner (e.g., configuration profiles as distinct groups derived from cluster analysis), so that the domain experts can gain a quick overview about the method’s basic functionality to assess its suitability for the given problem.

![Figure 1. Intended functionality of a TBRS-DMMS to assist DM mapping task](image_url)

Based on the problem description and the intended functionality of a TBRS-DMMS, several requirements can be derived towards the development of an adequate solution artifact (cf. Table 1). First, the analysis and the mapping of a domain-specific problem space, as introduced in the current context, is based on descriptions in natural language (preferably in English). Thus, it requires the artifact to be able to process unstructured data in the form of texts (R1). Second, as natural language data usually exhibit a high degree of complexity and richness in terms of grammatical structures, ambiguous word meanings, additional filling words or irrelevant context supplements, an adequate solution artifact should be capable of removing any noise from the raw data that is irrelevant to the mapping task and thus may lead to distortions (R2). Third,
in addition to noise reduction, the artifact should also be able to detect semantics and extract central constructs from textual elements instead of solely treating them as pure character strings (R3). This may include, for example, the identification of important key words and phrases that signalize a match or at least the similarity between domain-specific problem descriptions and generic DM method descriptions. Moreover, this could involve the semantic recognition of domain-specific entities and their relationship to each other as well as their abstraction into more generic entities at a higher meta-level to realize the artifact’s functionality of generating input and output specification, as illustrated on the righthand side of Figure 1. Fourth, another requirement concerns the limitation of available data for the establishment and the application of an adequate knowledge base. While method descriptions might be less challenging to obtain (e.g., in the form of definitions from academic literature), problem descriptions from industrial practice are only sparsely available, as companies usually do not store such information in central repositories. In addition, data availability does not only refer to the number of text bodies, but also to the vocabulary used in it to express a corresponding range of terminology. Thus, the artifact must be capable of operating on a small amount of data while ensuring enough robustness for method recommendations (R4).

Fifth, given the fact that there is a broad variety of DM methods like classification, regression, cluster analysis, factor analysis, association rule mining, etc. (e.g., Hogl 2003), the artifact should be able to automatically select and propose those methods from a given set of alternatives where the degree of equivalence with the articulated problem description is the highest (R5).

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1: Dealing with text data</td>
<td>Requirement to process unstructured text data expressed in natural language</td>
</tr>
<tr>
<td>R2: Filtering relevant information</td>
<td>Requirement to remove any noise from the raw data that is irrelevant to the mapping task</td>
</tr>
<tr>
<td>R3: Extraction of semantics</td>
<td>Requirement to detect semantics and extract central constructs from textual elements</td>
</tr>
<tr>
<td>R4: Operation on small amount of data</td>
<td>Requirement to operate on a small amount of data while ensuring enough robustness</td>
</tr>
<tr>
<td>R5: Automated method selection</td>
<td>Requirement to select a DM method from a set of multiple alternatives</td>
</tr>
</tbody>
</table>

Table 1. Requirements for the development of a TBRS-DMMS solution artifact

Requirements Matching with Text-Based Recommender Methods

In the following section, we discuss and assess the identified text-based recommender methods (cf. Section 3) with regard to fulfilment of the requirements. For this purpose, both the requirements and the methods were organized into groups to reduce complexity and ensure a better comparability, since not every single method is suitable to be considered for each requirement due to their heterogeneous nature and different application emphases. Hence, the broad amount of methods were consolidated into six partially overlapping groups, covering methodical approaches for i) text preprocessing, ii) vector space modeling, iii) keyword extraction, iv) topic modeling, v) text classification, and vi) deep learning.

The requirements, on the other hand, were organized into different modules, which may also be interpreted as architectural layers within a TBRS-DMMS design. The first module describes functionalities for preparing any kind of textual input for further processing (R1), where raw data are modified, transformed, filtered and cleared of potential noise (R2). A second module then describes the creation the knowledge base as introduced in the previous section, where pure text strings (R1) are converted into semantically enriched model elements (R3) to realize a mapping between problem descriptions and DM methods based on a limited amount of available data (R4). Finally, a third module describes the application of the knowledge base to derive a recommendation. For this purpose, a limited extent of model elements (R4) from the knowledge base is used to determine which DM method among all modelled alternatives is most suitable for the problem description at hand (R5). Please note, though, that a possible solution artifact not necessarily has to be designed in this particular form, neither do the modules have to be arranged in a strict sequential order. The current structure is rather intended to provide an overview in which areas the identified recommender methods have their emphases as well as strengths and weaknesses, while concrete design choices such as architectural components are subject of subsequent work.

Table 2 depicts the modular structure with the assigned requirements depending on their relevance in each module, while the grey boxes indicate a mapping of the method groups to the different modules. In the following paragraphs, each group will be presented and discussed in detail by referring to selected examples of representative techniques, algorithms and models with respect to a requirement fulfilment.
Text preprocessing. The first group of methods can be summarized under the area of general text preprocessing. While methods in this category cannot directly be used for recommendation purposes, they are often a fundamental component in existing TBRS to prepare natural language data for subsequent model creation and application steps. Frequently used methods include, for example, tokenization (i.e., segmentation of running text into words and sentences), stop word removal (i.e., elimination of words without important connotations, such as “the”, “in” or “yet”), part-of-speech (POS) tagging (i.e., classification of words into categories with similar grammatical properties, such as nouns, verbs or adjectives), parsing (i.e., categorization of words with respect to their position and function within a sentence, such as subjects, objects or predicates), lemmatization (i.e., reduction of inflected forms of a word to a single form, such as reducing “going”, “goes” and “went” to the lemma “go”) and stemming (i.e., reduction of inflected forms of a word to the root form, such as reducing “cats”, “catlike”, “catty” to the word stem “cat”) (Jurafsky & Martin 2018). All these approaches have in common, that they can be applied to transform raw texts into a more accessible form (R1) while eliminating irrelevant information (R2).

Vector space modeling. For the second group of methods, the vector space model by Salton et al. (1975) is to be mentioned, where several approaches exist to transform raw text into vector representations to obtain a more processable form of natural language data (R1). In general, they can be divided into i) classical models such as one-hot encoding and bag-of-words and ii) distributed representations such as proposed by Mikolov et al. (2013). Due to the sparsity of one-hot encoding and bag-of-words models, distributed representations are most widely used nowadays. Furthermore, these approaches include semantics by building vector models based on the words' context and can be extended to create so-called paragraph embeddings that include semantics of whole sentences or documents (R3) (Perone et al. 2018). In this way, the semantics of different DM methods could be projected into the vector space so that they could be used as characteristic features in the next step of applying the knowledge base. However, there is a detriment of these methods and models concerning the amount of required data (R4). Generally, it can be stated that the more data available, the better the quality of embeddings can get due to the higher availability of varying contexts. And even if there is the possibility to narrow down the scope and limit the application to a single domain by creating specific embeddings that require substantially less data, such as demonstrated by Garcia-Silva & Gomez-Perez (2018), there is still the necessity to obtain sufficient data that captures the semantics of this particular domain.

Keyword extraction. A third group of methods can be listed under the umbrella of keyword extraction. In this group, algorithms like TextRank (Mihalcea & Tarau 2004), RAKE (rapid automatic keyword extraction, Rose et al. 2010) or TF-IDF (term frequency-inverse document frequency, Salton & Buckley 1988) are used to automatically identify key terms, phrases or segments that best describe the subject of a document. This is mostly realized by statistical, structural or syntactic scoring or based on co-occurrences and structured graphs of words, which allow their application on texts (R1) (Hasan & Ng 2014). As a result, the extracted sets of words define characteristic elements and properties that can be included in a knowledge base for the abstraction of problem statements and DM method descriptions (R3). Moreover, even if there are more advanced methods that combine different approaches for model building purposes, such as those summarized under the group of topic modeling (Sterckx et al. 2015), the requirement of dealing with a small amount of data (R4) is still given for most keyword extraction algorithms. This is due to the basic principle that with the reduction of the corpus size, also the number of keyword candidates to be evaluated decreases and thus the task of extracting the most important ones becomes less challenging.

### Table 2. Overview of method-requirement matching

<table>
<thead>
<tr>
<th>Method Group</th>
<th>Input Preparation (Raw Text → Prepared Text)</th>
<th>Creation of Knowledge Base (KB) (Text → Model)</th>
<th>Application of KB (Model → Recommendation)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R1</td>
<td>R2</td>
<td>R3</td>
</tr>
<tr>
<td>Text Preprocessing</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Vector Space Modeling</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Keyword Extraction</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Topic Modeling</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Text Classification</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Deep Learning</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
**Topic modeling.** The fourth group of methods is concerned with the concept of topic modeling, which aims at discovering hidden semantic structures from a collection of documents in the form of abstract topics, i.e. related groups of words that best represent the information in the collection. Thus, there are some functional similarities to the group of keyword extraction. The main difference, however, is that topic models identify recurring patterns of words based on probabilistic models. For this purpose, approaches like probabilistic latent semantic analysis (Hofmann 1999) or latent Dirichlet allocation (Blei et al. 2003) extract i) probability distributions over words that define a topic and ii) probability distributions over topics that define a document (R1). Following this principle, it is possible to use topic models with their respective distributions for building a knowledge base that stores topic definitions as semantically relevant constructs (R3). Simultaneously, based on the assumption that a topic model is able to capture the characteristic attributes of different DM methods, the opportunity arises to apply the knowledge base and select the best suitable DM method by choosing the most present topic in unseen problem definitions (R5). However, this leads to the question of the required data availability to receive representative topics. Overall, for training a topic model, it requires a corpus that is big enough to contain all the terms and information that describe the different semantic constructs that should be captured within the problem descriptions. Following this argument, the amount of data depends on the application of the model and should be big enough to counteract variability and ambiguity of natural language. In the current case, for example, the aim is to extract the characteristic attributes of DM methods on the basis of definitions derived from academic literature. As such, a broad database can be assumed considering the amount of documents; however, the vocabulary used in these documents is rather limited, as many definitions are very likely to apply similar terminologies, which do not necessarily reoccur in this form within domain-specific problem descriptions. This leads to a lack of generalization, where a small amount of data becomes a limiting factor when using topic models to apply a respective knowledge base in order to draw a connection between both sides.

**Text classification.** The fifth group consists of algorithms/models for text classification, which generally pursue the goal in TRBS to automatically assign text documents to one or more predefined categories (R5). Frequently used examples are support vector machine and naive Bayes classifier (Garcia-Silva & Gomez-Perez 2018), but there are also many other suitable approaches for this task, such as neural networks, logistic regression, k-nearest neighbors, decision trees, random forests or adaptive boosting (Bishop 2006). All these models have in common that they do not directly operate on raw text bodies, but rather require more structured and better defined inputs created in a preliminary stage, called feature engineering. In the current context, such a preliminary stage could also be interpreted as a kind of knowledge base which a classifier relies on in order to generate recommendations for selecting the appropriate DM method. Possible features for this purpose could include outcomes derived from previously discussed methods, such as topic models, word and paragraph embeddings, keyword-based features like TF-IDF vectors or other text features like frequency distribution of POS tags (Bansal 2018). Once a feature space is established, classifiers are trained in a supervised manner, i.e., a training set of correctly labelled instances is available and the models learn a relationship to discriminate all categories from each other based on the given features. Thus, it can be stated that the requirement to work with a small amount of data (R4) is rather a hurdle for the prior stage of feature engineering and therefore does not pose a particular limitation for the actual classifiers in the sense of applying the knowledge base.

**Deep learning.** The sixth group of methods summarizes approaches based on neural networks that consist of multiple, hierarchically organized processing layers, which is why they are often referred to as deep learning (DL). Their multi-layered architecture allows them to be feed with raw input data and then automatically discover internal representations at different levels of abstraction that are needed for detection and classification tasks. Higher representation layers then amplify input properties that are important for discrimination purposes while irrelevant variations are suppressed (LeCun et al. 2015). For this reason, DL models can be used for both the creation and the application of a respective knowledge base. On the one hand, they work similarly to traditional “shallow” neural networks to classify text bodies to predefined categories (R5), while on the other hand, they are able to automatically learn features from text without additional feature engineering (R1, R3). There are multiple architectures available for this task, such as deep feedforward networks, convolutional networks or recurrent networks (Goodfellow et al. 2016). However, having the scope on textual data, recurrent architectures are most often the preferred choice due of their ability to handle sequential data. Moreover, deep learning architectures can be used to extract semantic constructs. This can be achieved, for example, with distributed representation models (Mikolov et al. 2013) and similar variants like ELMo (embeddings from language models, Peters et al. 2018).
where the semantics of words are represented in dense vectors that can be stored in a knowledge base (R3). In general, the application of DL methods promises state-of-the-art results in terms of creating distributed representations and using them regarding different downstream tasks (Perone et al. 2018). Nevertheless, those methods require a huge amount of empirical observations due to their deep architectural structure and the resulting number of parameters to be optimized, putting them in contrast to the limitation to operate on a small amount of data (R4).

Additionally to the six categories above, a seventh high-level group of methods could be identified referring to the field of ontology learning (OL). In general, OL aims at extracting terms, concepts, relations and axioms from texts to build ontologies in a (semi-)automated manner. Therefore, it draws on a combination of methods that can be classified into statistics-based, linguistics-based and logic-based approaches. However, while the latter group is only rarely used, since its methods primarily focus on constructing heavyweight ontologies, the remaining two groups rely on already discussed methods from the fields of DM, ML and NLP, such as latent semantic analysis, co-occurrence analysis, TF-IDF, POS tagging or sentence parsing (Wong et al. 2012). For this reason, they were not explicitly taken into account again.

**Conclusion, Discussion and Outlook**

The goal of the current paper was to propose the concept of a TBRS-DMMS that supports the automated mapping and selection of suitable DM methods during the initiation of DSA projects, starting from problem descriptions expressed in domain-specific language. Therefore, we first drew a connection to related work and outlined how our approach is distinct to existing ones in established fields like meta-learning or other intelligent assistant systems. Subsequently, we described the particularities of the DSA mapping problem, highlighted intended functionalities of an automated support system and derived requirements towards the development of such a solution artifact. Moreover, we identified and discussed potential TBRS methods with regard to requirement fulfillment while organizing both dimensions in a structured framework for better comparability and a compact assessment. The results show that there is a broad set of methods available, which could be consolidated into six superior groups. It could be seen that they provide different partial support to address the individual requirements, which we organized into separate modules along a possible TBRS-DMMS architecture. While some methods only cover delimited areas, such as classifiers for the discrimination of alternative DM methods or pre-processing techniques for the preparation of raw text data, other methods cover even broader functionalities, such as deep learning models that can combine text classification and internal feature extraction within a single approach.

In the next step, these findings will serve as a crucial basis for subsequent design and development activities within the overall DSR methodology. Using an iterative development procedure, prototypical solutions will be implemented consisting of varying processing pipelines in order to verify the different identified methods and method combinations and examine their effective applicability for the particular mapping problem. Thereby, the proposed architectural division into a tripartite modular structure is currently only to be understood as a rough and generic orientation and can deviate from the presented version depending on the concrete implementations. It is certain, however, that a corresponding textual knowledge base is to be built as the core component of the TBRS-DMMS and that the underlying database will be the decisive factor for corresponding design options. Ideally, a knowledge base would consist of a large set of already labeled problem descriptions, where suitable DM method are already correctly assigned by experts or historical experiences and approaches like deep learning are used to find relevant internal structures to build a model that is able to correctly classify new problem instances. However, for this purpose, a tremendous amount of observations is necessary, which is not easily accessible. For this reason, it is not yet definite to say which concrete form this database will take and therefore, we are currently working on the collection, creation and preparation of a suitable database, where various design aspects have to be considered. This includes, for example, the decision whether a knowledge base should primarily be built on i) domain-specific problem descriptions from real world cases, ii) artificially created problem descriptions, iii) universal DM method descriptions, which could be gathered, for example, from generic DSA literature, iv) DM method descriptions embedded in specific domain-specific contexts, or v) a hybrid combination of the aforementioned options. Another design aspect could be the limitation of considered problem descriptions to a particular domain, such as industrial maintenance (e.g., Horn & Zschech 2019), healthcare (e.g., Hogl 2003) or agrarian management (e.g., Heinrich et al. 2019), in order to keep the complexity reflected by domain-specific entities manageable.
Moreover, we acknowledge that DSA problems in industry are usually much more complex than illustrated by the simplified scenarios throughout the paper. This includes, for example, that DSA problems are rarely straightforward but rather consist of numerous smaller sub-problems, which do not necessarily have to be expressed as such by means of a problem description based on ambiguous natural language. In addition, it is not mandatory that every DSA problem is addressable with an explicit DM method. Nevertheless, we still believe that a considerable amount of problems in practical environments could be supported with such a recommender tool as long as the corresponding problem descriptions show an appropriate level of granularity. Therefore, we currently start our studies with simplified examples at a high level of abstraction concerning the choice of problem descriptions and DM methods and then incrementally increase complexity. In this way, it is planned to iteratively derive prescriptive design knowledge on how to build an adequate TBRS-DMMS that guarantees the corresponding functionality on different levels.

Despite all limitations and upcoming challenges, we are confident that there is a chance for an automated system to support the introduced DM mapping problem. As shown by our contribution, there are already many promising approaches and methods, which now have been brought together within an appropriate solution artifact. For this purpose, our paper should not only serve as the basis for subsequent research steps, but rather as a visionary foundation for a joint discussion with other researchers and practitioners to draw the attention to a mapping problem that goes beyond the scope of traditional recommender tools, meta-learning approaches and other intelligent assistant systems.

REFERENCES


