

# Tabular open government data search for data spaces based on word embeddings

Alberto Berenguer  
University of Alicante  
Alicante, Spain  
aberenguer@dlsi.ua.es

David Tomás  
University of Alicante  
Alicante, Spain  
dtomas@dlsi.ua.es

Jose-Norberto Mazón  
University of Alicante  
Alicante, Spain  
jnmazon@dlsi.ua.es

## ABSTRACT

Nowadays, data spaces are envisioned as a prominent mechanism for data sharing, boosting growth and creating value. Open government data providers should be considered as important participants in data space reference infrastructures, since open data portal initiatives are adopted by most of the governments to supply their public sector information. However, open data is mostly published in the form of tabular data such as spreadsheets or CSV files. Therefore, reusing open data in data space is challenging due to the friction that may occur when combining the use of data shared in data spaces and the use of tabular data published in open government portals. To alleviate this situation, tabular open data search engines can be a promising solution. Actually, most open data portals allow reusers to retrieve and federate tabular open data by means of a keyword-based search engine over metadata. Unfortunately, these search engines rely on the (not so often good enough) metadata quality, which must be complete, descriptive, and representative of the content. Moreover, keyword-based search is not always an adequate solution for retrieving open data, since it does not consider their tabular nature and search results can be useless for reusers (e.g., when they attempt to find data useful for extending rows or columns of a given tabular dataset). To overcome these problems, this paper presents an approach that uses word embeddings for tabular open data search based on unionability and joinability. Our approach could be seamlessly integrated in a data space infrastructure. A prototype of our approach has been developed. Finally, both, an intrinsic and an extrinsic evaluation with end users, have been carried out.

## 1 INTRODUCTION

Trustworthy data sharing among several stakeholders is one of the main drivers of the data economy [29], for both (i) creating data-driven services and products, as well as (ii) supporting an informed decision making process. Within this scenario, data spaces are gaining momentum [21] and several initiatives have been launched, such as Gaia-X<sup>1</sup> or International Data Spaces Association.<sup>2</sup> A data space is a federated data infrastructure that supports trustworthy data sharing among data providers and data consumers, while ensuring data interoperability and data provider sovereignty.

Unfortunately, most data spaces initiatives rely on data from private partners and not enough attention is being paid to open government data coming from the public sector, which is considered as a relevant source in data spaces for boosting growth and creating value, as stated in [8]. Also, in a broader sense,

as highlighted by [9], open data is considered as a great way of foster innovation and enabling the creation of disruptive IT products and services. Indeed, a recent EU-funded innovation action called The Open Data Incubator for Europe<sup>3</sup> (ODINE) is aimed at incubating business ideas based on open data to create startup companies. ODINE reached an estimated €110M of cumulative revenues in the period 2016-2020, plus 784 jobs created. Interestingly, most funded companies within ODINE combined several open data sources to improve their products and services, thus suggesting that fully unleashing the potential of open data as an innovative business enabler requires to provide mechanisms to search and integrate open data coming from different sources [12].

Furthermore, the current amount of open government data available on the Web is increasing due to the strong interest of governments and institutions around the world in adopting open data initiatives [1]. Within these initiatives, open data portals are developed to publish public sector information under the appropriate formats and licences to encourage its re-use. Therefore, these portals publishing open government data must be considered by data spaces initiatives. Ideally for considering open government data within a data space, Linked Open Data (LOD) should be provided to enable stakeholders to use semantic web technologies to identify relationships among data [3], facilitating data searching and integration. Unfortunately, due to the complexity perceived by government adopters to publish LOD [26], linked data is not always available in open data portals. Furthermore, prevalence of tabular formats in open data portals has been highlighted in recent studies such as the Open Government Report from Organisation for Economic Co-operation and Development<sup>4</sup> (OECD). Also, the most used formats in open data portals are tabular (e.g. CSV), accounting for 46.5% [23].

This scenario may hamper the involvement of open government data portals in data space infrastructures. According to the review conducted by [8], most data spaces initiatives lack support from open government data providers. Interestingly, as an essential part of a data space is allowing users to search for required data [24], to fully consider open government data portals as a source in data spaces, additional efforts must be done to develop novel tabular open data search services.

However, tabular open data search has some problems that should be overcome. To illustrate these problems of searching tabular open data, imagine the following motivating example: a data journalist that wishes to expand an initial dataset containing the number of refugees arriving in France last year with information about refugees arriving in Spain. This open data can be found in the World Bank data portal,<sup>5</sup> which includes a keyword-based search engine. the data journalist poses the query “refugees in

<sup>1</sup><https://www.data-infrastructure.eu/>.

<sup>2</sup><https://internationaldataspaces.org/>.

<sup>3</sup><https://opendataincubator.eu/>.

<sup>4</sup><https://www.oecd.org/gov/open-government-data-report-9789264305847-en.htm>.

<sup>5</sup><https://databank.worldbank.org>

Spain” in the search engine, but there are no relevant results. Modifying the query to “refugees by country” also returns no results. Therefore, the data journalist cannot retrieve the required open data (although World Bank data portal indeed contains it in the “World Development Report” dataset), as metadata does not properly represent the content of the dataset and does not match the keywords that were considered as intuitive by the user.

This motivating example illustrates two main problems:

- Tabular open data search engines suffer from high dependence on metadata quality (e.g. title or description of datasets must be complete and accurate) in order to be able to provide reliable results. Keeping updated and accurate metadata is a major challenge for data publishers, as stated by the European Commission in its 2020 Open Data Maturity Report,<sup>6</sup> which can seriously affect the performance of search engines.
- It is not easy for open data reusers to express their intentions with a keyword-based search. If reusers want to expand an initial dataset with additional tabular data [33] by applying union and join operations, it seems more appropriate to represent the input query as a table rather than as a set of keywords.

To overcome these problems, this paper presents an approach based on word embeddings [19] to search tabular open data based on their unionability (related to the ability of a tabular dataset to be extended by adding rows from other dataset, i.e., applying a union operator) and joinability (related to the ability of a tabular dataset to be extended by adding columns from other dataset, i.e., applying a join operator). Word embeddings are a dense vector representation of texts, commonly used in Natural Language Processing (NLP) tasks, where words that have the same meaning have a similar vector representation. More specifically, our approach uses word embeddings to calculate the semantic similarity between tables, thus avoiding the need for exact matches between queries to retrieve relevant datasets. It also provides a table-based interface (beyond keyword-based search) in order to retrieve datasets considering the intentions of the users to extend rows (union operation) or columns (join operation). This approach could be seamlessly added into a data space infrastructure as a novel search service for tabular open government data that supports further integration of data from data spaces and open government data portals. A prototype has been developed<sup>7</sup> and evaluated carrying out an intrinsic and an extrinsic evaluation with end users.

The remainder of this paper is structured as follows: Section 2 presents related work to the field of data search systems; Section 3 describes the approach proposed for searching tabular open data based on word embeddings; the experimental evaluation of the system is described in Section 4; finally, Section 5 sketches out conclusions and future work.

## 2 RELATED WORK

Data search is a task studied for decades. However, there are still open problems like disconnected datasets, meeting user needs, and the availability and reliability of data [6].

In order to make open data more findable, organisations responsible for publishing datasets provide metadata like title, description, column names, author, creation date or language. To

carry out this task, publishers can use standards like DCAT,<sup>8</sup> a vocabulary that proposes metadata to describe datasets in an open data catalog to facilitate their consumption. Open data portals often use data publishing platforms aligned with DCAT, such as CKAN.<sup>9</sup> These platforms include search engines that make use of metadata from datasets. They are called centralised search engines [6] and present limited searching capabilities as they focus on a unique data catalogue. Also, they entirely depend on metadata availability and quality (i.e., the ability of metadata to correctly describe the data content).

Decentralised search engines, on the other hand, go beyond the boundaries of a single open data portal and provide ways to discover datasets across multiple catalogues [6], for instance, searching on LOD. There are also decentralized search engines on tabular open data, such as the general-purpose Google Dataset Search<sup>10</sup> that uses crawlers to search and index datasets that follow schema.org or DCAT standards. Decentralised search engines aim to adequately manage and enrich metadata for each dataset in order to improve retrieval results. However, they are keyword-based search engines, which is not always an adequate solution for retrieving open data, since it does not consider the tabular nature of (most) open data and search results can be useless for reusers. For instance, when they attempt to find open data to be integrated with a given tabular dataset by means of row or column extension. Unlike existing tabular open data search engines, the approach proposed in this paper makes use of word embeddings to provide semantic similarity capabilities that improve the recall of relevant datasets beyond metadata-based search.

There are several approaches for performing table-based search based on word embeddings. Zhang and Balog [32] combined two semantic vector spaces: one based on a knowledge base (DBpedia) and the other using pre-trained word embeddings (Word2vec). They used all the information available in the tables for the retrieval task (e.g. title, caption, headings, and entities).

The work in [28] also used Word2vec as the source for semantic vectors. The information of the table was separated in four semantic spaces: description (title and caption), schema (column headings), records (table rows), and facets (table columns). Then, different neural network architectures were applied to each semantic space, including recurrent convolutional neural network (description), multilayer perceptron (schema), and 3D convolutional neural network (records and facets).

To retrieve tables compatible with an input table, Nargesian et al. [22] tried to estimate if the table contents belonged to the same domain. They applied three statistical models: intersecting values between two columns, semantic similarity between values mapping the columns to classes in an ontology, and using word embeddings to measure similarity between textual values.

All the word embedding models mentioned above are non-contextual. The works presented in the following paragraphs use contextual word embeddings for table-based search. In [17] a survey on contextual embeddings is presented.

In [7] the authors used a pre-trained version of BERT [11], leveraging different information available in the table (both textual and numerical) to provide BERT with context: title, caption, column headings, and cell values. An important difference between the present work and that of Chen et al. [7] is the purpose of the table retrieval task.

<sup>6</sup><https://data.europa.eu/en/dashboard/2020>

<sup>7</sup><https://wake.dlsi.ua.es/datasearch/>

<sup>8</sup><https://www.w3.org/TR/vocab-dcat-2/>.

<sup>9</sup><https://ckan.org>.

<sup>10</sup><https://datasetsearch.research.google.com/>.

Our approach uses word embeddings, but two novel relevance measures are implemented to retrieve tabular datasets based on their unionability or joinability, considering intentions of users such as row or column extension.

There exist several works on determining whether it is possible to extend a query table with compatible rows (unionability) or new columns (joinability) [34]. *Row extension* task is similar to concept expansion, where an initial set of entities has to be completed with additional entities [34]. Previous approaches to row extension have used some sort of similarity between tables to find their compatibility. For example, Wang et al. [30] introduced concept names as input, together with seed entities to prevent lexical ambiguity. As far as the authors know, only the work by Deng et al. [10] previously used word embeddings (Word2vec) in this task. In the area of *column extension* (also known as *attribute discovery*), the approach presented in [5] was based on a database that included frequency statistics of attributes and co-occurring attribute pairs in a large table corpus (5.4 million unique attribute names). More recently, the authors of [31] took advantage of table captions and similarity between tables. The approach proposed in [2] was based on Wikipedia tables. The relatedness between tables was estimated based on the link intersections of Wikipedia pages. However, these studies were focused only on column headings and were not aimed to consider ability to apply join operations between tables.

### 3 TABULAR DATA SEARCH

Tabular data are usually searched for different purposes regarding data completion or data extension [33]. Traditional data completion refers to the use of a look-up table to complete missing values of an initial table. Interestingly, there are other intentions regarding how an initial table can be extended with additional data:

- Column extension (joinability): given an initial table, column extension means that new columns from other target table can be added to the initial one. It could correspond to a join operation in relational algebra (initial and target tables could be considered as joinable).
- Row extension (unionability): given an initial table, row extension means that new instances coming from other target table can be added to the initial table. It could correspond to a union operation in relational algebra (initial and target tables could be considered as unionable).

Therefore, when searching for tabular data, relevant data retrieved must fit the intentions of users for extending data. To do that, the approach proposed is based on using similarity measures for tabular data based on unionability and joinability. These measures are computed by using word embeddings and applied to search tabular open data.

Consequently, operators considered for handling input tabular data are union and join from relational algebra. For the sake of readability, the definition of these operators is borrowed from SQL (the well-known implementation of the relational algebra and a recognised standard for querying and handling tabular data):

- Union operator is denoted by  $\cup$  symbol in relational algebra. Given two tabular datasets A and B, union operator gets a unique dataset that contains rows that are in A or in B or both (denoted as  $A \cup B$ ). A and B must have the same columns (number, order, and datatype) to be computed.

Also, each column of each dataset must refer to the same concept to be meaningful.

- Join operator is denoted by  $\bowtie$  symbol in relational algebra. Given two tabular datasets A and B, join operator gets a unique dataset that includes every column from A and B (denoted as  $A \bowtie B$ ) and contains rows that fulfil a matching condition (applied to values of some columns).

#### 3.1 Similarity measure for tabular data

Word embeddings is a Natural Language Processing (NLP) technique in which words or phrases are mapped to vectors of real numbers, capturing the semantic regularities in this vector space. Similar semantic words which are more likely to share the same context have vectors that are closer in the embedding space [20].

Using word embeddings overcomes the problems of traditional search approaches based on string similarity, used frequently when searching on open data portals. For example, terms such as “city” and “location” could be considered as being very different in terms of string matching, but in a word embedding space these two terms may be closely related and considered as highly similar.

Examples of this type of word representations are Word2vec [19], fastText [4] and Glove [25]. These word embedding techniques build a global vocabulary using unique words in the documents, assigning a single representation for each word and ignoring that they can have different meanings or senses in different contexts. They are considered as static representations unable to capture the different senses of a word. On the other hand, recent contextual word embeddings [11] are able to capture the different meanings of polysemous words, since each vector represents not a word but a sense. In this way, each word is represented with different word embeddings, one for each context in which the word can occur. During the training process, contextual word embeddings are generated taking into consideration the surrounding words, that is, the sequence of words in the sentence or text span in which a word appears. Examples of these type of representation are ELMo [27], ULMFit [15] and BERT [11].

Next subsections describe the measure defined to compute column and table similarity by using word embedding models. These similarity measure will be used in the search approach proposed for retrieving relevant tables (as tabular open datasets) based on their ability to be integrated by means of join or union operators between tables (i.e., joinability and unionability).

**3.1.1 Column similarity.** The table similarity measure proposed is based on the similarity of individual columns. In order to calculate the similarity between columns of two different tables, two elements are taken into account: the name of the columns (headers) and the content (values) of the cells for each row. These values are normalized by splitting CamelCase and hyphenated words, removing punctuation, and converting text to lowercase.

A word embedding model is then used to extract two vectors for each column: one represents the name of the column and the other the content of the cells. In those situations where the name of the column includes more than one word, the vectors representing each word are averaged to get a single vector. Averaging word embeddings is one of the most popular methods of combining embedding vectors, outperforming more complex techniques especially in out-of-domain scenarios [14]. The same strategy is applied to the content of the cells, where the final vector is the result of calculating the mean between the vectors of each of the values contained.

The cosine similarity is used to compute the distance between vectors in the embedding space:

$$\text{sim}(v_1, v_2) = \frac{v_1 v_2}{\|v_1\| \|v_2\|} = \frac{\sum_{i=1}^n v_{1i} v_{2i}}{\sqrt{\sum_{i=1}^n (v_{1i})^2} \sqrt{\sum_{i=1}^n (v_{2i})^2}} \quad (1)$$

where  $v_1$  and  $v_2$  are the word embedding vectors of the name of the columns or the content of the cells, while  $\text{sim}(v_1, v_2)$  is a float value in the range  $[-1, 1]$  that represents the similarity between these two vectors, where  $-1$  means no similarity and  $1$  means maximum similarity between the vectors considered.

If the word embedding model does not provide coverage for the name of the column or its content (i.e. their tokens are not in the vocabulary of the model), the Levenshtein distance [16] is used as a backup strategy to ensure that the system always returns a similarity value between columns. This string metric is based on the number of single-character edits (insertions, deletions or substitutions) required to change one string into the other. We applied the normalised edit distance to obtain values in the range  $[0, 1]$ , computed as  $(\text{length} - \text{distance})/\text{length}$ , where distance is the Levenshtein distance and length is the sum of the lengths of the two strings compared.

For each two columns compared, we obtain a similarity value of the name of the column and a similarity value of its content. To obtain a single final similarity score of two columns  $C_1$  and  $C_2$ , the linear combination is computed.

$$\text{sim}(C_1, C_2) = \alpha \cdot \text{sim}(C_{n1}, C_{n2}) + (1 - \alpha) \cdot \text{sim}(C_{c1}, C_{c2}), \quad (2)$$

where  $\text{sim}(C_{n1}, C_{n2})$  is the similarity of the column names,  $\text{sim}(C_{c1}, C_{c2})$  is the similarity of their content calculated using Equation 1, and  $\alpha$  is a parameter in the range  $[0, 1]$  that weights the relevance of the two similarity scores in the final result.

**3.1.2 Table similarity.** The column similarity previously defined is used to compute the similarity between two tables. Two different ways of computing table similarity are proposed. If table similarity is computed for row extension (related to unionability), the following formula is used:

$$\text{sim}(t_1, t_2) = \frac{\sum_{i=1, j=i}^{i \leq n, j \leq m} \text{sim}(c_{1i}, c_{2j})}{|C_1||C_2|}, \quad (3)$$

where  $C_1 = \{c_{11}, c_{12} \dots c_{1n}\}$  and  $C_2 = \{c_{21}, c_{22} \dots c_{2m}\}$  are the set of columns from table  $t_1$  and table  $t_2$ , respectively. That is, the similarity between two tables is computed as the average similarity of their columns, since row extension requires a great number of similar columns. Therefore, the proposed measure will be useful to find potentially unionable tables for a given query table.

Otherwise, if the table similarity is computed to extend columns (related to joinability), then the following formula is used:

$$\text{sim}(t_1, t_2) = \max_{i \leq n, j \leq m} \{\text{sim}(c_{1i}, c_{2j})\}. \quad (4)$$

That is, the similarity between two tables is computed as the maximum similarity of a pair of columns, since column extension implies a minimum number of similar columns to join tables and then adding new columns. Thus, Equation 4 will be useful to find potentially joinable tables for a given a query table.

## 3.2 Searching unionable and joinable tabular data

Our searching approach aims to detect the tabular datasets that are more likely to be integrated by means of join and union operations. Unionability and joinability of tabular open datasets are computed according to the previously-described similarity measures between columns and tables, based on word embeddings.

Specifically, given a set of tables  $\tau = \{t_1, \dots, t_n\}$  and a query table  $Q$ , the top-k tables in  $\tau$ , whose unionability or joinability with  $Q$  is the highest, must be found. Therefore, the search process requires a previous step of indexing available tabular datasets (tables) in  $\tau$ .

**3.2.1 Indexing tabular data.** In our approach, tabular open data are retrieved from open data portals that comply with the DCAT vocabulary and is available in CSV format. The following information is retrieved and stored for each table: (i) metadata extracted from the portal,<sup>11</sup> such as title, description, publication year, etc.; (ii) tabular open data content (both column names and row instances); and (iii) word embedding vectors from each column.

**3.2.2 Searching tabular data.** Our searching approach is divided into seven steps, as shown in Figure 1. The following list describes each of these steps:

- (1) Considering table  $Q$  as the input query. This table will be extended according to the user's search intention (i.e., column or row extension).
- (2) Sampling  $Q$ : it takes the query table  $Q$  and checks its size. If the  $Q$  size is difficult to handle, a random sample of the rows is obtained (avoiding duplicated values). Therefore, the time required during the search process is decreased without affecting the word embedding representation.
- (3) Normalisation of  $Q$ : it takes the query table  $Q$  and normalises each column  $C$ .
- (4) Obtain embeddings from  $Q$ : word embeddings  $v$  for each term in  $C_n$  (column header) and  $C_c$  (cell content) are computed by using an embedding model. In order to obtain a single embedding for  $C_n$  and  $C_c$ , the average embedding  $V$  of column header and cell content is obtained as follows:
$$V = \frac{\sum_{v \in \vec{c}} \vec{v}}{n_c}$$
- After this step, all the columns in  $Q$  with their two average embeddings corresponding to  $C_n$  and  $C_c$  are obtained.
- (5) Search: for each column  $C$ , the identifiers of the previously stored top-k most suitable  $C_n$  and  $C_c$  (indexed from  $\tau$  tables as explained in at the beginning of this subsection) are obtained together with their corresponding similarity measures.
- (6) Search a table for completion: until now, a set of top-k candidates for each column  $C$  is determined. Now, table  $T$  from  $\tau$ , which belongs to each  $C$  candidate, must be retrieved together with metadata about column identifiers, table identifiers, and scores.
- (7) Final scoring: the final similarity measure is computed differently if the intention of the user is row extension (the user is searching for unionable tables) or column extension (the user is searching for joinable tables). Row extension requires applying Equation 2 and Equation 3,

<sup>11</sup>It is not used in the search approach, but in the Web interface of the prototype to allow users filtering results.

while column extension requires applying Equation 2 and Equation 4. Finally, these scores are sorted and returned to the user, retrieving the top-k most suitable tables that could be integrated with the query table  $Q$ .

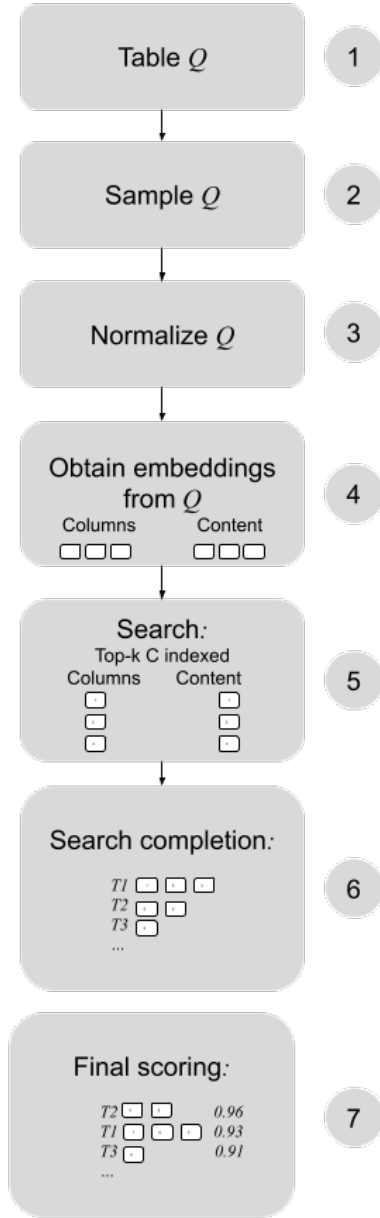


Figure 1: Search process workflow.

### 3.3 Search! prototype

Our approach was implemented in a prototype named *Search!* that can be accessed at <https://wake.dlsi.ua.es/datasearch/>. Our *Search!* prototype indexes tabular open data from DCAT-based open data portals as the Chicago open data portal or the World Bank data portal. To store both tabular data and their metadata, the prototype proposed uses Solr.<sup>12</sup> To store the word embedding

<sup>12</sup><https://solr.apache.org/>.

vectors, instead of using Solr (which was proven highly inefficient in the experimental setup, taking as long as 20 minutes to manage 200 tables), Faiss<sup>13</sup> is used. Faiss is a library for efficient similarity search and clustering of dense vectors by using binary representations. Each instance indexed by Faiss generates a numeric identifier stored in Solr with the corresponding metadata. By using Faiss, *Search!* prototype takes only 2.3 seconds to manage around 200 tables. The search interface for tabular data is shown in Figure 2, while the interface of relevant retrieved datasets is shown in Figure 3.

## 4 EVALUATION

In order to show the feasibility of our approach, two experiments were carried out. The first one provides an intrinsic evaluation of the search algorithm, testing different word embedding models to identify the best performing solution for the table retrieval task. The second experiment consists of an extrinsic evaluation involving a user study in which we compared two approaches for searching open data: (i) using search engines of open data portals, and (ii) using the *Search!* prototype that implements our word embedding approach for tabular open data search.

### 4.1 Intrinsic evaluation

This section describes the evaluation of different word embedding models in retrieving the most relevant tables for a given one. As described in Equation 3 and Equation 4, two different objectives have been defined depending on whether the goal is row or column extension. The performance of the models was measured using precision, that is, the fraction of relevant instances among retrieved instances.

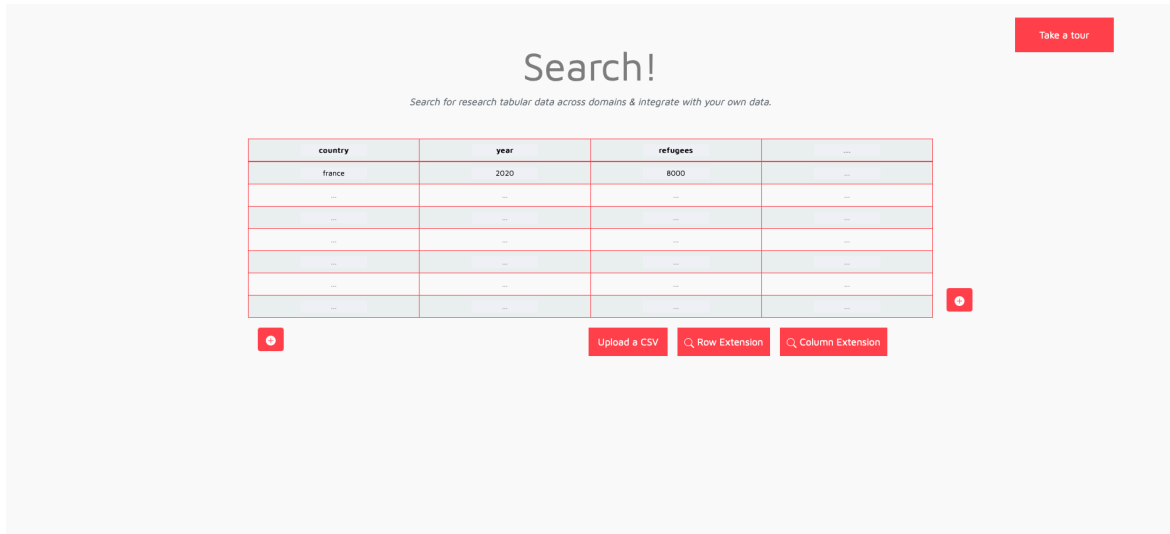
The dataset used in these experiments was developed by Nargesian et al. [22]. It was originally intended for table union search, but in the following experiments it has been adapted to also evaluate join operations, thus considering unionability and joinability criteria. This dataset consists of more than 5,000 tables in CSV format extracted from USA, Canada, and UK open data portals, providing a ground truth that identifies which columns of a table match the columns of another table. The dataset was built starting with 32 base tables manually aligned to identify matching columns. The final set was created by first issuing a projection on a random subset of columns of a base table, and then a selection with some limit and offset on the projected table.

To perform the experiments, a subset of 1,000 tables was randomly selected. Every table in this subset was used as a query to the system and compared with all the other tables in the subset.

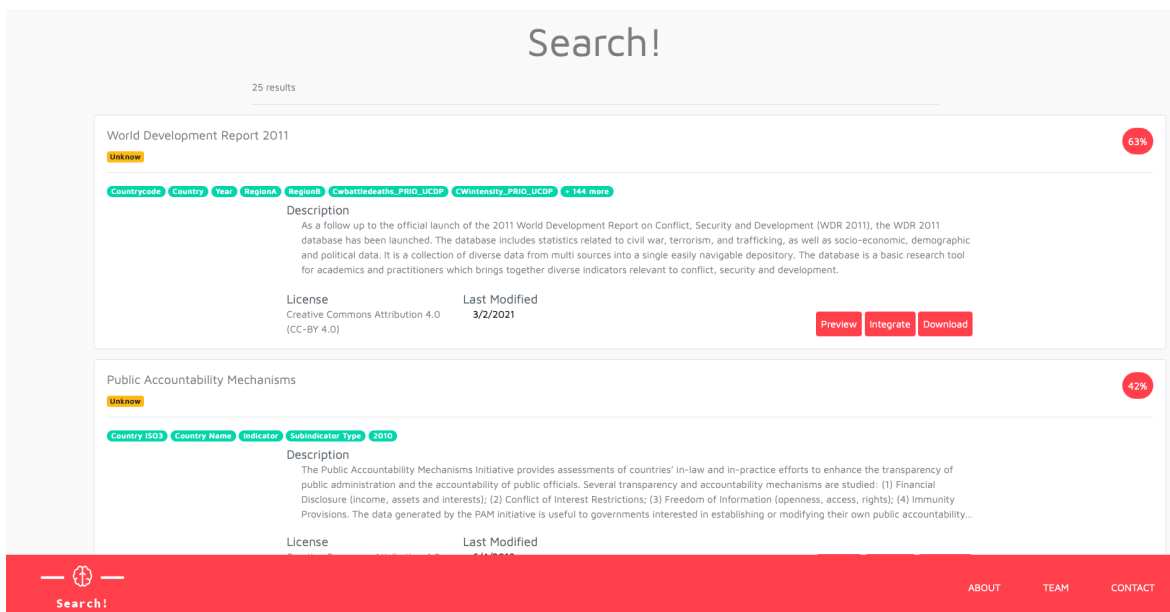
Four different word embedding models were evaluated, including two non-contextual (Word2vec and fastText) and two contextual (BERT and RoBERTa) models:

- Word2vec: embedding vectors pre-trained on part of Google News dataset, comprising about 100 billion words [19]. The model contains vectors for 3 million words and phrases.
- fastText: embedding vectors pre-trained on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset, comprising about 16 billion words [4].
- BERT: the version of the model evaluated is BERT-base, containing 12 layers (transformer blocks), 12 attention heads, and 110 million parameters [11].
- RoBERTa: the version evaluated is RoBERTa-base, containing 12 layers, 12 attention heads, and 125 million parameters [18].

<sup>13</sup><https://github.com/facebookresearch/faiss>.



**Figure 2: Tabular search interface.**



**Figure 3: Interface to provide ranking of most suitable retrieved tabular datasets.**

The implementation of Word2vec and fastText was carried out with the Gensim library.<sup>14</sup> The contextual models were implemented using the Transformers library developed by Huggingface.<sup>15</sup>

**4.1.1 Row extension.** The goal of row extension is to retrieve the most appropriate tables to perform union operations for a given table. The ranking criterion was described in Equation 2. In the experiments presented here, a ranked table is considered to be relevant if it contains at least one column that could be aligned with another column of the query table in a union operation, as specified in the ground truth provided in [22].

Each embedding model was evaluated using different values of  $\alpha$  (from 0 to 1 inclusive, in 0.1 increments), as described in

Equation 2, to analyse the influence of the column headings and the cell values in the performance of the system.

Table 1 shows precision for the four models mentioned above. It can be observed that BERT obtained the best results in these experiments for every  $\alpha$  value considered, followed by RoBERTa. The non-contextual models (Word2vec and fastText) performed significantly worse than the contextual models.

**4.1.2 Column extension.** The objective of column extension is to retrieve the most relevant tables to perform join operations. The ranking function for this task is defined in Equation 3.

The ground truth used in the row extension experiment only identifies on which columns the union operation can be applied. The way in which the previous dataset was obtained has been leveraged here to also evaluate the models in column extension. As mentioned above, tables were obtained by projection and

<sup>14</sup><https://radimrehurek.com/gensim/>.

<sup>15</sup><https://github.com/huggingface/transformers/>.

**Table 1: Precision of word embedding models in row extension.**

$\alpha$	Word2vec	fastText	BERT	RoBERTa
0.0	0.7380	0.8363	0.9088	0.8971
0.1	0.8094	0.8842	0.9871	0.9836
0.2	0.8129	0.8854	0.9930	0.9825
0.3	0.8117	0.8889	0.9930	0.9813
0.4	0.8129	0.8889	0.9918	0.9298
0.5	0.8129	0.8877	0.9906	0.9255
0.6	0.8129	0.8877	0.9895	0.9275
0.7	0.8129	0.8842	0.9848	0.9333
0.8	0.8129	0.8807	0.9836	0.9263
0.9	0.8117	0.8784	0.9825	0.9275
1.0	0.7579	0.7228	0.9789	0.8386

selection of 32 original tables manually aligned. On this basis, the criterion to identify whether two tables from the dataset can be joined is to verify if they were both obtained from the same original table (one of the 32 mentioned before), and if they have at least one column in common with the same name. Meeting these conditions ensures that the tables can be joined by that column.

On the other hand, the original ground truth identifies what columns can be matched between tables. Thus, two tables cannot be joined if they do not have any columns in common based on this ground truth. For the pairs of tables that do not fulfil any of these conditions, it cannot be guaranteed whether they can be joined or not, so they were discarded in the evaluation. Since column headings are chosen as a basis to determine whether two columns can be joined, to conduct an unbiased assessment models were evaluated using  $\alpha = 0.0$ , avoiding the use of headings as an evidence to perform the join operation.

Table 2 shows the precision of the four word embedding models.

As in the previous experiment, BERT obtained the best results for the column expansion. Thus, this model was used (setting  $\alpha = 0.3$ , the best value for row extension) in the extrinsic evaluation carried out with the final users.

## 4.2 Extrinsic evaluation

In this experiment, the search algorithm defined is integrated in a search portal and evaluated with end users according to their intentions for retrieving datasets (row extension or union operation, as well as column extension or join operation). This experiment tries to answer the following questions:

- How accurate is this approach proposed with regards to search engines from open data portals?
- Could this approach help to save time to data consumers when searching for tabular open data?

To answer these questions, a set of surveys were conducted. Each survey consists of four data request to be resolved. A data request states an initial table and asks for searching the right tabular open data to expand the initial table (either row or column extension) together with the time taken to fulfill the task.

Data requests are related to specific scenarios that needs tabular open data from the City of Chicago open data portal<sup>16</sup> and

from the World Bank Open Data portal,<sup>17</sup> as well as different intentions (unionable or joinable tabular data to retrieve).

Specifically, the scenarios for data requests of each survey are:

- *Scenario 1.* Given cultural events that have happened in Chicago in a given year, add more events from a different year (row extension / unionability).
- *Scenario 2.* Given terrorist incidents by country and year, add information from additional countries (row extension / unionability).
- *Scenario 3.* Given Gross Domestic Product (GDP) per country, add the population of each country (column extension / joinability).
- *Scenario 4.* Given city and street names, add daily average cars in each street (column extension / joinability).

The subjects of the experiment were 44 students of the a summer course on big data technologies held in University of Alicante (Spain) during July 2021 with a duration of 20 hours. Attendees were mainly practitioners and final year students from IT-related degrees. Subjects of the experiment were divided into two groups. One group of 22 participants tried to solve the questions using directly search engines from open data portals. The other group of 22 used the system proposed by using the *Search!* prototype. It is worth noting that all the participants had similar previous knowledge of technologies used in the experiments, and they were also simultaneously instructed in the use of concrete open data portals and *Search!*. When participants responded to the survey, they had to provide the data sets that satisfy each data request, as well as the required invested time. Results of both groups are analysed to answer the aforementioned questions.

The results obtained from the analysis of the time that participants employed to give a response for each data request (or query) have a different pattern depending on whether they were using open data portals or the *Search!* prototype, as shown in Figure 4. Also, Table 5 shows the average time employed by participants to solve each data request scenario. Results of solving each scenario are shown in both Table 6) and Figure 5. All these results show that users spend less time using the *Search!* prototype and also achieved better results. Also, most of the participants using the *Search!* prototype correctly answered all the questions.

To achieve a more solid comparison of the results, the *t-student* statistical test was applied to check if the difference between both groups was statistically significant. To do so, the standard error and deviation of the results were computed. If both group

<sup>16</sup><https://data.cityofchicago.org/>.

<sup>17</sup><https://datacatalog.worldbank.org/>.

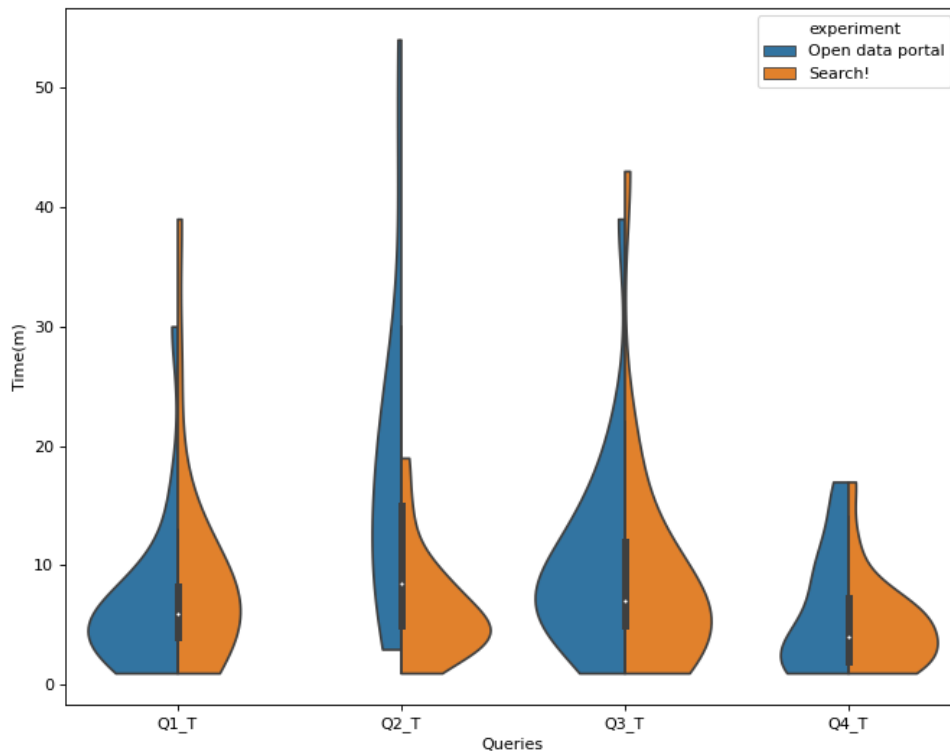


Figure 4: Time by query and type.

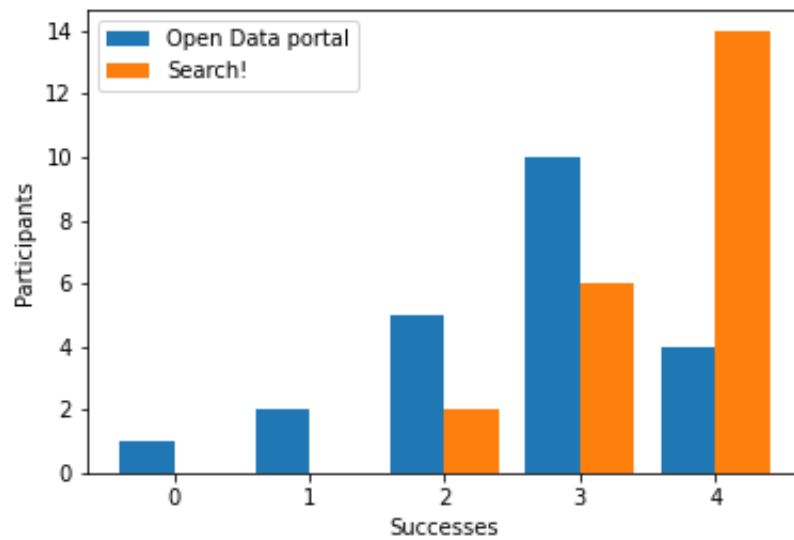


Figure 5: Participants by number of successful searches and type.



**Table 2: Precision of word embedding models ( $\alpha = 0.0$ ) in column extension.**

Model	Precision
Word2vec	0.6141
fastText	0.7656
BERT	0.8258
RoBERTa	0.8144

averages are equal, it means that the null hypothesis is found. This is evaluated by comparing the critical values from the t-distribution. The critical value can be calculated using the degrees of freedom and significance level with the percentage point function. If  $abs(t - statistic) \leq criticValue$ , the null hypothesis is accepted and the average of both groups is equal. If  $abs(t - statistic) > criticValue$ , the null hypothesis is rejected (i.e., the average is not equal). The  $\rho$ -value can be compared by choosing a significance level  $\alpha = 0.05$  to check if the null hypothesis is rejected. Therefore, if  $\rho > \alpha$ , the null hypothesis is accepted, while if  $\rho \leq \alpha$ , it is rejected.

First, it is assumed that the average search time from both groups is equal and the amount of correct answers is the same, i.e., as null hypothesis ( $h_0 : \mu_0 = \mu_1$ ) and as an alternative hypothesis ( $h_1 : \mu_0 \neq \mu_1$ ).

With a significance level  $\alpha = 0.05$ , results in Tables 3, and 4 are obtained. Both Table 3 and Table 4 state that  $abs(t - statistic) \geq criticValue$  and ( $\rho - value < 0.05$ ). Therefore, null hypothesis is rejected in both cases, and there exists a significant difference between the two groups of participants. That is, the results given by our approach for searching tabular data are significantly better than using search engines of open data portals.

On one hand, the group of subjects that used the *Search!* prototype had an average of 88.5% of accuracy resolving the search scenarios as well as an average of time spent of 7.36min by scenario. On the other hand, the group of subjects that used open data portal search engines to resolve the scenarios had an accuracy of 65% as well as an average time of 10min spent by scenario.

Therefore, the use of the *Search!* prototype performed better than the use of search engines for open data portals, both in terms of search time and search success.

## 5 CONCLUSIONS AND FUTURE WORK

An essential part of a data space is allowing users to search for required data [24]. Within a data space infrastructure, the development of mechanisms that support retrieval of tabular datasets beyond keyword-based search on metadata is highly required for considering government open data as a relevant source in data spaces.

In this paper, a novel approach for tabular open data search is proposed. This approach (i) makes use of word embeddings to provide semantic similarity capabilities that improve the recall of relevant datasets beyond metadata-based search, and (ii) states an input query in tabular form (beyond keyword-based search) in order to retrieve datasets based on their unionability or joinability (i.e., considering intentions such as row or column extensions). Moreover, a prototype of the approach named *Search!* was implemented.

A controlled experiment was conducted to evaluate the approach. The intrinsic evaluation carried out compared four different language models in order to identify the best word embedding

representation for the tables in the search task. The results allowed to identify that contextual word embeddings performed better than non-contextual models. BERT was selected as the best model to be included in the system further used in the extrinsic evaluation.

The extrinsic evaluation with end-users shows that they can use *Search!* more successfully than existing search engines in open data portals.

As the word embedding models used in the experiments were language-dependent, they only worked for English. As a future work, models in other languages can be used to make the system suitable for a wider audience. Another possibility is to use multilingual word embeddings that can be applied seamlessly to different languages [13].

Also, a new evaluation should be performed beyond search engines in open data portals. For example, by comparing *Search!* with decentralised search engines like Google Dataset Search. Finally, fully integration with data space infrastructures is also planned to be explored as a future work.

*Acknowledgements.* This work is part of the project TED2021-130890B-C21, funded by MCIN/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR. Also, this work is partially funded by GVA-COVID19/2021/103 project from “Conselleria de Innovación, Universidades, Ciencia y Sociedad Digital de la Generalitat Valenciana”. Alberto Berenguer has a contract for predoctoral training with the “Generalitat Valenciana” and the European Social Fund, funded by the grant ACIF/2021/507.

## REFERENCES

- [1] Mohammed Saleh Altayar. 2018. Motivations for open data adoption: An institutional theory perspective. *Government Information Quarterly* 35, 4 (2018), 633–643. <https://doi.org/10.1016/j.giq.2018.09.006>
- [2] Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. 2013. Methods for Exploring and Mining Tables on Wikipedia. In *Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics (IDEA '13)*. Association for Computing Machinery, Chicago, Illinois, 18–26. <https://doi.org/10.1145/2501511.2501516>
- [3] Christian Bizer, Tom Heath, and Tim Berners-Lee. 2011. Linked data: The story so far. In *Semantic services, interoperability and web applications: emerging concepts*. IGI global, 205–227.
- [4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics* 5 (2017), 135–146. [https://doi.org/10.1162/tacl\\_a\\_00051](https://doi.org/10.1162/tacl_a_00051)
- [5] Michael J. Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. 2008. WebTables: Exploring the power of tables on the web. In *Proceedings of the Very Large Data Base Endowment*. 538–549. <https://doi.org/10.1145/14778/1453856.1453916>
- [6] Adriane Chapman, Elena Simperl, Laura Koesten, George Konstantinidis, Luis-Daniel Ibáñez, Emilia Kacprzak, and Paul Groth. 2020. Dataset search: a survey. *VLDB J.* 29, 1 (2020), 251–272. <https://doi.org/10.1007/s00778-019-00564-x>
- [7] Zhiyu Chen, Mohamed Trabelsi, Jeff Hefflin, Yinan Xu, and Brian D. Davison. 2020. Table Search Using a Deep Contextualized Language Model. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, Virtual, 589–598. <https://doi.org/10.1145/3397271.3401044>
- [8] Oscar Corcho and Elena Simperl. 2022. data.europa.eu and the European Common Data Spaces. *Publications Office of the European Union* (2022). [https://doi.org/sites/default/files/report/EN\\_data\\_europa\\_eu\\_and\\_](https://doi.org/sites/default/files/report/EN_data_europa_eu_and_)

**Table 3: t-student results: time spent when searching datasets.**

t-statistic	degrees of freedom	critic value	$\rho$ -value
2.143	164	1.653	0.033

**Table 4: t-student results: correct answers when searching datasets.**

t-statistic	degrees of freedom	critic value	$\rho$ -value
-3.716	164	1.653	0.0002

**Table 5: Average elapsed time by scenario (in minutes).**

Approach	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Open data portal	6.45	17.36	10.40	6.00
Search!	9.13	6.59	8.72	5.00

**Table 6: Average of correct solutions by scenario.**

Approach	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Open data portal	82%	50%	45%	86%
Search!	86%	95%	73%	100%

- the\_European\_common\_data\_spaces\_0.pdf
- [9] Diego Corrales-Garay, Eva-Maria Mora-Valentin, and Marta Ortiz-de Urbina-Criado. 2019. Open data for open innovation: An analysis of literature characteristics. *Future Internet* 11, 3 (2019), 77.
- [10] Li Deng, Shuo Zhang, and Krisztian Balog. 2019. Table2Vec: Neural Word and Entity Embeddings for Table Population and Retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, Paris, France, 1029–1032. <https://doi.org/10.1145/3331184.3331333>
- [11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Minneapolis, MN, USA, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- [12] Robert Enriquez-Reyes, Susana Cadena-Vela, Andrés Fuster Guilló, Jose-Norberto Mazón, Luis-Daniel Ibáñez, and Elena Simperl. 2021. Systematic Mapping of Open Data Studies: Classification and Trends From a Technological Perspective. *IEEE Access* 9 (2021), 12968–12988.
- [13] Félix Gaschi, François Plesse, Parisa Rastin, and Yannick Toussaint. 2022. Multilingual Transformer Encoders: a Word-Level Task-Agnostic Evaluation. In *International Joint Conference on Neural Networks, IJCNN*. IEEE, Padua, Italy, 1–8. <https://doi.org/10.1109/IJCNN55064.2022.9892197>
- [14] Sukrat Gupta, Teja Kanchinadam, Devin Conathan, and Glenn Fung. 2020. Task-optimized word embeddings for text classification representations. *Frontiers in Applied Mathematics and Statistics* (2020), 67.
- [15] Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Melbourne, Australia, 328–339. <https://doi.org/10.18653/v1/P18-1031>
- [16] Vladimir Iosifovich Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady* 10, 8 (feb 1966), 707–710. *Doklady Akademii Nauk SSSR*, V163 No4 845–848 1965.
- [17] Qi Liu, Matt J. Kusner, and Phil Blunsom. 2020. A Survey on Contextual Embeddings. *CoRR abs/2003.07278* (2020). <https://arxiv.org/abs/2003.07278>
- [18] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR abs/1907.11692* (2019). <https://arxiv.org/abs/1907.11692v1>
- [19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'13)*. Curran Associates Inc., Lake Tahoe, Nevada, 3111–3119.
- [20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* 26 (2013).
- [21] Marco Minghini, Alexander Kotsev, and Carlos Granell. 2022. A European Approach to the Establishment of Data Spaces. *Data* 7, 8 (2022). <https://doi.org/10.3390/data7080118>
- [22] Fatemeh Nargesian, Erkang Zhu, Ken Q. Pu, and Renée J. Miller. 2018. Table Union Search on Open Data. *Proceedings of the Very Large Data Base Endowment* 11, 7 (2018), 813–825. <https://doi.org/10.14778/3192965.3192973>
- [23] Sebastian Neumaier, Jürgen Umbrich, and Axel Polleres. 2016. Automated quality assessment of metadata across open data portals. *Journal of Data and Information Quality (JDIQ)* 8, 1 (2016), 1–29.
- [24] Boris Otto. 2022. A federated infrastructure for European data spaces. *Commun. ACM* 65, 4 (2022), 44–45. <https://doi.org/10.1145/3512341>
- [25] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar, 1532–1543. <https://doi.org/10.3115/v1/D14-1162>
- [26] Bruno Elias Penteado, José Carlos Maldonado, and Seiji Isotani. 2022. Methodologies for publishing linked open government data on the web: a systematic mapping and a unified process model. *Semantic Web Preprint* (2022), 1–26.
- [27] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 2227–2237. <https://doi.org/10.18653/v1/N18-1202>
- [28] Roei Shraga, Haggai Roitman, Guy Feigenblat, and Mustafa Cannim. 2020. Web Table Retrieval using Multimodal Deep Learning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, Virtual, 1399–1408. <https://doi.org/10.1145/3397271.3401120>
- [29] Gürkan Solmaz, Flavio Cirillo, Jonathan Fürst, Tobias Jacobs, Martin Bauer, Ernő Kovacs, Juan Ramón Santana, and Luis Sánchez. 2022. Enabling data spaces: existing developments and challenges. In *Proceedings of the 1st International Workshop on Data Economy*. 42–48.
- [30] Chi Wang, Kaushik Chakrabarti, Yeye He, Kris Ganjam, Zhimin Chen, and Philip A. Bernstein. 2015. Concept expansion using web tables. In *Proceedings of the 24th International Conference on World Wide Web*. Florence, Italy, 1198–1208. <https://doi.org/10.1145/2736277.2741644>
- [31] Shuo Zhang and Krisztian Balog. 2017. EntiTables: Smart assistance for entity-focused tables. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, Shinjuku, Tokyo, Japan, 255–264. <https://doi.org/10.1145/3077136.3080796>
- [32] Shuo Zhang and Krisztian Balog. 2018. Ad Hoc Table Retrieval using Semantic Similarity. In *Proceedings of the 2018 World Wide Web Conference*. Lyon, France, 1553–1562. <https://doi.org/10.1145/3178876.3186067>
- [33] Shuo Zhang and Krisztian Balog. 2020. Web Table Extraction, Retrieval, and Augmentation: A Survey. *ACM Trans. Intell. Syst. Technol.* 11, 2, Article 13 (Jan. 2020), 35 pages. <https://doi.org/10.1145/3372117>
- [34] Shuo Zhang and Krisztian Balog. 2020. Web Table Extraction, Retrieval, and Augmentation: A Survey. *ACM Transactions on Intelligent Systems and Technology* 11, 2 (2020), 1–35. <https://doi.org/10.1145/3372117>