Domain-focused dataset discovery for tabular datasets, using easily-available information about the domain

by

Riaas Mokiem

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday, November 9, 2022, at 15:15.

Student number:	1150405	
Project Duration:	February 01, 2022 - November 9, 2022	
Thesis committee:	Geert-Jan Houben	TU Delft, Head of thesis committee
	Christoph Lofi	TU Delft, Supervisor
	Jana Weber	TU Delft

An electronic version of the thesis is available at `http://repository.tudelft.nl`.

TU Delft

Domain-focused dataset discovery for tabular datasets, using easily-available information about the domain

Riaas Mokiem

ABSTRACT

Dataset discovery techniques originally required datasets to have the same domain which made them unsuitable to be used on a larger scale. To avoid this requirement, newer techniques use additional information, aside from the datasets being processed, to better understand the data. They might rely on a knowledge base that describes the meaning of data, or a lexical database that defines meaningful relations between the words contained within the data. The main problem with this approach is that these types of additional information have poor coverage of the data being analyzed.

I propose to use a type of information that I call *dataset domain terms*. These are terms, or data values, that represent the domain of a dataset. I provide a technique that can derive these dataset domain terms automatically from existing datasets which means they are easily available. The problem of poor coverage can also be mitigated by only discovering datasets for the domain represented by these dataset domain terms. I provide a dataset discovery technique that takes this approach with these *dataset domain terms*.

Through an evaluation, I show that these dataset domain terms are sufficiently representative of the domain to be used for dataset discovery. The accuracy of the dataset discovery technique is also shown to be comparable to state-of-the-art dataset discovery techniques, though its precision is lacking.

This makes it highly suitable to filter datasets before other dataset discovery techniques can be performed on them. The data from these filtered datasets should also have a limited range of domains. So subsequent dataset discovery techniques should be less affected by the poor coverage of the additional information they use to understand the data. This allows dataset discovery to be performed on a larger scale.

Contents

Abstract 1
Contents 1
1 Introduction 1
1.1 Terminology 3
1.2 Approach overview 3
2 Related work 3
2.1 Data-driven domain discovery (D4) 3
2.2 Semantic type detection 4
2.3 Additional information sources 4
3 Dataset domain terms 5
3.1 Implementation overview 5
3.2 Input requirements 6
3.3 Design choices 6
4 Dataset discovery 8
4.1 Implementation overview 8
4.2 Calculating similarity scores per dataset 9
4.3 Determine domain-focused datasets 9
4.4 Design choices 10
5 Limitations 12
5.1 Dataset domain terms 12
5.2 Dataset discovery 12
6 Evaluation 13
6.1 Criteria for dataset domain terms 13
6.2 Criteria for dataset discovery 13
6.3 Evaluation data 14
6.4 Evaluation methodology 14
7 Results 15
7.1 Dataset domain terms 15
7.2 Dataset discovery 15
8 Discussion 16
9 Conclusion 16
References 17

1 INTRODUCTION

Any organization nowadays produces data. For example, IMDb[1] produces data about movies, governments produce population data, and hospitals produce healthcare data. Many organizations choose to make this data available to others in the form of datasets. This is reflected in the large and growing amount of datasets available. The data will commonly be structured as sets of tables in these datasets, which can then be called tabular datasets. My research focuses on these tabular datasets which can, for example, be expressed as CSV[2] files, spreadsheets, or relational databases.

[1] https://www.imdb.com
[2] Comma-separated values

Data scientists can use this wealth of data to gain useful insights. For example, they may combine population and healthcare data, along with other datasets, to gain insights into how current healthcare policies influence the health of a population. This in turn can help governments improve these policies. But any task that involves more than one dataset requires first integrating those datasets, i.e. creating a unified view of their data. This in turn requires dataset discovery, which is the process of finding datasets relevant to your needs. This naturally involves comparing datasets to each other.

When comparing datasets, information about their domain, or area of interest, is needed[9, 10]. Without this, it is easy to misinterpret the data values, or terms, contained in these datasets. For example, if we compare two datasets that both contain the term 'Titanic', this might indicate that these datasets have something in common. But if we know that one dataset is about movies and the other about nautical history, it becomes more likely that this term refers to the movie[3] in one dataset and the famous ship[4] in the other. In that case, this term is not a point of commonality for these datasets as it has different meanings.

To avoid this problem, earlier techniques require, often implicitly, that the datasets used in that technique are in the same domain[20]. The domain itself remains unknown so it is unknown whether the datasets are about movies, population data, or nautical history. But knowing that the domain is the same is already enough domain information. If we look back at the example, the term 'Titanic' can indeed be considered a commonality in the datasets if we know that both datasets are in the same domain. Regardless of whether that domain concerns movies or nautical history. The main drawback here is that these datasets had to be manually selected as it requires extensive knowledge about domains to determine whether datasets are in the same domain. This manual selection then becomes the main bottleneck for how many datasets can be compared. A human may still be able to select hundreds of datasets but not millions.

This shows that these techniques have a scalability issue. They work well at small scale, where the datasets can be manually selected. On a larger scale, manually selecting the datasets is no longer feasible. With a large and growing amount of datasets becoming available, we need these techniques to scale well.

Addressing this scalability issue, techniques incorporated additional information to better understand the domain, aside from the datasets they process. This may take the form of a knowledge base, like WikiData[5], that describes many domains[4]. Or it may provide a better understanding of language, by using a natural language model[14]. These newer techniques use this to better understand the domain of the data being analyzed, which helps clarify their intended meaning. For our example with the term 'Titanic', these knowledge bases would indicate that this term can refer to a movie, a ship, and other things. One way the technique can figure out which meaning is intended is to look up domain information for other data around this term. It may find another term 'Inception' in the same column, which the knowledge base indicates as being a movie as well as other things. With both terms possibly referring to movies, it is more likely that this column describes movies. But as acknowledged by Nargesian et al.[14] in their research, these knowledge bases have poor coverage. They are not available for all domains or do not describe each domain fully. This means these techniques cannot understand the domain, or intended meaning, for most of the data being analyzed.

State-of-the-art techniques focus on end-to-end solutions, like Aurum[3] or Nargesian et al.'s research[13]. These techniques combine dataset discovery, data integration, and querying the data which allows them to derive information to understand the data from all parts of this process. This makes more information available, resulting in a better understanding of the data. For example, both Aurum and Nargesian et al.'s research derive information about the domain from the input provided to query the data. But this kind of end-to-end solution has an inherent drawback, i.e. that it is more difficult to improve or replace any part of the whole end-to-end solution as all parts need to work together. It can also increase requirements for performance, especially when all the work needs to be done while a user is waiting for a response to their query.

Using information that is more easily available seems to be beneficial to these end-to-end solutions, despite the information being limited in how well it describes a domain. I think this can be applied to dataset discovery separately as well, using a less descriptive but more-easily-available type of information to understand the data. To that end, I formulate the following research questions.

Research Question 1. *What type of information, that can be used to understand data, can be easily made available?*

Research Question 2. *How can dataset discovery techniques deal with the lack or incompleteness of information used to understand its data?*

For Research Question 1, I propose a type of information that I call *dataset domain terms*. These are terms, i.e. data values, that represent the domain of the data in a dataset. Unlike a knowledge base, these terms do not describe the domain at all, but their presence in a dataset indicates that the dataset focuses on that specific domain. For example, the dataset domain terms for a movie domain might contain terms like 'Titanic' and 'Inception' which are movie titles. But it may also include terms referring to actors or directors, like 'Leonardo DiCaprio' or 'Christopher Nolan'. The terms themselves do not describe the movie domain even though we can implicitly understand them to refer to movie titles, actors, and directors. If a dataset contains many of these terms, it seems likely that it focuses on the movie domain. I provide a technique to generate these dataset domain terms automatically from existing datasets that are representative of a domain. This means they can be easily made available for any domain, as long as such domain-representative datasets can be found.

Even with such an easily available type of information, it would not be feasible for that information to cover all data that is processed during dataset discovery. It would require a list of all possible domains, for which we would still need to manually find domain-representative datasets to generate their dataset domain terms. Even just listing all possible domains is not feasible if you take into account that any domain can be split into multiple subdomains. Previous dataset discovery techniques would take a dataset as a reference and look for other datasets that can likely be integrated with it by looking for commonalities in their data. Those datasets

[3] https://www.imdb.com/title/tt0120338/
[4] https://en.wikipedia.org/wiki/Titanic
[5] https://www.wikidata.org

are likely to be related to the reference dataset. I propose to instead use the information about a single domain, like the dataset domain terms, as a reference to find other datasets that focus on that same domain. This is more restricted in what can be discovered as related datasets from different domains can no longer be found. But it should allow this technique to deal with the limited amount of information that is available to understand the data, which addresses Research Question 2. This can be done with other types of information as well but I provide a dataset discovery technique that takes this approach with dataset domain terms.

1.1 Terminology

Since I use specific terminology in my research, I provide their definitions here.

Definition 1.1. Domain: an area of interest.
(Examples: movies, entertainment, city name)

Definition 1.2. Dataset domain: The domain of an entire dataset.
(Examples: movies, healthcare, FIFA players)

Definition 1.3. Column domain: The domain of a column of data. Also referred to as a semantic type.
(Examples: movie title, hospital name, player position)

Definition 1.4. Term: A single data value for one row and column in a table. Also known as a table cell.

Definition 1.5. Domain-representative dataset: A dataset that primarily contains data about a specific domain.
(See Section 3.2.1 for more details)

Definition 1.6. Dataset domain terms: The *terms* that represent a *dataset domain*, derived from *domain-representative datasets*.

1.2 Approach overview

My approach consists of the following two techniques.

(1) A technique to derive dataset domain terms from 2 or 3 domain-representative datasets.
(2) A dataset discovery technique that finds datasets with a strong focus on a specific domain.

The premise of the first technique is that terms that these datasets have in common would only relate to the dataset domain and not those specific datasets. In recent research, Ota et al.[15] have proposed D4, a system that can determine the domain from datasets in this same way. However, this system finds *column domains* whereas this thesis focuses on *dataset domains*. Each column domain is represented by a list of terms that are representative of that column domain. For example, if a column domain is about movie titles, it might be a list containing the terms 'Titanic' and 'Inception'. I assume that the terms from these column domains can be combined to form the dataset domain terms. For example, the combination of terms referring to movie titles, actor names, and movie director names should together represent the movie domain.

The premise for the second technique, which finds datasets with a strong focus on a specific domain, is that a large portion of these domain-focused datasets would consist of dataset domain terms. So this technique calculates a similarity score based on the number of terms that match the dataset domain terms. Similar scores are then grouped in a way that puts the lowest scores together in a single group. Excluding this group of lowest scores, the remaining scores should represent domain-focused datasets.

My contributions are as follows.

(1) A technique for generating dataset domain terms from domain-representative datasets.
(2) A dataset discovery technique that finds datasets with a strong focus on a specific domain, using dataset domain terms to understand data for that domain.
(3) An evaluation of these techniques.

This thesis is organized as follows. Section 1 introduces the problem that this thesis addresses and the solution that I propose for it. Next, I describe related work in Section 2. Section 3 explains the first technique, to generate dataset domain terms. Section 4 explains the dataset discovery technique to find domain-focused datasets. Section 6 describes the criteria and methodology used to evaluate these techniques. The results are then presented in Section 7, followed by a discussion of these results and a conclusion, in Sections 8 and 9.

2 RELATED WORK

2.1 Data-driven domain discovery (D4)

As described before, my technique for generating dataset domain terms is heavily based on the D4 system created by Ota et al[15]. As such, I give a general overview of this D4 system here but more details are provided in the rest of the thesis as needed.

The authors propose a system that performs data-driven domain discovery, called D4. It discovers groups of terms that should be meaningfully related[8], i.e. column domains. A group of terms is considered meaningfully related if those terms occur in the same set of columns. Figure 1 provides a simplified overview of the D4 system using example data representing the movie domain.

The 4 steps that are shown describe the following.

(1) On the far left of Figure 1, we see the example input. In this example, the input consists of two tables from IMDb and MovieDB that both describe movies. They contain terms like 'Avatar', 'Inception', and 'Titanic' but also 'en' or 'nl', which describe movie titles and the language used in a movie.
(2) In the second step, D4 extracts all terms into a single list. Since the same term may occur multiple times, in different columns, D4 also notes how often each term occurs and which columns they originally occurred in. For example, the term 'Inception' occurs twice in the input tables, in the columns 'original title' and 'movie title'.
(3) This allows D4 to create a signature for each term, in the third step shown in Figure 1. This signature mainly identifies, for each term, all other terms that occur in the same columns. For example, the term 'Inception' occurs in the columns 'original title' and 'movie title'. Those columns also contain the terms 'Avatar' and 'Titanic'. So the signature for the term 'Inception' is created with the terms 'Avatar' and 'Titanic'.

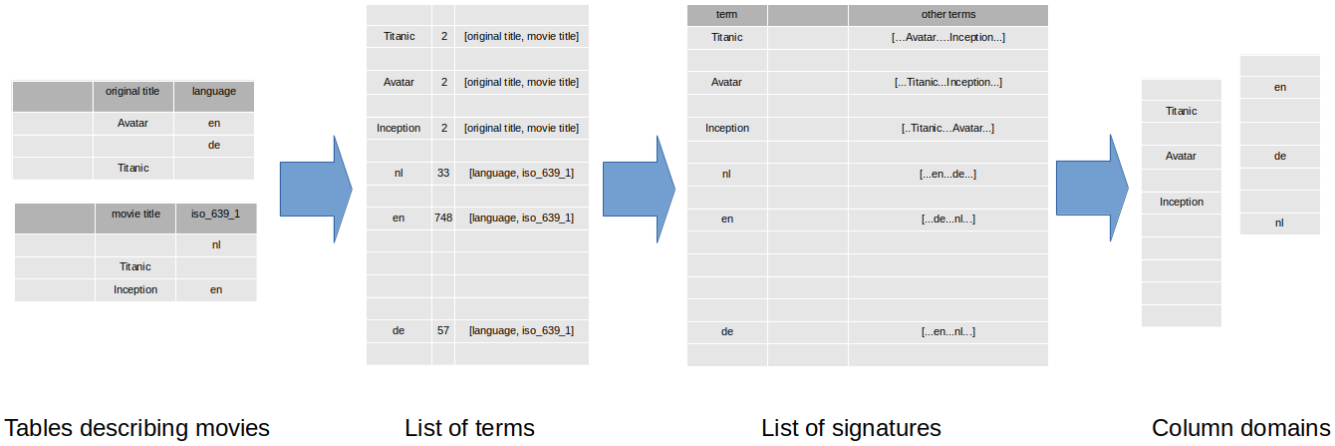

Figure 1: Overview of the D4 system.

(4) The last step of D4 is to derive *column domains* from these terms. D4 uses the signatures to find out which terms originally occurred in the same set of columns. Those terms are considered a *column domain*. For example, the terms 'Avatar', 'Inception', and 'Titanic' belong to the same column domain. This can be easily noticed since each of them has the other two terms in its signature. Looking at the original input tables, we can see that these terms indeed occur in the same two columns.

An interesting observation is that D4 never knows what these column domains mean, nor does it need to. It has grouped the terms 'Avatar', 'Inception', and 'Titanic' without knowing that they represent movie titles. The column names indicate what these terms mean but D4 does not rely on this information.

This is very similar to the *dataset domain terms* that I propose, except that these terms represent the domain of a single column in a table. This is too fine-grained for the dataset domain terms, as these need to represent the domain of an entire dataset which can contain multiple tables. But my first technique aims to turn these *column domains* into *dataset domain terms*.

2.2 Semantic type detection

The research by Ota et al.[15] can be categorized as semantic type detection. A semantic type is a fine-grained description of a real-world concept[6]. The goal is to detect these semantic types from columns in datasets. This means that a semantic type essentially describes a column domain.

In recent research, techniques like Sherlock[6] and Sato[19] apply neural networks to detect semantic types. These techniques are similar to D4 in that they can all take datasets as input. Unlike D4, a significant amount of labeled data is required to train the neural network. This data needs to be labeled with the corresponding semantic type. So these techniques are limited by the availability of training data.

The semantic types that these techniques provide are also less suitable for my research. They are intended to be descriptive, allowing you to understand the meaning, similar to what a knowledge base might provide. But in my research, I need to compare with datasets. From this perspective, the column domains derived by D4 are more suitable as they consist of terms just like any dataset would contain.

2.3 Additional information sources

For better automation, dataset discovery techniques need a better understanding of their data. The datasets they use will, at most, contain descriptive names for the columns and tables. Using only these datasets to perform dataset discovery is only sufficient when the datasets are in the same domain[10].

Inevitably, additional information is needed and much of the recent research focuses on this. Recent surveys, such as the one done by Koutras et al.[7] for their Valentine system or by Ali et al.[1], highlight such dataset discovery techniques which I classify based on the type of information that they use.

I define three categories for dataset discovery techniques based on the type of information used.

(1) Domain information
(2) Linguistic information
(3) Derived information

I explain each category in more detail in the following sections.

2.3.1 Domain information. This type of information describes one or more domains. It is usually called a *knowledge base*. It describes things in the real world according to suitable *ontologies* in a machine-readable format, like RDF[6]. These ontologies may also be used on their own for dataset discovery, instead of a knowledge base.

Several knowledge bases describe common domains, such as WikiData (formerly known as FreeBase), DBpedia[7], and YAGO[18]. For specialized domains, a separate ontology or knowledge base has to be created. For example, there are ontologies related to chemical experimentation[8] or the environment[2].

Dataset discovery techniques use this domain information to understand data as best they can. An earlier technique by Das Sarma et al. [4] searched for relations between their data and multiple

[6] https://www.w3.org/TR/rdf11-concepts/
[7] https://www.dbpedia.org/
[8] https://www.ebi.ac.uk/efo/

knowledge bases. It then determined weights for these relations to each of the knowledge bases based on how well that knowledge base describes the domain for that piece of data. The data are then related to their most suitable descriptions and incorporated into the dataset discovery technique.

Domain information has the benefit of being highly descriptive but its main drawback is its poor coverage[14].

2.3.2 Linguistic information. This type of information describes the relations between words. For example, one word can have the same meaning as another, which is called a synonym. This relation can be defined manually in what is called a *thesaurus*.

Of course, words can have other relations, like antonyms which have the opposite meaning. And other aspects of words may be defined, such as whether it is a verb or a noun. Such a variety of information about words can also be defined manually, in a *lexical database*.

Linguistic information can also be derived automatically. Techniques like Word2Vec[11] and GloVe[16] automatically determine the similarity between words from existing text. This similarity indicates which words are likely from the same domain. This could find similarities between words like 'salary' and 'wage' which are synonyms. But it can also find the words 'kilometer' and 'mile' to be similar. These are not synonyms but they are from the same domain since both represent distance. These automated techniques produce what is called a *pre-trained word embedding* or *probabilistic language model*.

Dataset discovery techniques can use this linguistic information to better understand the words in textual data. The language model would be used to transform words from the textual data into vectors for which the distance between vectors indicates the similarity between those words. This allows easy and meaningful comparisons between those words.

For example, SemProp[5] uses pre-trained word embeddings as linguistic information. It introduces a technique to compare terms containing multiple words to these embeddings for single words. However, this technique also uses a knowledge base as domain information. The word embeddings are only used to match words that are not covered by the knowledge base. This allows the technique to use these pre-trained word embeddings to understand data that is not covered by the knowledge base.

Of course, newer techniques produce models with more accurate and diverse relations between words. This could improve dataset discovery as well, depending on how they are used. However, I have not seen more recent language models used in dataset discovery techniques. Research in this area may have decreased in favor of end-to-end solutions for dataset discovery.

Linguistic information is not suitable for my research as it does not provide much understanding of the domain. Given that I propose a technique to discover datasets for a single domain, it does not make much sense to use linguistic information for this.

2.3.3 Derived information. Acknowledging the cost of creating domain information and language information, some research has shifted its focus to creating and using information that is derived from readily available sources. I categorize this as *derived information*.

Recent research has focused on end-to-end solutions which derive information from user input to better understand the data. The premise is that the user input provides indications of the intended domain of the data the user is looking for. For example, a user that provides the search query 'movie about sinking ship' is likely interested in movies, or nautical disasters. This narrows down the likely domain for the data that is queried and influences which datasets are discovered and how they are integrated. This user input can take many different forms, like the search query[3] or its navigational context[13]. The drawback of these end-to-end solutions is that it becomes more difficult to improve or change any single part of it, as it is all tightly coupled.

This thesis also belongs to this category since it proposes using a new type of information, i.e. *dataset domain terms* that are derived from existing datasets. The aim is to use derived information for dataset discovery separately, avoiding the need for a tightly-coupled end-to-end solution.

3 DATASET DOMAIN TERMS

This technique generates *dataset domain terms* by deriving them from domain-representative datasets. It is heavily based on D4 introduced by Ota et al.[15], with its implementation performing most of the work for this technique.

3.1 Implementation overview

Figure 2 provides an overview of how this technique is implemented. There are only three steps.

(1) All tables from the domain-representative datasets are provided to D4 as input.
(2) D4 turns those tables into *column domains*, as described in Section 2.1.
    (a) Extract terms from the tables.
    (b) Create a signature for each term.
    (c) Derive column domains from these terms, using their signatures.
(3) The most relevant terms from those column domains are combined to form the *dataset domain terms*.

Most of the work for this technique is done by D4. This requires that it is configured appropriately for discovering column domains that can be turned into dataset domain terms. The default configuration options provided by D4 are used, except that the pruning strategy is set to 'conservative'. This configures D4 to only include terms in the column domain if they are highly likely to be relevant to that domain. For the dataset domain terms, it means that the column domains are less likely to contain terms that do not represent the dataset domain.

What this technique adds, aside from what is done by D4, is to combine the terms from the column domains into dataset domain terms. This relies on the following assumption, which I will validate as part of the evaluation.

Assumption 1. *The terms from the column domains derived from domain-representative datasets are together representative of the dataset domain for those domain-representative datasets.*

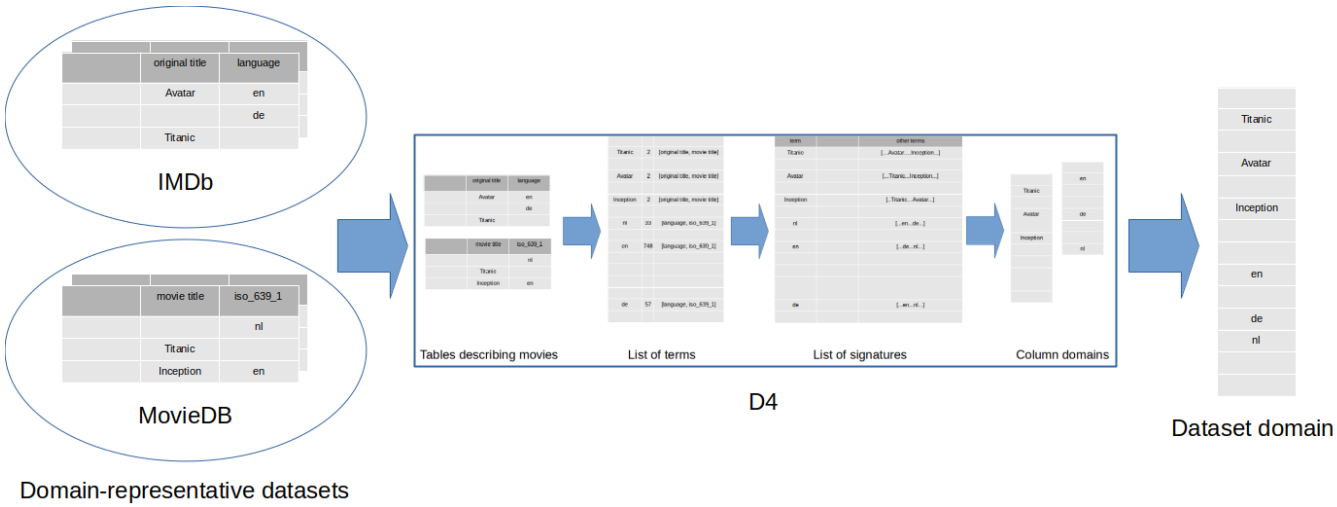

Figure 2: Generating the dataset domain terms.

Only the most relevant terms from each column domain are included in the dataset domain terms. This again reduces the likelihood that the dataset domain terms contain terms that do not represent the dataset domain.

The output of this technique should be a list of terms that are representative of the domain that the domain-representative datasets focus on, i.e. *dataset domain terms*.

3.2 Input requirements

3.2.1 Domain-representative. As indicated before, the datasets used as input for this technique must be domain-representative. Ideally, the data in the dataset *only describes things from a single domain* and *describes everything from that domain*. While this is not feasible for existing datasets, we should try to get as close to this as possible. As such, I formulate the following requirements for domain-representative datasets.

(1) The data in the dataset must primarily be focused on describing the domain.
(2) The dataset must describe the domain as completely as possible.

As an example, we can consider these requirements for the dataset from IMDb. The data in this dataset primarily covers movies and series. Similar organizations, like MovieDB[9] and TheTVDB[10], also describe both movies and series. So this indeed seems to be the domain for these types of organizations.

Now we need to ensure that the IMDb dataset is as complete as possible. This means that it should describe as many movies and series as possible, which means it should have an international focus. IMDb contains movies and series from 261 countries[11] which exceeds the 249 regions defined by the UN[12]. So it seems that IMDb indeed has an international focus which means their dataset should be as complete as is feasible for us to verify.

This example also highlights a problem with finding suitable domain-representative datasets. If you search for a dataset with a domain in mind, it is hard to find one that is suitable. The IMDb dataset may seem the most suitable for a 'movie' domain. But as the example shows, its domain would be better described as 'movie and series information'. And this indeed matches the domain of similar organizations. So when looking for domain-representative datasets, you will likely not find existing datasets that match that domain exactly. But it should be sufficient to find datasets that are close to that domain. Like how the 'movie and series information' domain was found instead of the 'movie' domain. This means you may need to narrow or widen the scope of the domain you are looking for, depending on your needs and the available datasets.

3.2.2 Amount. The reason that this technique requires using more than one domain-representative dataset is to use the commonalities between them to filter out any terms that are specific to a single dataset. As such, I need at least two domain-representative datasets. Increasing the amount further may indeed define the dataset domain terms more clearly. Since I use D4 to look for commonalities between datasets, the dataset domain terms are derived by looking at the pairwise co-occurrence of terms. This means that using more datasets also increases the risk that non-domain terms, like terms from a different domain or dataset-specific terms, are included. These terms might be shared by some pairs of datasets.

So it is recommended that this technique use only two domain-representative datasets as input.

3.3 Design choices

When designing and implementing this technique, several choices needed to be made. Some design choices required experimentation to understand what the best option is. Here, I document these design choices and the experiments to determine the most suitable option.

3.3.1 Using D4. The first design choice is regarding the use of D4 in this technique. The options here are as follows.

[9] https://www.themoviedb.org
[10] https://thetvdb.com/
[11] As shown on its search page: https://www.imdb.com/search/title/
[12] https://unstats.un.org/unsd/methodology/m49/

- Create a new implementation of D4. This would be easier to modify and integrate with my technique but would take a lot of time.
- Use the existing implementation provided by the authors of D4. This would save time but could be more difficult to modify and integrate with my technique.

As shown when describing the technique, D4 should not need to be modified. The implementation provided by the authors also produced files as output and provided sufficient instructions on how it can be used. This seemed sufficient to integrate with my implementation. So I decided to use the implementation that was provided. I use version 0.30.1[13] of the D4 system in my implementation.

3.3.2 D4 strategy. D4 allows for three different strategies to create the column domains.

- Conservative: include only the most relevant terms in the column domain. This optimizes D4 for precision.
- Liberal: include many terms in the column domain, unless they are highly unlikely to be relevant. This optimizes D4 for recall.
- Centrist: This strategy aims to balance precision and recall.

For the dataset domain terms, the most suitable strategy should be to optimize for precision, i.e. the *conservative* strategy. This would reduce the likelihood that terms that do not represent the domain are included in the dataset domain terms.

When performing dataset discovery, this strategy should lead to an improvement in similarity scores. I validated this experimentally by comparing it with the similarity scores produced by following the *centrist* strategy. Figure 3 shows that the 'centrist' variation does not result in better similarity scores than the 'conservative' variation. It shows no difference for movie datasets but increases the similarity score slightly for non-movie datasets. While the difference is minor, an improvement would be the other way around. More details on the experiment are provided in Section 3.3.6.

3.3.3 Column expansion. D4 tries to ensure that the column domains are as complete as possible using a feature called column expansion. This adds terms to the existing columns in the datasets, i.e. expanding them. It looks for new terms that likely also belong in that column, according to their signatures. However, there is a risk that terms are incorrectly added to the existing column, which means it might be better to disable this feature. The choice here is whether this feature should be disabled, given the risk it presents.

I determined experimentally that disabling this feature would be highly detrimental. Figure 3 shows that the 'no-expand' variation resulted in significantly worse similarity scores. The similarity score for movie datasets decreased far more than that of non-movie datasets. So column expansion seems to be useful overall, despite the risk it carries, and should not be disabled.

3.3.4 TF-ICF. When creating signatures, D4 calculates the similarity of the term to other terms from the same columns. This is determined by how many columns those terms have in common and is calculated using a similarity function. In the original paper, D4 uses Jaccard Index as its similarity function. The D4 implementation contains two more similarity functions, one that uses a logarithmic scale for Jaccard Index and another that includes weights for each term. So these are the options for this choice.

- Jaccard Index, as used in the original paper.
- Logarithmic Jaccard Index.
- Weighted Jaccard Index, also called TF-ICF.

Changing the scale of the Jaccard Index to a logarithmic scale is unnecessary for me so the logarithmic Jaccard Index can immediately be rejected as a choice.

The weighted Jaccard Index is based on the well-known TF-IDF[14] statistic used in the field of information retrieval. The basic premise is to weigh the Jaccard Index for each term based on how often it occurs in a document and how rarely it occurs in other documents. TF-ICF applies this to columns instead of documents, which is why it has such a similar name. It stands for *Term Frequency-Inverted Column Frequency*[15]. These weights could reduce the number of generic terms included in the dataset domain terms.

But I determined experimentally, as shown in Figure 3 with the 'tf-icf' variation, that this is detrimental overall. It increased the similarity score for non-movie datasets far more than for movie datasets. So the weighted Jaccard Index, or TF-ICF, does not seem suitable for my technique and I will use the normal Jaccard Index instead.

3.3.5 Column domains terms. The terms in each column domain produced by D4 are grouped according to how relevant they are to that column domain. When combining the terms from these column domains into dataset domain terms, there are three ways that we can use this grouping of terms:

- Include all terms from the column domain.
- Include only the most relevant group of terms from each column domain.
- Include terms with a relevance score above some determined threshold.

Since the dataset domain terms should only contain terms that represent the domain, the most suitable choice seems to be to only include the most relevant group of terms. The inclusion of other terms, known to be less relevant, might include terms that do not represent the domain. The reasoning here is similar to why the *conservative* strategy was chosen for D4.

To validate this choice, I experimented by comparing this against a variation where all terms were included which I called 'all-terms'. I did not experiment with a variation that only includes terms above a threshold as that would introduce unnecessary complexity to determine a threshold value.

With this experimentation, I determined that this choice is indeed correct. Figure 3 shows that the 'all-terms' variant increases the similarity score for non-movie datasets much more than for movie datasets. This means that the 'all-terms' variation results in worse similarity scores. So the better choice is indeed to include only the most relevant group of terms from each column domain in the dataset domain terms.

[13] https://github.com/VIDA-NYU/domain-discovery-d4/releases/tag/0.30.1

[14] Term Frequency-Inverse Document Frequency

[15] This is different from earlier research that introduced the same initialism[17], where it stands for *Term Frequency-Inverse Corpus Frequency*. The research is unrelated to this new similarity function though both are based on TF-IDF.

Figure 3: The average increase in similarity score, using different variations of the dataset domain terms for both movie and non-movie datasets.

	movie dataset	non-movie dataset
conservative	0.00%	0.00%
centrist	0.00%	0.03%
no-expand	-25.92%	-16.78%
tf-icf	1.11%	7.89%
all-terms	0.64%	8.53%

3.3.6 Experiments. For the experiments, I define several variations to validate the choices that were made.

- *conservative*: Uses the conservative pruning strategy in D4 and includes only the top terms from the column domain. Used as the baseline for comparison with the other variations.
- *centrist*: Uses the centrist pruning strategy in D4 and includes only the top terms from the column domain.
- *no-expand*: As '*conservative*', but with column expansion disabled.
- *tf-icf*: As '*conservative*', but with D4 using weighted Jaccard Index, TF-ICF, instead of Jaccard Index.
- *all-terms*: As '*conservative*', but includes all terms from the column domain.

For the design of this technique, the primary focus is on how these variations impact my proposed dataset discovery technique. I used the *IMDb* and *MovieDB* datasets to create dataset domain terms for each variation, each representing the 'movie and series information' domain. I then used each variation with my dataset discovery technique on the same group of datasets, some movie-related and some not.

The datasets are described in more detail in Section 6.3 and Figure 6. For this experiment, I used the following datasets as input. The first four are movie datasets and the remaining five are non-movie datasets.

- Indian Movies
- Movies Dataset
- Netflix
- TMDB 350K+ Movies
- Education
- Finance
- Services
- Steam
- Rotten Tomatoes

I used an intermediate outcome from the dataset discovery technique, the similarity score, to evaluate these variations. I used the 'conservative' variation as the baseline for all other variations, as it represents the choices that are expected to lead to the best performance for the similarity scores. All other variations are based on the 'conservative' variation, with only one part changes for each. This allows measuring whether that changed part would improve the similarity scores. An improvement would mean that *the similarity scores of movie and non-movie datasets would have a larger difference.* This is considered an improvement as it means that the movie and non-movie datasets become easier to distinguish from each other.

The results are presented in Figure 3. It shows that the 'conservative' variation was indeed the most suitable for generating dataset domain terms. This validates the design choices that were made, as already explained in the previous sections.

4 DATASET DISCOVERY

The other technique that I propose is intended for dataset discovery. It discovers datasets with a strong focus on a specific domain, which is represented by *dataset domain terms*. Its main design principle is to look for commonalities between the dataset and the dataset domain terms, as described in Section 4.4.1. This differs from other dataset discovery techniques, which look for commonalities between datasets that indicate that they can likely be integrated.

This difference in design addresses Research Question 2, allowing it to better deal with information being unavailable or incomplete. This information is used to understand the data but may not cover all data that needs to be understood. It refers to the dataset domain terms in this technique. With this different approach, any data that is not covered by the dataset domain terms can be considered to belong to another domain. It can thus be ignored by design.

This is the main improvement over previous dataset discovery techniques. They would still need to judge whether that data should be considered related or not, despite having insufficient information about its domain.

4.1 Implementation overview

This technique consists of the following steps, separated into two main parts. The first part, calculating the similarity score, is repeated for each dataset that is provided as input. The second part is only done once after the similarity scores for all datasets were calculated.

(1) Calculate the similarity score for each dataset.
   (a) Extract terms from the dataset.
   (b) Find matches with the dataset domain terms.
   (c) Calculate the similarity score for the dataset.
(2) Based on the similarity scores for all datasets, determine which datasets have a strong focus on the domain.
   (a) Determine the grouping threshold.
   (b) Group the similarity scores according to this grouping threshold.
   (c) Remove the lowest-scoring group of similarity scores.
   (d) List the datasets corresponding to the remaining similarity scores.

The dataset domain terms should represent the domain of a dataset. So if a dataset contains a sufficient number of terms that match these dataset domain terms, that dataset contains a large number of terms that represent the same domain. It seems fair to consider that dataset to have a strong focus on that domain. This number of terms is used to calculate a similarity score for each dataset such that they can be compared to each other, in the first part of the technique.

Datasets that do not have a strong focus on the domain should have very few terms that match the dataset domain terms. This leads to the following assumption, which the second part of the technique is based on.

Assumption 2. *A domain-focused dataset contains significantly more terms that match the dataset domain terms than a non-domain-focused dataset.*

If this assumption is valid, it means that there should be a significant difference between the similarity scores for domain-focused and non-domain-focused datasets. However, the latter should have the lowest similarity scores and are likely close together. These low scores should mostly reflect accidental or erroneous matches. The domain-focused datasets can have a wider range of similarity scores since datasets may contain more or fewer terms that match the dataset domain terms. So these similarity scores may still differ a lot from each other.

If we sort the similarity scores and look at all the differences between consecutive scores, this should result in mostly 'small' differences and one or more 'larger' ones. It is unknown how small or large these differences are so I just name them as 'small' and 'larger'. This technique tries to split this sorted list whenever one of these 'larger' differences occurs, which is also depicted in Figure 5. To do this, we need to determine a lower-bound value for these 'larger' differences. In other words, we need a threshold that determines when the sorted list of similarity scores should be split.

We can take the halfway point between the largest and smallest difference between similarity scores as the threshold. This would likely be between the 'small' and 'larger' differences. However, if one of the 'larger' differences is significantly larger than the other 'larger' differences, the threshold might become too large. This would mean that the list of similarity scores would be split less often, resulting in fewer groups. As a result, the lowest-scoring group would likely include similarity scores for domain-focused datasets which would in turn not be considered domain-focused by this technique. Though this is unavoidable, it can be mitigated somewhat by taking the halfway point between the 2nd largest and 2nd smallest difference between similarity scores as the threshold.

In the next sections, I explain each of the two main steps in more detail and then discuss the design choices made for this technique.

4.2 Calculating similarity scores per dataset

Figure 4 provides an overview of how terms are matched, with example data.

4.2.1 Extracting terms. For this technique, all terms first have to be extracted from the dataset. This is shown in Figure 4 where the terms from the tables in the 'Netflix' dataset are extracted to a single list. This is the same as the first step performed by the D4 system so this technique reuses the D4 implementation for this.

Duplicate terms are also removed since this technique only needs to know whether a term is present in the dataset. It is irrelevant how often that term is present in the dataset. This is easiest to show with an example.

Take the movie titles 'Titanic' and 'Inception' as dataset domain terms for the movie domain. This technique would consider a dataset that has both these terms more similar to the movie domain than a dataset that only contains the term 'Titanic'. Even if that dataset contains the term 'Titanic' ten times, this does not mean it has a stronger focus on the movie domain. It may have a stronger

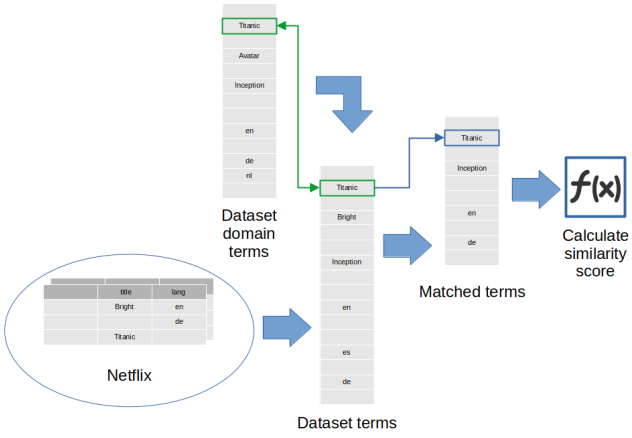

Figure 4: An overview of the first part of the dataset discovery technique; shows how terms are matched between the Netflix dataset and dataset domain terms for the movie domain.

focus on the movie title 'Titanic' but that is irrelevant to this dataset discovery technique.

4.2.2 Finding matches. Each term is then compared to the dataset domain terms. If that term matches one of the dataset domain terms, it is added to the list of matched terms. Figure 4 shows that the term 'Titanic' from the extracted terms matches one of the dataset domain terms. Therefore, it is added to the list of matched terms.

4.2.3 Calculating similarity score. The similarity of a dataset to a domain is based on the number of terms that match between them. According to Assumption 2, domain-focused datasets should have more matched terms than non-domain-focused datasets. That means that the technique needs to compare the number of matches from all datasets to find which ones can be considered similar to the domain. Since the maximum possible number of matched terms can differ for different datasets, the number of matches is not directly comparable. But we can make the number of matched terms comparable by relativizing it to the maximum possible number of matches. This statistic is also known as the overlap coefficient. It is defined in Equation 1 as the *similarity score* for a dataset. This is the calculation that is done in the last step of Figure 4.

$$similarity\ score = \frac{|matched\ terms|}{min(|dataset\ terms|, |dataset\ domain\ terms|)} \quad (1)$$

4.3 Determine domain-focused datasets

4.3.1 Determine threshold. To create groups of similarity scores, this technique first needs to determine a threshold value. This threshold is used to split the list of similarity scores, allowing this technique to group the similarity scores of non-domain-focused datasets.

The first step in Figure 5 shows that we add four artificial similarity scores with values 0.0, 0.0, 0.5, and 1.0. These similarity scores have no relation to the datasets and are entirely artificial. Their purpose is to ensure the threshold can be determined correctly, as

this requires at least two similarity scores that correspond to both domain-focused and non-domain-focused datasets.

Then we can sort the similarity scores in descending order, and calculate the consecutive difference between those scores. Those differences are again sorted in descending order. These three steps are shown in Figure 5, followed by the calculation of the threshold based on the sorted differences in similarity score. The threshold value is the average of the 2^{nd} largest and 2^{nd} smallest difference between consecutive similarity scores.

$$threshold = \frac{2^{\text{nd}}\ largest\ difference + 2^{\text{nd}}\ smallest\ difference}{2} \tag{2}$$

4.3.2 Group similarity scores. The threshold value is used to group the similarity scores, as shown in Figure 5. Starting from the sorted list of similarity scores, they are grouped by separating consecutive scores that have a difference larger than the threshold value. When the difference to the next score is less than the threshold, those scores belong to the same group. When it exceeds the threshold, the list is split which results in separate groups.

4.3.3 Remove lowest-scoring group. After grouping is completed, the artificial similarity scores are removed. The group containing the lowest scores is now removed, as these are considered to correspond to the non-domain-focused datasets. In Figure 5, the similarity scores *0.14*, *0.08*, and *0.05* belong to this group and are removed from the list.

4.3.4 List domain-focused datasets. So the datasets corresponding to the remaining scores are determined to have a strong focus on the domain. In Figure 5, these would be the datasets corresponding to the three highest similarity scores, *0.87*, *0.48*, and *0.41*. In this example, these would correspond to the datasets *Indian movies*, *Netflix*, and *Movies dataset*. These are the domain-focused datasets that have been discovered with this technique.

4.4 Design choices

As with the previous technique, some choices had to be made while designing and implementing this technique. This section documents and explains these choices.

4.4.1 Main design principle. The purpose of a dataset discovery technique is to find related datasets. It will often only accept two datasets as input and compare them directly. This comparison aims to find commonalities in their data that may be used to integrate them. If it seems likely that the datasets can be integrated, they are assumed to be related. With more datasets, this would be repeated for every pairwise combination.

As explained in the introduction, this only works when the datasets are in the same domain. And dataset discovery techniques can use additional information to better understand the domain of the data, like a knowledge base. This allows these techniques to verify if the data are in the same domain. But such additional information does not fully cover all data being analyzed, limiting the usefulness of such techniques. Research Question 2 addresses the need to deal with this incomplete or lacking information.

This is why this dataset discovery technique is designed to work differently from previous techniques. It does not look for commonalities between the datasets. Its main design principle is to *look for commonalities between the dataset and dataset domain terms instead.* In doing so, the technique does not need the dataset domain terms to cover all terms from the datasets. The dataset domain terms only need to cover a single domain as any terms that were not covered can be considered to belong to another domain. These terms can thus be ignored, instead of making inaccurate conclusions about their meaning.

The possibility remains that the dataset domain terms do not represent the complete domain. This means that terms that do represent the domain but are missing from the dataset domain terms would be incorrectly ignored. However, if the domain-representative datasets were chosen well (see Section 3.2.1), the dataset domain terms should be as complete a representation of the domain as any dataset might contain. This should reduce the impact of lacking and incomplete information in this dataset discovery technique.

Given this design principle, this technique outputs a list of datasets with a strong focus on a single domain. By design, it does not find related datasets from different domains as previous dataset discovery techniques would. While this narrows the scope of what kinds of datasets can be discovered, it also allows the use of much less descriptive information to understand the domain, i.e. *dataset domain terms*.

4.4.2 Extracting terms. For this dataset discovery technique, the terms need to be extracted from the dataset. This is the same process that is done in D4 and it is implemented such that only that part can be used. So the options are as follows.

- Reuse part of the D4 implementation that extracts terms.
- Create a new implementation to extract terms.

The benefit of reusing the D4 implementation is to save time while the benefit of a new implementation is that it can be created to better suit this dataset discovery technique. One restriction is that the D4 implementation mainly accepts TSV[16] files as input. With a new implementation, many other formats could be supported. However, I focus only on tabular datasets which can all be converted into TSV format. So this restriction should not pose a problem.

Another benefit of reusing the D4 implementation is that it ensures consistency for the terms. The terms are extracted in the same way as the dataset domain terms. This should ensure that the same data results in the same extracted term. These benefits were sufficient reasons to reuse the D4 implementation for my technique.

4.4.3 Similarity score. The amount of matching terms is the main indicator of how similar a dataset is to the domain represented by the dataset domain terms. However, this amount is not suitable for comparing against the amount for other datasets. So this value needs to be normalized to allow such comparisons.

A common statistic for this is the Jaccard Index, which would normalize the number of matches relative to the amount of all unique terms in both the dataset and the dataset domain terms. Another option would be to normalize the number of matches relative to their maximum possible amount, which is called the

[16] Tab-separated values

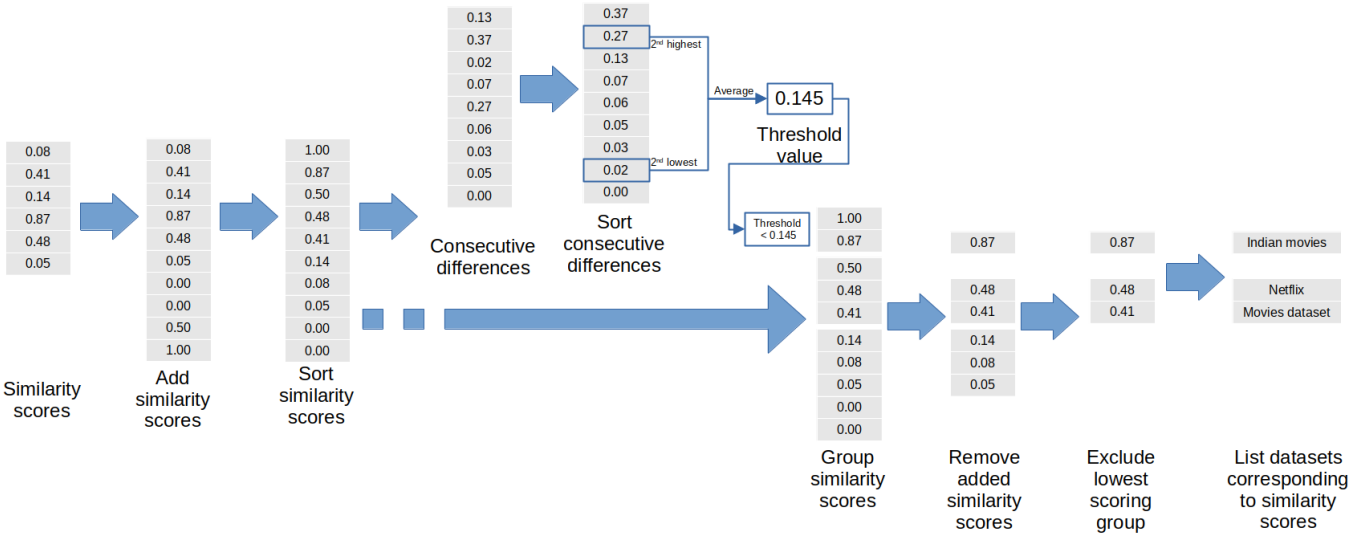

Figure 5: Overview of the second part of the dataset discovery technique, which shows how the threshold is determined and used to group the example similarity scores shown here.

overlap coefficient or Szymkiewicz–Simpson coefficient. These two are the main options.

- Jaccard Index
- Overlap coefficient

With the Jaccard Index, the difference in size between the terms from the dataset and the dataset domain terms can cause problems. For example, if the number of dataset domain terms is five times the number of terms in the dataset. Even if all of the terms from the dataset were to match the dataset domain terms, the score according to the Jaccard Index would still only be 0.2. Such a low score may be hard to distinguish from the low score of a non-domain-focused dataset. The Jaccard Index also incorporates how many terms were not matched, but this is not relevant to this technique. This means that the Jaccard Index is unsuitable for this technique.

The overlap coefficient, on the other hand, normalizes the number of matches relative to the maximum possible amount of matches This means it could achieve a maximum score of 1.0, regardless of the size of the dataset and the dataset domain terms. So I decided to use the overlap coefficient to determine the similarity score of a dataset in this technique.

4.4.4 Identified datasets. The purpose of this dataset discovery technique is to find datasets from the domain represented by the dataset domain terms. There are two ways to achieve this.

- Directly identify datasets that focus on the domain.
- Identify datasets that do not focus on the domain and exclude them, leaving only datasets that focus on the domain.

Directly identifying the datasets we want to find would be the most obvious choice. The problem is that their similarity scores are likely very different. While none of them should be low, within the range of 0.0 to 1.0 it can just as easily be *0.3* as *0.9*. This makes them much harder to identify by their similarity score.

Datasets that do not focus on the domain should all have a low similarity score that should not differ much from each other. The similarity scores should also be significantly lower than domain-focused datasets. This makes these datasets easier to identify, as their similarity scores can be grouped into a single group. So this technique should identify datasets that do not focus on the domain.

4.4.5 Artificial similarity scores. When determining the grouping threshold, four artificial similarity scores are added. This is necessary to determine the threshold correctly since this process is based on the difference in similarity scores of domain-focused and non-domain-focused datasets. This means that similarity scores for both are needed. As this cannot be guaranteed from the datasets provided as input, these artificial similarity scores are added to ensure this.

The values of these artificial similarity scores are chosen to represent idealized similarity scores as well as the difference between them. For non-domain-focused datasets, their ideal similarity score would be 0.0 and they would have as small a difference as possible. So the two artificial similarity scores that represent non-domain-focused datasets have values of 0.0 and 0.0, resulting in a difference of 0.0.

For domain-focused datasets, the ideal similarity score would be 1.0 and we want the difference between similarity scores to be large. We can consider the lowest similarity score which should always correspond only to domain-focused datasets to be 0.5. The similarity score for domain-focused datasets can be much lower than this but non-domain-focused datasets could also achieve those scores. A score of 0.5 would also mean that half of the terms of the dataset match the dataset domain terms, or half of the dataset domain terms matched the dataset. In either case, such a dataset seems like it should be considered to focus on the domain. So the two artificial similarity scores that represent domain-focused datasets have values of 1.0 and 0.5, resulting in a difference of 0.5.

5 LIMITATIONS

During the design and implementation of these two techniques, their limitations have become clear. I document the limitations of each technique separately, in the following sections.

5.1 Dataset domain terms

In this section, I document the limitations of the technique used to derive dataset domain terms from domain-representative datasets.

5.1.1 Text only. This technique uses textual data in the datasets that are provided which is a limitation inherited from D4. The reason for this limitation is that other kinds of data, like numbers or dates and times, give almost no indication of their meaning. For example, the meaning of the number '42' cannot be derived only from this number. It requires more information from the text surrounding it. This may belong to a column with a header 'age', indicating that the number refers to the age of someone or something. It may also be labeled as 'answer', indicating that is the answer to a question. But D4 does not try to derive this meaning from surrounding text so non-textual data is ignored instead.

Limitation 1. *The dataset domain terms are only derived from textual data in the domain-representative datasets used as input.*

5.1.2 Exact term matching. The D4 system used by this technique looks for terms that occur in multiple columns. However, it determines which terms are the same by looking for exact matches. This means that even small differences between terms would cause them to not match. For example, one column might contain the movie title *Home Alone* and the other *Home Alone 1*[17]. These would not be considered the same term though both refer to the title of the same movie.

This means that D4 does not attempt to guess whether terms might be the same. While this ensures that the terms are matched very precisely, the obvious drawback is that it may miss some matches. This can be especially problematic when different datasets have consistent differences in their terms. For example, a dataset might replace or remove certain characters. This could result in terms like *Home_Alone* instead of *Home Alone*, or *AEon_Flux* instead of *Æon Flux*. This could cause a significant amount of terms to incorrectly fail to match. If this amount is significant enough, the resulting dataset domain terms may no longer be representative of the domain.

Limitation 2. *When the domain-representative datasets have a consistent structural difference in their terms, the resulting dataset domain terms may not be representative of the domain.*

5.1.3 Less-representative dataset domain terms. When generating dataset domain terms, the technique tries to filter out dataset-specific terms from the domain-representative datasets. The idea is that terms that occur in both domain-representative datasets should be specific to the domain and not just that dataset.

This is mainly done using D4, which creates column domains based on the co-occurrence of terms. However, D4 looks at terms that occur in different columns. This means that if terms occur in different columns in the same dataset, D4 may include them in the column domains.

[17]This might be done to differentiate it from its sequels

Experimentation showed that this mainly resulted in primary keys from the dataset being included in the dataset domain terms. While these consisted of up to 70% of the dataset domain terms, their specific format meant that the terms from other datasets would be unlikely to match them. So the inclusion of these primary keys should not hurt dataset discovery. But they do make the dataset domain terms less representative of the domain.

This can also occur with other terms, not just primary keys. So this limitation means that the dataset domain terms may be less representative of the domain than they should be.

Limitation 3. *The dataset domain terms can contain many terms that only occur in one of the domain-representative datasets and not both. These terms might not represent the domain which makes the dataset domain terms less representative of the domain.*

5.1.4 No domain description. The dataset domain terms only represent a domain. Unlike knowledge bases, they do not describe this domain. This means that these dataset domain terms provide very little information about the domain itself. They even do not identify which domain they represent. This is sufficient for finding other datasets focused on the same domain. But it means that the dataset domain terms cannot be used for much else.

Limitation 4. *The dataset domain terms do not describe their domain at all, not even which domain they represent.*

5.2 Dataset discovery

In this section, I document the limitations of the dataset discovery technique that discovers datasets with a strong focus on a domain.

5.2.1 Text only. Same as Limitation 1, this technique only works with textual data in datasets. The reason for this limitation is the same since the terms are extracted from the datasets by D4.

Limitation 5. *The dataset discovery technique only matches textual terms from the dataset against the dataset domain terms.*

5.2.2 Limited by availability. Existing dataset discovery techniques that use information like knowledge bases are limited by the availability of those knowledge bases. This technique has the same drawback, as it is limited by the availability of the dataset domain terms. This research tries to mitigate this drawback by providing a type of information that is easier to create, i.e. dataset domain terms. But it can only mitigate this drawback, not remove it entirely. So it remains a limitation of this dataset discovery technique as well.

Limitation 6. *This technique is limited by the availability of dataset domain terms.*

5.2.3 Only strong focus. This technique looks for datasets with a strong focus on the domain. Having a strong focus on a domain means either the dataset covers a large portion of the domain, or the domain covers a large portion of the dataset. This is why datasets that have a small focus on the domain cannot be included in the result. For example, the Rotten Tomatoes dataset contains movie data but scores low because its focus is mainly on reviews and not movies.

Limitation 7. *This technique cannot detect datasets with a small focus on the domain.*

5.2.4 Small datasets. This technique depends on the relative number of terms matching between the dataset and dataset domain terms, compared to its total possible amount. But some terms match for many different domains which is why non-domain-focused datasets are expected to have a low number of matching terms but not zero. Unfortunately, this also means that if the dataset is very small, this low number of matching terms can already be a large percentage of that dataset. The consequence of this is that this technique has poor accuracy on small datasets.

This could have been mitigated by relating this amount to both their sizes, like with a Jaccard Index. As explained in Section 4.4.3, this was an explicit design choice. The choice of something like the Jaccard Index would reduce the score for all datasets, making it harder to detect low-scoring datasets.

Limitation 8. *This technique has poor accuracy on small datasets.*

6 EVALUATION

To ensure that this research provides suitable answers to the research questions, it must be evaluated. For my thesis, two techniques have been designed and implemented to answer the research questions. To ensure that they are suitable answers, I define evaluation criteria for them. These are the minimum requirements that these techniques must achieve in the evaluation. These criteria will also validate the assumptions that these techniques are based on.

In the next section, I first define the criteria for the technique to generate dataset domain terms. In the section after that, I define the criteria for the dataset discovery technique that uses these dataset domain terms. Then I explain the methodology used to evaluate these techniques according to these defined criteria.

6.1 Criteria for dataset domain terms

Research Question 1 asks what type of information can be used to understand data but is also easily available. In this thesis, I propose the use of *dataset domain terms*. The main requirement for these dataset domain terms is that they are representative of the domain of the datasets they are derived from. This also corresponds to Assumption 1 which states that deriving these dataset domain terms from column domains should result in terms that represent the domain. This means that, ideally, all dataset domain terms should represent the domain of the datasets they were derived from.

Due to Limitation 3, we know that the dataset domain terms can include terms that may not be representative of the domain. So instead of evaluating the overall representativeness of the dataset domain terms, it would be better to evaluate whether they are sufficiently representative to use in dataset discovery. Dataset domain terms that are most suitable for dataset discovery are ones that only represent their domain and no other domains, except for highly related ones. For example, a movie title like *Home Alone* would represent the 'movie and series information' domain but also other movie-related domains. It would not represent domains unrelated to movies, like 'healthcare' or 'population data'. This makes this term sufficiently representative of the 'movie and series information' domain.

To ensure a high similarity score, the majority of dataset domain terms should consist of such terms. This results in the following criterion for evaluating dataset domain terms.

Criterion 1. *At least 50% of the dataset domain terms are likely to only represent the dataset domain, or domains highly related to it.*

6.2 Criteria for dataset discovery

Research Question 2 asks how dataset discovery techniques can best use information to understand the data being analyzed when it may be missing or incomplete for some of this data. In this thesis, I address this with a dataset discovery technique that only focuses on a single domain. Its goal is to only discover datasets that focus on that domain, which is different from other dataset discovery techniques. But this difference ensures that any data that is not covered by the information about the domain can be considered to belong to a different domain and should be ignored by design.

So to evaluate this research question, we only need to ensure that this dataset discovery technique is sufficiently accurate in determining whether datasets focus on the domain or not. This accuracy should be sufficient if it is comparable to state-of-the-art dataset discovery techniques but does not need to exceed it.

My dataset discovery technique is based on Assumption 2. It states that domain-focused datasets should contain more terms that match the dataset domain terms than non-domain-focused datasets. Ensuring that the technique has sufficient accuracy also validates this assumption since such accuracy could not be achieved if the assumption were not valid.

However, since the goal of this dataset discovery technique is different from others, it cannot be compared directly to those other dataset discovery techniques. I would like to instead compare their evaluated level of accuracy to ensure that my technique performs comparably. But these dataset discovery techniques are typically only evaluated based on precision and recall and do not include their evaluated level of accuracy. These evaluate how accurately these techniques can discover the datasets they are intended to, and how completely those datasets can be discovered, respectively.

For my technique, it is equally important to discover datasets that have a strong focus on the domain as it is to exclude datasets that do not. This is why it should be evaluated based on accuracy, not precision or recall. Due to time constraints, reproducing the evaluation for other techniques to determine their accuracy is not feasible. So the best I can do is to ensure that the accuracy of my technique is comparable to the level of precision and recall of other dataset discovery techniques.

For this comparison, I look at the evaluation of dataset discovery techniques by Nargesian et al.[14] and Fernandez et al.[3] (called Aurum). The former uses linguistic information to understand its data whereas the latter derives domain information from the user query. The former published a precision and recall of 0.9095 and 0.8377. The latter published multiple measurements of precision and recall which averaged out to 0.8375 and 0.8475.

I define that accuracy is comparable to state-of-the-art techniques if it is no lower than the lowest accuracy (or precision or recall, if unavailable) of state-of-the-art techniques. This means that the accuracy of my dataset discovery technique must be at least 0.8375 to ensure that it is comparable to state-of-the-art techniques.

Criterion 2. *The accuracy of this technique is at least 0.8375.*

Dataset Name	Movie domain	FIFA players
IMDb	yes	no
MovieDB	yes	no
Indian Movies	yes	no
Movies Dataset	yes	no
Netflix	yes	no
TMDB 350K+ Movies	yes	no
Education	no	no
Finance	no	no
Services	no	no
Steam	no	no
Rotten Tomatoes	no	no
FIFA 22 players	no	yes
FIFA Players & Stats	no	yes
Fifa Players Ratings	no	yes
FIFA23 official dataset	no	yes
Football Events	no	yes
Energy consumption	no	no
Basket Analysis	no	no
NIPS Papers	no	no
Uber Pickups NYC	no	no
US Baby Names	no	no

Figure 6: Datasets used in the evaluation, along with whether they focus on the *movie* or *FIFA players* domain

6.3 Evaluation data

Before evaluating these techniques, we must define the datasets used in the evaluation. All datasets are listed in Figure 6. They are primarily taken from Kaggle[18]. The source for each dataset is documented on GitHub[19]. Due to Limitation 8, small datasets are likely to result in poor accuracy. So I made sure that each dataset contained at least 10,000 textual terms. This amount was sufficient to avoid being affected by this limitation in the evaluation.

6.3.1 Domain-representative datasets. I perform this experiment for two different domains, the *movie and series information* domain and the *FIFA players* domain. For the sake of convenience, I refer to this *movie and series information* domain as just the *movie* domain.

For the *movie* domain, I use the *IMDb* and *MovieDB* datasets. The explanation for why these datasets are domain-representative was provided in Section 3.2.1.

For the *FIFA players* domain, I use the *FIFA 22 players* and *FIFA Players & Stats*. Each of these datasets derives its data from a different source, which are *sofifa.com* and *fifaindex.com* respectively. This ensures that they do not contain duplicate data.

These datasets focus on describing these players as they appear in the *FIFA* video game series[20]. As such, the *FIFA players* domain represents FIFA players from the video game series. While highly related to the real-world FIFA players that the players in the video game are based on, as they have much information in common, they are not the same. For example, these datasets contain terms like 'CPU only' which could only relate to video games.

These datasets only cover the FIFA video games from the last decade, though the FIFA video games have been around since 1993. They should still be sufficiently complete to serve as domain-representative datasets since datasets for this domain seem to focus more on providing data from recent games.

6.3.2 Non-domain-focused datasets. For the non-domain-focused datasets, I include the datasets used in the evaluation of D4, *Education*, *Finance*, and *Services* datasets[12].

Representing datasets that might be automatically provided as input to the dataset discovery technique, I selected several CSV datasets between 100MB and 200MB from Kaggle at random, sorted by the highest vote count[21]. Datasets with fewer than 10,000 textual terms were manually excluded, as these would cause the evaluation to be affected by Limitations 5 and 8. This provided five datasets that do not focus on either of the domains for this evaluation.

I also included datasets that can be considered close to the domain but should not be considered to have a strong focus on it. For the *movie* domain, I included the *Rotten Tomatoes* and *Steam* datasets. The former does focus on movies but its main focus is on reviews. The latter describes video games, which are a different form of entertainment than movies. For the *FIFA players* domain, I included the *Football Events* dataset which describes real football events. This means it focuses on general football events and not on FIFA players specifically, neither in the real world nor in the video game series.

6.3.3 Domain-focused datasets. I include several datasets that focus on the domain to ensure that the dataset discovery technique has datasets that it can discover.

For the *movie* domain, I include *Indian Movies*, *The Movies Dataset*, *Netflix*, and *TMDB 350K+ Movies* which all describe movies.

For the *FIFA players* domain, I include *Fifa Players Ratings* and *FIFA23 official dataset*. The former focuses on the rating for FIFA players while the latter is a more up-to-date dataset focused on FIFA players. This latter dataset is very similar to the datasets used for the dataset domain terms, though it contains more recent data.

6.4 Evaluation methodology

The techniques are evaluated based on the two criteria defined earlier. This section describes the methodology used to take the measurements that need to be compared with these criteria.

The methodology is as follows, and should be performed for both the *movie* domain as well as the *FIFA players* domain.

(1) Generate the dataset domain terms.
(2) Categorize the dataset domain terms based on the column domains they were derived from.
(3) For each category, determine whether its terms are likely to represent only that domain (or ones highly related to it).
(4) Determine the percentage of each category relative to all dataset domain terms to evaluate Criterion 1.
(5) Use the dataset discovery technique with the generated dataset domain terms to discover domain-focused datasets.
(6) Calculate the accuracy of the technique based on the ground truth provided in Figure 6 to evaluate Criterion 2.

[18] https://kaggle.com/
[19] https://arucard21.github.io/domain-similarity
[20] https://en.wikipedia.org/wiki/FIFA_(video_game_series)
[21] The vote count can be considered a measure of quality so this provides the highest quality datasets first.

Figure 7: Categories of dataset domain terms for *movie* domain and their amount of terms. The domain relation describes whether it relates to the domain, only to the domain, or does not relate to any specific domain at all.

Category	Amount	Percentage	Domain relation
IMDb name ID	11617236	34.05048%	only movie
IMDB title ID	8908757	26.11184%	only movie
Title	4,469,372	13.09987%	only movie
Person name	9,041,625	26.50128%	only movie
Series name	80,496	0.23594%	only movie
Country code	113	0.00033%	movie
Language code	85	0.00025%	movie
Unknown	4	0.00001%	none

7 RESULTS

7.1 Dataset domain terms

7.1.1 Movie domain. Figure 7 shows the categories of terms for the *movie* domain, along with the number of terms in each category as well as whether they relate only to the *movie* domain (or domains highly related to it). There are five categories of terms that relate only to the *movie* domain.

The two biggest are identifiers used by the IMDb dataset, which are dataset-specific terms that are included in the dataset domain terms, as described by Limitation 3. While these identifiers are specific to IMDb, their presence in another dataset can only indicate that that dataset also focuses on the *movie* domain. This does show how this limitation can become problematic if the dataset-specific terms do not relate to its domain.

The *Title* category contains titles of movies and episodes and the *Series name* category contains the names of series. These relate only to the *movie* domain. While the *Person name* category might seem generic, the terms it contains are the names of people that work on movies. These are the names of the cast and crew of movies and series, which can be considered to relate only to the *movie* domain.

The remaining categories may also relate to other domains or not indicate a domain at all. As such, they are less useful when performing dataset discovery.

These results show that *99.99941%* of terms relate only to the *movie* domain which is well above the 50% required by Criterion 1.

7.1.2 FIFA players domain. Figure 8 shows the categories of terms for the *FIFA players* domain, similar to before. There are five categories of terms that relate only to the *FIFA players* domain.

The *Person name* category is again used, now consisting of the names of FIFA players. So this category relates only to the *FIFA players* domain. There are several categories of terms that describe different characteristics of FIFA players, i.e. the traits, tags, specialties, and positions. This means that these categories of terms only relate to the *FIFA players* domain.

As before, the remaining categories are less useful when performing dataset discovery. They may also relate to other domains or not indicate a domain at all. The dataset domain terms for this domain do not seem to contain many dataset-specific terms, showing that it is less affected by Limitation 3.

Figure 8: Categories of dataset domain terms for *FIFA players* domain and their amount of terms. The domain relation describes whether it relates to the domain, only to the domain, or does not relate to any specific domain at all.

Category	Amount	Percentage	Domain relation
Person name	9033	30.84304%	only FIFA players
Player traits	6653	22.71656%	only FIFA players
Player tags	165	0.56339%	only FIFA players
Player specialties	121	0.41315%	only FIFA players
Player positions	4287	14.63789%	only FIFA players
Club name	1433	4.89296%	FIFA players
League name	37	0.12634%	FIFA players
URL (flag/logo)	899	3.06962%	FIFA players
Country name	178	0.60778%	FIFA players
Body type	10	0.03414%	FIFA players
Date (birth)	6400	21.85270%	none
Unknown	58	0.19804%	none
Low/High	9	0.03073%	none
Left/right	4	0.01366%	none

Figure 9: Accuracy, precision, and recall for domain similarity technique.

Domain	Accuracy	Precision	Recall
Movie	0.85714	0.66666	1.0
FIFA players	1.0	1.0	1.0

These results show that *69.17404%* of the terms relate only to the *FIFA players* domain. While this is significantly less than with the *movie* domain, it is still well above the 50% required by Criterion 1.

7.2 Dataset discovery

The datasets detected as similar to the *movie* domain are:

- IMDb
- MovieDB
- Indian Movies
- Movies Dataset
- Netflix
- TMDB 350K+ Movies
- *FIFA Players & Stats*
- *Fifa Players Ratings*
- *US Baby Names*

Of these, the last three are incorrectly considered to be part of the *movie* domain. Though all other movie-related datasets were detected correctly. This results in an accuracy of *0.85714*, as specified in Figure 9 along with its precision and recall. This exceeds the minimum accuracy defined in Criterion 2 of 0.8375.

The datasets detected as similar to the *FIFA players* domain are:

- FIFA 22 players
- FIFA Players & Stats
- Fifa Players Ratings
- FIFA23 official dataset

All datasets related to the *FIFA players* were detected and no datasets were detected incorrectly. This means the accuracy was *1.0*, as

specified in Figure 9 along with its precision and recall. This exceeds the minimum accuracy defined in Criterion 2 of 0.8375.

8 DISCUSSION

The evaluation shows that both techniques satisfy the criteria defined for them. This means that the dataset domain terms are sufficiently representative for use in dataset discovery. And the accuracy of the dataset discovery technique that uses these dataset domain terms is comparable to state-of-the-art dataset discovery techniques.

The evaluation also shows that Limitation 3 can be a problem for these dataset domain terms. This limitation allows many dataset-specific terms to be included in the dataset domain terms. As with the IMDb identifiers in the evaluation, they are likely related only to the domain. But this is not guaranteed which means the dataset domain terms might include many dataset-specific terms that are not related to the domain. This would make those dataset domain terms unsuitable for dataset discovery. So this limitation needs to be overcome before these dataset domain terms can reliably be used for dataset discovery.

A less visible consequence of this limitation is that terms related to video games are included in the dataset domain terms for the *movie* domain. These terms are specific to the IMDb dataset as they are not contained in the MovieDB dataset. This contributed to the datasets for the FIFA video games being incorrectly considered to focus on the *movie* domain.

However, this limitation was not the only reason that these datasets for the FIFA video games were considered to focus on the *movie* domain. These datasets are indirectly related to the *movie* domain. The players in these FIFA video games represent football players in the real world that have the same names. These football players may appear in documentaries or short movies. So they would also be considered actors.

Similarly, the *US Baby Names* dataset is indirectly related to the *movie* domain as well. Many titles of movies, series, and episodes are just names, which would match the baby names in this dataset. For example, the name 'Bob' matches many titles[22], including some series and many short movies.

The discovery of these indirectly related datasets is the main reason that the precision of this dataset discovery technique is much lower than state-of-the-art dataset discovery techniques. This is a consequence of how domains can be related to each other in unexpected ways. As such, it would be difficult to improve the precision of this technique much further. It may be best to use another dataset discovery technique on these remaining datasets. Preferably one that analyzes the column names instead of the data. This should allow these indirectly related datasets to be excluded.

This shows that this dataset discovery technique would be most useful to filter datasets for other dataset discovery techniques. It can drastically reduce the number of datasets that subsequent dataset discovery techniques have to process. These subsequent dataset discovery techniques may still use additional information that is hard to create and has poor coverage. But these remaining datasets should only focus on a single domain or other domains that are indirectly related. This means that the range of domains for the data in these datasets is much more limited. As such, the dataset discovery techniques that are performed on them should be less affected by the poor coverage of the information they use to understand the data. In effect, this can mitigate the poor coverage of such information in existing dataset discovery techniques.

[22] https://www.imdb.com/find/?q=Bob&s=tt&exact=true

9 CONCLUSION

In this thesis, I identify a problem with dataset discovery, when performed on a large scale. The information that dataset discovery techniques use to better understand data has poor coverage. This means that this information may not be available for all data that these techniques need to understand. Part of the reason for this poor coverage is that current types of information, like knowledge bases and lexical databases, are difficult to create. As such, it is difficult to improve their coverage.

I propose a different type of information that is easier to create, i.e. *dataset domain terms*. I provide a technique to derive these dataset domain terms from existing datasets, as well as a technique to use them for dataset discovery. This latter technique further mitigates the problem of poor coverage by only focusing on a single domain. As such, its purpose is to discover datasets from the domain represented by the dataset domain terms.

The evaluation of these techniques shows that the dataset domain terms are suitable for dataset discovery. The accuracy of the dataset discovery technique is shown to be comparable to state-of-the-art dataset discovery techniques. While its recall was excellent, the precision of the dataset discovery technique was less impressive. This lower precision is due to the discovery of datasets that are indirectly related to the domain. These datasets are unlikely to be suitable for data integration.

This means that this dataset discovery technique may not be as useful as others on its own. But it would be useful to filter datasets before further dataset discovery is performed. Aside from drastically reducing the number of datasets, it would also limit the range of the domains for the data in these datasets. Subsequent dataset discovery techniques would need to understand data for a more limited range of domains, making them less affected by the poor coverage of the information they use to do so. This should allow dataset discovery to be performed on a larger scale than before.

REFERENCES

[1] Ali A., Azlin Nordin, Mogahed Alzeber, and Abedallah Zaid. 2017. A Survey of Schema Matching Research using Database Schemas and Instances. *International Journal of Advanced Computer Science and Applications* 8, 10 (2017). https://doi.org/10.14569/IJACSA.2017.081014

[2] Pier Luigi Buttigieg, Norman Morrison, Barry Smith, Chris J. Mungall, and Suzanna E. Lewis. 2013. The environment ontology: contextualising biological and biomedical entities. *Journal of Biomedical Semantics* 4 (2013), 43 – 43.

[3] Raul Castro Fernandez, Ziawasch Abedjan, Famien Koko, Gina Yuan, Samuel Madden, and Michael Stonebraker. 2018. Aurum: A Data Discovery System. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, Paris, 1001–1012. https://doi.org/10.1109/ICDE.2018.00094

[4] Anish Das Sarma, Lujun Fang, Nitin Gupta, Alon Halevy, Hongrae Lee, Fei Wu, Reynold Xin, and Cong Yu. [n. d.]. Finding related tables. In *Proceedings of the 2012 international conference on Management of Data - SIGMOD '12* (Scottsdale, Arizona, USA, 2012). ACM Press, 817. https://doi.org/10.1145/2213836.2213962

[5] Raul Castro Fernandez, Essam Mansour, Abdulhakim Ali Qahtan, Ahmed K. Elmagarmid, Ihab F. Ilyas, Samuel Madden, Mourad Ouzzani, Michael Stonebraker, and Nan Tang. 2018. Seeping Semantics: Linking Datasets Using Word Embeddings for Data Discovery. *2018 IEEE 34th International Conference on Data Engineering (ICDE)* (2018), 989–1000.

[6] Madelon Hulsebos, Kevin Hu, Michiel Bakker, Emanuel Zgraggen, Arvind Satyanarayan, Tim Kraska, Çagatay Demiralp, and César Hidalgo. [n. d.]. Sherlock: A Deep Learning Approach to Semantic Data Type Detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage AK USA, 2019-07-25). ACM, 1500–1508. https://doi.org/10.1145/3292500.3330993

[7] Christos Koutras, George Siachamis, Andra Ionescu, Kyriakos Psarakis, Jerry Brons, Marios Fragkoulis, Christoph Lofi, Angela Bonifati, and Asterios Katsifodimos. 2021. Valentine: Evaluating Matching Techniques for Dataset Discovery. *2021 IEEE 37th International Conference on Data Engineering (ICDE)* (2021), 468–479.

[8] Keqian Li, Yeye He, and Kris Ganjam. [n. d.]. Discovering Enterprise Concepts Using Spreadsheet Tables. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Halifax NS Canada, 2017-08-13). ACM, 1873–1882. https://doi.org/10.1145/3097983.3098102

[9] Jayant Madhavan, Shawn R. Jeffery, Shirley Cohen, Xin Dong, D. Ko, Cong Yu, and Alon Y. Halevy. 2007. Web-scale Data Integration: You can only afford to Pay As You Go.

[10] Hatem A. Mahmoud and Ashraf Aboulnaga. 2010. Schema clustering and retrieval for multi-domain pay-as-you-go data integration systems. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, Indianapolis Indiana USA, 411–422. https://doi.org/10.1145/1807167.1807213

[11] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *ICLR*.

[12] Heiko Mueller, Masayo Ota, Juliana Freire, and Divesh Srivastava. 2020. Data-Driven Domain Discovery (D4) - Evaluation Datasets.

[13] Fatemeh Nargesian, Ken Q. Pu, Erkang Zhu, Bahar Ghadiri Bashardoost, and Renée J. Miller. 2020. Organizing Data Lakes for Navigation. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. ACM, Portland OR USA, 1939–1950. https://doi.org/10.1145/3318464.3380605

[14] Fatemeh Nargesian, Erkang Zhu, Ken Q. Pu, and Renée J. Miller. 2018. Table Union Search on Open Data. *Proc. VLDB Endow.* 11 (2018), 813–825.

[15] Masayo Ota, Heiko Müller, Juliana Freire, and Divesh Srivastava. [n. d.]. Data-driven domain discovery for structured datasets. 13, 7 ([n. d.]), 953–967. https://doi.org/10.14778/3384345.3384346

[16] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *EMNLP*.

[17] Joel W. Reed, Yu Jiao, Thomas E. Potok, Brian A. Klump, M.T. Elmore, and Ali R. Hurson. 2006. TF-ICF: A New Term Weighting Scheme for Clustering Dynamic Data Streams. *2006 5th International Conference on Machine Learning and Applications (ICMLA'06)* (2006), 258–263.

[18] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. [n. d.]. YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia. ([n. d.]), 10.

[19] Dan Zhang, Madelon Hulsebos, Yoshihiko Suhara, Çağatay Demiralp, Jinfeng Li, and Wang-Chiew Tan. [n. d.]. Sato: contextual semantic type detection in tables. 13, 12 ([n. d.]), 1835–1848. https://doi.org/10.14778/3407790.3407793

[20] Meihui Zhang, Marios Hadjieleftheriou, Beng Chin Ooi, Cecilia M. Procopiuc, and Divesh Srivastava. 2011. Automatic discovery of attributes in relational databases. In *SIGMOD '11*.