ChemTables: A Dataset for Semantic Classification of Tables in Chemical Patents

Zenan Zhai
The University of Melbourne https://orcid.org/0000-0003-1391-6950
Christian Druckenbrodt
Elsevier Data Science - Life Science https://orcid.org/0000-0002-3819-6067
Camilo Thorne
Elsevier - Data Science, Life Science https://orcid.org/0000-0002-6960-772X
Saber A Akhondi
Elsevier - Data Science, Life Science https://orcid.org/0000-0003-2855-5633
Dat Quoc Nguyen
VinAI https://orcid.org/0000-0001-8214-2878
Trevor Cohn
The University of Melbourne https://orcid.org/0000-0003-4363-1673
Karin Verspoor (karin.verspoor@unimelb.edu.au)
The University of Melbourne https://orcid.org/0000-0002-8661-1544

Research article

Keywords: Neural networks, Table classication, Chemical patents, Chemical patents

Posted Date: December 16th, 2020

DOI: https://doi.org/10.21203/rs.3.rs-127219/v1

RESEARCH

ChemTables: A dataset for semantic classification on tables in chemical patents

Zenan Zhai[1], Christian Druckenbrodt[2], Camilo Thorne[2], Saber A. Akhondi[2], Dat Quoc Nguyen[1,3], Trevor Cohn[1] and Karin Verspoor[1*]

*Correspondence:
karin.verspoor@unimelb.edu.au
[1]School of Computing and Information Systems, The University of Melbourne, Melbourne, Australia
Full list of author information is available at the end of the article

Abstract

Chemical patents are a commonly used channel for disclosing novel compounds and reactions, and hence represent important resources for chemical and pharmaceutical research. Key chemical data in patents is often presented in tables. Both the number and the size of tables can be very large in patent documents. In addition, various types of information can be presented in tables in patents, including spectroscopic and physical data, or pharmacological use and effects of chemicals. Since images of Markush structures and merged cells are commonly used in these tables, their structure also shows substantial variation. This heterogeneity in content and structure of tables in chemical patents makes relevant information difficult to find.

We therefore propose a new text mining task of automatically categorising tables in chemical patents based on their contents. Categorisation of tables based on the nature of their content can help to identify tables containing key information, improving the accessibility of information in patents that is highly relevant for new inventions.

For developing and evaluating methods for the table classification task, we developed a new dataset, called ChemTables, which consists of 7,886 chemical patent tables with labels of their content type. We introduce this data set in detail. We further establish strong baselines for the table classification task in chemical patents by applying state-of-the-art neural network models developed for natural language processing, including TabNet, ResNet and Table-BERT on ChemTables. The best performing model, Table-BERT, achieves a performance of 88.66 micro F_1 score on the table classification task.

Availability: A 10% sample of the ChemTables dataset[1] has been made publicly available, subject to a data usage agreement.

Keywords: Neural networks; Table classification, Chemical patents; Chemical patents

1 Introduction

A large number of chemical compounds are first published in patents. It takes on average one to three years for compounds disclosed in patents to appear in scientific literature [1], and only a small fraction of these compounds ever appear at all in publications. Therefore, chemical patents are an important resource for the development of information management tools to support chemical research. Information in patents is crucial for novelty or fact checking and understanding compound prior art [2].

Chemical patents typically present novel compounds, either specifying the chemical structure of compounds in the form of an image or through their systematic

chemical name in the text, for which state of the art name-to-structure tools such as OPSIN [3] and MarvinSketch [4] can be used to reliably generate the structure. However, to back up the inventive claims, patents also contain additional information related to these compounds — characterising them further, such as with physical or spectroscopic data (Figure 1 (a)), providing information related to their preparation (Figure 1 (b)), or exemplifying their claimed use through further information or numerical data. In addition to natural language text descriptions, such information is also presented in the form of tables and lists presenting the data in a compact and highly organized way. In fact, numerical data of very high interest to researchers, such as novel pharmacological results, are typically presented in this structured form [5].

In this context, manual excerption (or extraction) of key knowledge of compounds and their reactions from tables in chemical patents has been undertaken for large commercial chemical knowledge databases such as Elsevier's Reaxys® database[2], providing reliable and comprehensive data within the set of chemical literature and patents. The information provided by these commercial resources is of high-quality, but they are very expensive and time-consuming to build. As the number of new patent applications has been drastically increasing [6], it is infeasible for researchers and commercial chemical database providers to extract all of the relevant information manually from patent documents.

To reduce the time and effort needed for information extraction from chemical literature and patents, datasets and text mining approaches have been developed on a wide range of information extraction tasks, including named entity recognition and relation extraction [7, 8, 9, 10, 11, 12]. Most of these methods focus on processing plain text by leveraging state of the art Natural Language Processing (NLP) approaches, and tabular data is usually ignored or discarded, which causes significant loss in the amount of compound-related information that can be extracted from patents. A key reason for ignoring tabular data is the lack of existing publicly available chemical patent corpus with gold standard annotations on tables.

In this paper, we therefore present a novel dataset – the ChemTables corpus – consisting of tables extracted from chemical patents augmented with gold standard annotations of semantic types reflecting the key content of each table. This dataset enables further research in information extraction from chemical tables. We make a 10% sample of the full dataset publicly available [13].

It is challenging to develop text mining approaches for extraction of information from tabular data in chemical patents. Firstly, the number of tables and the average size of tables in chemical patents are much larger than in other contexts such as the web. For example, in the Web Data Commons Web Table Corpus 2012[3] [14] which consists of tables crawled from the web, the average number of rows is 12.41, respectively, whereas in our ChemTables dataset, the average number of rows is 38.77, 3 times more than in web tables. Thus, although a wide range of methods and datasets for text mining in web tables have been proposed, the performance of these methods might be compromised when applied to tables in chemical patents.

[2]Reaxys® Copyright ©2020 Elsevier Limited except certain content provided by third parties. Reaxys is a trademark of Elsevier Limited. https://www.reaxys.com
[3]http://webdatacommons.org/webtables/

Furthermore, as shown in Figure 2, tables containing different types of information are often structured differently and not all tables contain valuable data relevant to key chemical compounds described in the patents. Therefore, it may be most effective to develop different approaches for extracting information from tables that are specific to the semantic type of the table. This means that table classification and identification of tables containing valuable data are fundamental steps to enabling high quality information extraction from tabular data within chemical patents.

In addition to introducing the ChemTables data set, we provide here an empirical comparison of several strong baseline approaches to table classification using this corpus, including conventional machine learning models based on the Naïve Bayes (NB) and Support Vector Machine (SVM) algorithms, as well as neural models TabNet [15], ResNet [16] and Table-BERT [17]. The experimental results show that all neural methods outperform the conventional machine learning baselines. Among the three neural models, Table-BERT produces the best classification performance which we ascribe to the power of pre-trained language models. We also examine potential issues specific to classifying chemical patent tables, and finally identify directions for further improvements in the classification performance.

2 Related Work

In this section, we summarise previous efforts to apply machine learning methods to tabular data, including table layout classification (section 2.1), table question answering (section 2.1) and table fact verification (section 2.3) and discuss how methods developed for these tasks can guide our approach to semantic classification task for chemical patent tables.

2.1 Layout classification of web tables

Tables can be extracted from HTML format web pages by extracting content from elements surrounded by a `<table>` tag. These tables are structured with different layouts, such as vertical lists or matrices. A layout classification task can be defined, which aims to automatically identify the layout category of given web table (e.g. horizontal relational, vertical relational, matrix). We can determine the position of table headers and data cells more accurately if the layout of table is known. This task is fundamental for downstream table processing tasks. For example, in relation extraction, the column/row headers are often the most informative clue for identifying relations between two data cells in the same column/row. This task is challenging as web tables from diverse sources can have a huge vocabulary, making heuristic methods infeasible. Rule-based table classification methods based on selected keywords/tags often provide high recall but lack precision [18].

Here, we review several methods and datasets related to this task; these methods are directly relevant for our problem of classifying tables and several will be used as baseline methods in our experiments.

TabNet [15] is a supervised learning model for web table layout classification, which made one of the earliest attempts to apply neural networks for image processing on understanding table structures. It uses a Long Short Term Memory (LSTM) [19] network method to encode the sequence of tokens (words) within each

table cell with embedding vectors. The encoded table is then treated as an image and fed into a Residual Network [16] to derive a latent representation of the table. A linear transformation followed by a soft-max function is applied on the latent representation for generating a probability distribution over all classes.

This model and the baselines it compares to are evaluated on a web table dataset, built by extracting tables from top 500 web pages which contain the highest numbers of tables in a subset of the April 2016 Common Crawl corpus [20]. Tables in this dataset are categorized based on the logical structure of the table, such as *Vertical Relational* and *Horizontal Relational*.

The experimental results show that TabNet outperforms baseline models which are based on Random Forest with handcrafted features [21, 18, 22], and another neural model of bidirectional HAN—Hierarchical Attention Network [23]. This work also shows that an ensemble of 5 TabNets also outperforms an ensemble of 5 HANs and bidirectional HAN. This work shows that adapting models designed for image classification to tables in which cells are encoded as vectors of uniform size can outperform non-neural models with hand-crafted features engineered specifically for the table layout classification task.

TabVec Unsupervised methods have also been developed for table layout classification. TabVec [24] models the semantic of tables using a vector space model based on random indexing [25]. Four different types of contexts are used for learning word vectors, including the text within each table cell, text in column/row headers, text in adjacent cells and the text surrounding the table. Based on the idea that each cell in the table represents the same concept, the cell vector is then calculated by taking the median of word vectors of all tokens in the content. Instead of proposing a specific label set for annotated tables, only a general concept of table types is defined. In this work, a table type is defined as the way how different concepts are organized in a table, hence can be measured by calculating semantic consistency across rows, columns and the entire table. The semantic consistency is grounded by taking the deviation from mean and deviation from median at both row/column and table levels. The derived deviation vectors are then concatenated to form the final table vector representing the semantic of the table.

Since no label set is provided in this task, a K-Means clustering is applied on the vector representations of tables in the dataset. The label of each cluster is manually assigned by users, which eventually results in a label set similar to [15]. The proposed TabVec method is evaluated on 3 web dataset extracted from specific domains of human trafficking advertisements, fire arms trading, and microcap stock market. In addition, a random sample of July 2015 Common Crawl corpus is also used for a comparison with other methods which focus on generic domain only. The experimental results show that the proposed method TabVec significantly outperforms TabNet on all 4 datasets, indicating that contextual information can be particularly helpful for learning table semantics. Table2Vec [26] uses a similar idea as TabVec, except that the embeddings are trained using the Word2Vec skip-gram model [27], and used for row/column population and a table retrieval task instead of classification.

To classify the content of tables in chemical patents, we focus on interpreting the semantic information instead of structural information within tables. Therefore,

although there are methods developed for web table layout classification, they will need to be adapted for a more semantic classification task and specifically evaluated in the target context of chemical patents.

2.2 Table question answering

Question answering under active research in the NLP community as it requires detailed comprehension of both questions and documents where the answers may lie. Research in this area mainly focuses on answering questions based on unstructured text only. However, in real world scenarios, tables are usually used for presenting important data which tend to be frequently asked in articles. To build a well-rounded question answering system, the ability to extract answers from content within tables is needed. In the table question answering task, the goal is to answer a given question using data within a table.

Compositional Semantic Parsing [28] presented the WIKITABLEQUESTION dataset, consisting of 22,033 natural question-answer pairs over 2,108 tables extracted from Wikipedia pages. In this work, a novel method based on semantic parsing for table QA is proposed. The tables are first converted to knowledge graphs in which table cells are entity nodes; table rows are row nodes; table columns are directed edges from the row nodes to entity nodes of that column. Based on the knowledge graph converted from table, the questions are parsed to a set of logical forms. The logical forms of questions can be executed on the table knowledge graph as queries to retrieve the answer.

The authors proposed a novel semantic parsing method *Floating Parser* to address the difficulty of table-based parsing, such as the mismatch between words and utterance/predicates. This parser uses a relaxed lexical rule for anchoring predicates in logical forms to tokens by replacing the fixed span in chart parsing by a floating cell which only restricts the category and size of logical form.

The logical forms derived from tables and questions are fed into a log-linear model which optimizes the probability of the selected logical form retrieving the correct answer after execution on the table knowledge graph. Features used in this model include word n-grams in both question and table, the headword of the question (e.g. what, who, how many, etc) and the type of the answers (e.g. NUM, DATE, ENTITY) retrieved by using the logical form-table pair.

[29] proposed a CNN-based neural method for table QA task. In this approach, logical forms are first extracted using the same method as [28]. Then the logical forms are naturalized to a plain textual representation by applying a novel algorithm which recursively traverses the $\lambda-DCS$ logical form derived from the previous step.

The GloVe vectors of tokens in the naturalized logical forms are feed into 2 jointly trained CNN for obtaining sentence-level embeddings. The final answer are then selected based on the neural similarity between the sentence embeddings of the logical form and the question. This work also evaluated several ways to calculate the similarity of sentence embeddings on this task, including dot-product, Bi-Linear (BILIN), and a fully connected Feed-Forward network (FC) applied on the concatenation of 2 sentence embeddings. The experimental results show that using the

weighted average of BILIN and FC similarity and the ensemble of 15 models results in a performance exceeding [28] by 1.6% absolute accuracy. This work shows that neural models using semantic information from pre-trained embeddings with natural language input can produce better performance than a model based on logical forms and hand-crafted features.

[30] proposed a neural encoder-decoder architecture for semantic parsing in tables. In addition to the word embeddings of each token in the question, an entity linking embedding is also introduced in this model. To construct the entity linking embeddings, similarity between pairs of entity mentions in the knowledge graph and tokens in question is measured. The entity linking score consists of two parts, the similarity between word vectors of entity and token in questions and the output of a linear classifier built on hand-crafted features such as exact/lemma match, edit distance and NER tags. The entity linking score across all entities of the same type is then fed into a softmax function. Based on the resulting probability distributions, the weighted sum of embedding vectors of all entities forms the final entity linking embedding. To avoid the problem of ignored type constraints imposed by previous neural semantic parsers, strong type constraints are applied by only allowing the decoder to generate results in a grammar that guarantees well-formed logical forms.

The experimental results show a significant improvement based on previous state-of-the-art results [28, 29]. An absolute improvement of 7.2% in accuracy is observed comparing the CNN-based approach proposed by [29], showing that generating logical forms using a neural parser can produce logical forms more accurately than chart parser optimized for semantic parsing in tables.

In summary, most table question answering methods take the same strategy as knowledge base question answering in which tables play the role of the knowledge base. In these methods, tables and questions are first converted into logical forms by semantic parsing, and then the answer is extracted by querying the table. Since pre-defined logical forms and rules are required to perform table semantic parsing, such constraints might cause unexpected loss of information during the conversion process. In table classification, our objective is to obtain an accurate semantic representation of the table. Since there is no need for pairing questions with table contents, semantic parsing may not be the ideal way to extract semantic information from tabular data as there is some loss of information.

2.3 Table fact verification

Similar to table question answering, table fact verification also takes as input a table and statement pair. However, instead of extracting answers from the table, table text entailment models seek to determine whether a statement is true based on the data presented in a table.

Table-BERT [17] captures contextualized table semantics by applying BERT [31] on the concatenation of linearized tables and the statement to be verified. In this work, two different approaches to linearization are proposed. The first approach, serialization, simply concatenates content of all table cells with *[SEP]* (a special token which separates different sentences in BERT's input). Under this setting, the

position count of each token is reset to 0 at the beginning of each cell. The second approach, naturalization, uses a template filling method in which the content of each cell is enriched by adding the column and row headers.

The concatenation of linearized tables and the statement is then fed into a pre-trained BERT for extracting semantic representations. The same method for extracting sentence-level semantic representation in BERT is used for this task. The derived representation of tables will be fed into a linear layer with a binary classification objective to make a prediction on whether the table supports the given statement.

The experimental results show that Table-BERT performs significantly better than strong baselines, including a weakly-supervised latent program analysis approach (similar to [32]), showing that semantic information captured by pre-trained language models can improve upon semantic parsing. Moreover, pre-trained language model-based approaches also require less effort in adapting semantic parsers designed for other tasks to tabular data. Among different settings of Table-BERT, the naturalization approach outperforms the serialization approach by a large margin since more context can be incorporated for each table cell.

Table-BERT is capable of encoding the combination of tabular data and the statement into vectors that can be used as features for classification tasks, which means that Table-BERT can be used as a semantic encoder for tables, and especially if we use flattened tabular data as input. In the table verification task, the semantic representations of both table and the statement need to be obtained to identify whether the table entails the statement. Since our table classification task also requires semantic representations of tables, it is worth investigating whether Table-BERT can be adapted to table semantic classification task using only tables as input.

2.4 Bridging quantities between tables and texts

The goal of this task is to identify coreference between quantities in tables and main body of the documents. Besides direct mentions (i.e. values in a table are identical to those mentioned in body text), aggregations of values also need to be identified in this task. Hence, the first step of this task is to extract all pairs of quantity mentions in tables and texts. Then pairs which co-refer can be identified through a binary classification task. This task requires modeling of table content within cells as well as consideration of the broader document content outside of the tables.

ExQuisiTe [33, 34] proposed a multi-stage system for linking numeric mentions in tables and texts. Firstly, all quantity mentions are extracted from both and tables and texts. In this stage, aggregation of values such as total, difference, percentage and change ratio between values in the same column/row are also calculated and add to the collection of quantities.

In the second stage, all possible pairs between quantities are enumerated and fed into a Random Forest-based classifier to eliminate the pairs that are not likely to be relational. Features used for building this classifier include context features, such as word and phrase overlaps and quantity features such as relative difference between values, unit match and aggregation function match. After pruning, the remaining

candidate mention-pairs are classified based on whether the text mention is an aggregation value or a single-cell mention using indicator words, exact matches in tables and other context features.

Heuristic approaches are also considered in this work. Value and unit mismatches are discarded after ensuring mention type matches. Finally, the top-k candidate pairs will be selected for the next stage of processing. In this system, the selection of value k depends on the distribution entropy of the confidence score returned by the classifier for pairs with the same text mention.

CCTR-83 [35] is a corpus specifically built for information extraction tasks in tables. It consists of 83 tables in 43 biomedical literatures. Tables in this dataset can be divided into 2 groups of roughly equal sizes by their topics which are human cancer and mouse genetics (43 and 40 respectively). When constructing the dataset, similar cells are first grouped into *cell groups* and the terms which correspond to concepts in UMLS-NCI schemes will be annotated. The type of cell groups are decided based on the common parents of concepts presented in this group. Finally, suggested relations between cell groups will be generated based on matches to UMLS Metathesaurus and UMLS SN, which the annotator will decide whether to accept it as an annotation or not. In this dataset, tables are annotated by experts with post-graduate degree in Biology. Thus, this dataset has a very high inner-annotator agreement of Kappa value 0.88, 0.87, 0.82 for concept, cell type and relation annotations, respectively. However, owing to budgetary constraints, it is small in size and may not be sufficient for complex machine learning methods. In this dataset, it is possible to do classification based on semantic types by using topic of tables (human cancer and mouse genetics) as labels. However, the number of table instances is too small for any supervised machine learning approach.

3 ChemTables Corpus

In this section, we present the annotation guidelines and data preparation process of our ChemTables dataset (Section 3.1), general statistics of the dataset (Section 3.3), and more detailed dimension-level (Section 3.4) and cell-level (Figure 6) statistics. Finally, we present the standard data split we use for evaluating our table semantic classification methods 3.6.

3.1 Corpus definition

To enable automatic determination of semantic type and relevancy of tables in chemical patents, we have constructed a new table dataset named ChemTables. In contrast to WikiTableQuestions and other datasets that are built from web crawls, our ChemTables corpus focuses exclusively on tables in chemical patents, and makes use of a categorization scheme directly relevant to chemistry.

For the purpose of identifying containing data valuable to research, the taxonomy of tables must be well defined so that the relevancy of tables can be assessed based on their label. In order to reflect the categories of content that are of interest for researchers, we adapted the categorization system for facts in patents from Reaxys® [36], a highly curated database of experimental facts derived from relevant literature including patents. The storage of data follows strict organising principles. The same organisation is used in the manual excerption process. The main purpose

of patent excerption for Reaxys is to identify and capture significant data about compounds and their related reactions and facts in a reliable and comprehensive way. Reaxys has had positive reception from chemists as it can help accelerating the search of property information of chemical compounds and provides the ability to compare these data from different sources [37, 38]. Hence, we suggest that the key data types derived from Reaxys for the ChemTables labelling guidelines represent the most typical and important types of data in chemical patents.

Following the Reaxys Excerption Guideline, in our ChemTables dataset, tables in chemical patents are categorized based on their semantic types as listed in Table 1. We identified the 6 most relevant types of information to chemical research (rows 1 to 6 in Table 1). Since there can be different types of data presented in the same table, a single table can be assigned to multiple categories within the annotation scheme. However, if a group of data in the table cannot be categorized into any class in the Reaxys schema, only one of three out-of-schema category labels (`COMPOSITION`, `PROPERTY`, `OTHER`) will be assigned following the order of priority shown in the table.

3.2 Corpus construction

We first sampled 1K XML-formatted patents from the European Patent Office and the United States Patent and Trademark Office. Tables are automatically extracted from these patents using a toolkit developed internally. To achieve better downstream task performance, after extraction, we tokenize the textual content in each cell with a chemical tokenizer which is a component of a chemical named entity recognition (NER) tool `OSCAR4` [39]. This process results in a total of 7,886 tables consists of 3 million tokens. We selected 1K patents with in average more than 8 tables per patent. In the ChemTables dataset, we store all tables extracted from a patent document in a different worksheets in a single Excel (`.xlsx`) file. The original patents in both XML format and PDF format are provided with the table file, which means that tables in this corpus can be linked back to its context in the original patents by matching the caption.

This table dataset was sent to 3 groups of Reaxys excerptors who hold at least a Master's degree in chemistry. The excerptors have years of experience working on Reaxys excerption, thus are very familiar with relevant topics and concepts in the chemical patent table annotation task. Two annotator groups (Annotator 1 & 2 in Table 2) annotated the ChemTables dataset independently while the third annotator group works as to merge annotations from different annotators and make the final decisions in the case of disagreement. We then use the harmonized set as the final gold standard dataset. Here we use F_1 rather than Kappa score as the distribution of labels is highly skewed [40]. The inner-annotator agreement (IAA) scores in terms of F_1 score are computed by comparing the annotations of Annotators 1 & 2 against the gold set.

As shown in Table 2, both annotator groups achieve a high $> 80\% F_1$ score on average. This confirms that the annotation of ChemTables dataset is of high quality to domain experts. Among all labels, `SPEC`, `PHYS` and `IDE` gets 80+ F_1 score in both annotator groups. Since spectroscopic and physical data are more of interest to chemical researchers, it is not surprising that the annotation of these

3 labels has higher-level of consensus than the others. For labels with relatively low IAA scores, `OTHER` and `PROPERTY` are much lower than other categories. Since these labels are only used when the table is considered "Out of Reaxys Scheme", it becomes difficult to disambiguate these tables with others tables which are within the "Reaxys Scheme" while containing a similar type of information. The label `CHEM` also gets very low IAA scores since there are only 14 instances in the dataset.

3.3 Annotation Statistics

In this section, we show statistics over annotations in the dataset. Figure 3(a) shows the count of annotations per semantic type. Among 8,846 annotations, `SPECT` is the most common label as 23.2% of all tables in ChemTables contain spectroscopic data, while only less than 1% (14 instances) tables are annotated as `CHEM`.

As our annotation guideline allows one table to be annotated with different labels, we also show the statistics of co-occurrence between labels. As shown in 3(b), most overlaps occur between `RX`, `PHYS` and `SPECT`, indicating that physical, spectroscopic and reaction related data are usually presented in the same table.

3.4 Dimension-level Statistics

In this section we show statistics related to table size. Merged cells which takes up the space of multiple data cells are frequently used in headers that summarise the semantic of multiple columns/rows. During the extraction process, information about merged cells is not preserved. Thus, only one of the original cells which forms the merged cell is used to store the content of the merged cell and other cells are left empty. In addition, rows which are shorter than the longest row in the table are padded with empty cells to ensure all rows in the same table have the same length. We measure the size of tables by taking the product of number of rows and number of columns. Figure 4 shows average size of tables per semantic category. Tables with label `IDE` are on average larger than other table types as they are usually enumerations of compound identifications while tables labeled with `COMPOSITION` take least space in average.

We also perform fine-grained table statistics on row and column levels. As Figure 5(a) shows, the range of table heights in the dataset is relatively wide, from less than 5 rows to more than 1,000 rows. This is not surprising since patents usually contain many details, as the authors tend to maximize the scope of claims of their inventions. However, as Figure 5(a) shows, only a small fraction of tables in the dataset (approx. 10%) have more than 100 rows, the height of approximately 80% of the tables in the dataset is smaller than 35.

The width of tables in patents shows completely different patterns than their heights. The range of numbers of columns is very narrow comparing to that of rows, with the width of 99% of tables falling in the range from 1 to 15 while the maximum number of columns is 32. This implies that most tables in patents grow vertically (i.e. column headers control the content type of each column while rows represent different instances).

3.5 Cell-level Statistics

In this section we provide cell level statistics focusing on the length of text (number of tokens) in table cells. Statistic results in Figure 6 shows that most of the cells

in tables are generally short as 73+% of cells contain only 1 token (usually data in a single number/word) and 97+% of cells contain fewer than 20 tokens. Cells in tables with `SPECT` label have the longest average length (approx. 8 tokens per cell) while tables labeled with `PHARM` have the shortest average length of cells (< 2 tokens per cell).

The caption of table usually contains long text summarising the content of the table. The caption of every table in ChemTables dataset is also included at the first cell of the first row. If a table has no caption, we use "Unlabelled table n" as the caption, in which n is a numerical identifier. Thus, we also present cell-level statistics over the captions as it tends to have different property comparing to other cells. The captions of 90% of tables in the dataset contain 2-4 tokens, indicating that the table captions from chemical patents are much shorter and containing much less information than tables in scientific literature. These captions are usually only table identifiers, such as "Table 2" or "Unlabelled table 1". We also find that the average title length is very similar across all table categories. Noticeably, tables with label `PHARM` has the shortest average cell length but the longest average caption length, implying that information in pharmacological tables is usually more complex and thus needs longer text to summarise.

3.6 Data split for evaluation

As shown in Figure 3(b), the number of instances with multiple labels is low (10%). Among all tables with more than one type of category annotations, 82% instances are the combination of `SPECT` and `PHYS`. Hence, when pre-processing the dataset, we merge tables with both `SPECT` and `PHYS` labels into a new category `SPECT|PHYS`. For the remaining multi-label instances, we convert those into single-label instances by choosing the most frequent label. Besides, label `CHEM` has only 8 supporting table instances, it is not likely to be sufficient for training machine learning models. Thus, we ignore this label and their table instances in our experiments on table classification.

We release a sample[4] of the training set which contains 10% of the entire dataset [13]. This sample dataset is created by taking two folds from a 20-fold stratified split. In section 5.4, we explore the effect of training set size on classification performance. The 10% of the entire dataset used in this section is identical to this sample set that we will release.

4 Experimental Methodology

We present our empirical study of table semantic classification over our ChemTables dataset. We first outline the non-neural baseline methods (Section 4.1), and then introduce the neural methods (Section 4.2), starting from networks take 2-dimensional tabular data as input (Sections 4.2.1, 4.2.2 and 4.2.3) and pre-trained language model which takes flattened tables as input (Section 4.2.4). We also detail our evaluation metrics in Section 4.3.

[4] ChemTables Dataset Sample http://dx.doi.org/10.17632/g7tjh7tbrj.1

4.1 Baseline models

In this work, we compare state-of-the-art neural methods on table classification with two non-neural baseline models Naïve Bayes (NB) and Supported Vector Machine (SVM) which use bag-of-words features. For each table, texts in its cells are tokenized using the OSCAR4 [39] tokenizer. We then calculate the value of each bag-of-words feature by using the TF-IDF weighting scheme. Similar to neural models, we also explore the effect of input size for baseline models. We use the results with optimal input size to compare with other models.

4.2 Neural network-based models

Similar to images, tables are naturally 2-dimensional structured data, in which each table cell corresponds to a pixel in images. It thus seems possible to attempt generalizing well-developed methods in computer vision to the context of table processing. However, there are some challenges for applying computer vision methods to tables. One major obstacle is the difference between pixels and table cells. Pixels, in conventional image datasets, are represented in RGB encoding. Hence the size of the vectors representing pixels is the same. However, text in table cells could contain a various number of tokens. Thus, an embedder is needed to encode the textual content in table cells into vectors having a uniform number of dimensions. After embedding the table into a feature map, an image classification model can then be used as a decoder to determine the semantic type of a given table. In this work, we evaluate a residual network based neural model TabNet [15] that has been applied on web table layout classification task, and a more complex variation of a residual network [16] as well as a BERT-based model [17].

In an image dataset, the size of different images is usually the same, whereas tables are often vary substantially in size. Hence, to be able to apply image classification models on tables, we need to pad or truncate all tables in the dataset to the same size. We will explore the effect of input size on table classification performance in Section 5.2. Since the size of more than 80% tables in the dataset are within 32 by 32 as shown in 5, we only explore input size within this range.

4.2.1 Cell Embedder

We employ a Bi-LSTM encoder with the combination of word embedding w and character-level word representation c as shown in Figure 7 (a). We use the ChemPatent word embeddings from a skip-gram model pre-trained on chemical patent corpus [11]. We use fixed weights for tokens presented in the vocabulary of the ChemPatent word embeddings while using trainable weights for all out-of-vocabulary words that appear more than 5 times in the entire table training set. For character-level word embeddings, we employ CNN-based character-level word representations [41] with kernel size of 3.

4.2.2 TabNet

In TabNet [15] a Bi-LSTM encoder is applied on word embeddings of each token in table cells (w/o character-level word representation). Then, instead of concatenating the hidden output from both directions of a Bi-LSTM encoder, TabNet adopts the word-level additive attention used in [23] to calculate attention weights a_t for hidden

states $h_{i,j,t}$ of each token t in cell at the i-th row and j-th column and take the weighted average sum of them as the final cell-level representation.

The attention is formulated as

$$\begin{aligned} \mathbf{u}_{i,j,t} &= \texttt{tanh}(\mathbf{W}_w \mathbf{h}_{i,j,t} + \mathbf{b}_w) \\ a_{i,j,t} &= \frac{\texttt{exp}(\mathbf{u}_{i,j,t}^\top \mathbf{u}_w)}{\sum_t \texttt{exp}(\mathbf{u}_{i,j,t}^\top \mathbf{u}_w)} \\ \mathbf{c}_{i,j} &= \sum_t a_{i,j,t} \mathbf{h}_{i,j,t} \end{aligned} \tag{1}$$

where u_w is a trainable context vector and $\mathbf{W}_w$ is a projection matrix which maps the hidden states to the same dimensionality as the context vector.

The cell-level representation, $\mathbf{c}_{i,j}$, is then fed as an input to a 3×3 convolutional layer which is followed by 3 consecutive residual blocks. The residual blocks are illustrated in Figure 7(b), where x_l is the output of the l-th layer of residual block.

4.2.3 TBResNet

In TabNet, although residual blocks are used for encoding tables, the size and depth of the model is still not comparable with state-of-the-art deep residual networks for image classification. Hence, to investigate whether the effect of increasing model complexity on table classification performance, we use a 18-layer ResNet [16] for table classification. TBResNet takes the feature map generated by the cell-level embedder and feed it as input to a convolution layer, which down samples the output to match the input dimension of ResNet18 [16].

4.2.4 Table-BERT

Table-BERT [17] is proposed for table fact verification task. The goal of this task is to verify whether the table entails a textual statement which describes facts in the table. Different from TabNet and TBResNet, BERT is a language model pretrained on massive on annotated plain text corpora. Hence, tables must be flattened into sequences of sentences before being used as the input to BERT. Table-BERT takes BERT pre-trained on unstructured text as starting point and then uses the concatenation of the flattened table and the statement as input to fine-tunes BERT on a binary classification objective. There are two different methods for flattening tables proposed and evaluated in this work [17], namely linearization and naturalization.

Linearization The linearization approach simply takes the concatenation of tokens within all cells in the table to form a paragraph which will be used directly as the input to BERT. The cells are added to the paragraph following a top to bottom and left to right order. Each row here is regarded as sentence and will be separated by a '.' which represents the end of sentence. There is no separation added between content from neighboring cells in the same row.

Naturalization is a template filling approach. In this approach, row and column headers of each row are incorporated into each cell, which make the sentence structure of the flattened text more natural and provide extra semantic information

about the data within each table cell. Comparing to tables from Wikipedia where the determination of column and row header is relatively trivial, the heterogeneity of chemical patent tables makes it difficult to accurately locate the headers. Here we assume that the first non-repeating row in a table is the column headers (here, captions may be repeated in the first row of patent tables), and the first column as row headers. In this approach, the column headers and row headers are incorporated within each cell (e.g. "The name of row *[row_id]* is *[row_header]* and its *[column_header]* is *[cell_value]*.") instead of being added to the paragraph individually.

Figure 8 shows examples of the two approaches for flattening tables. Comparing to the WikiTableQuestion corpus on which Table-BERT was first evaluated [17], tables in the ChemTables dataset do not have explicit annotation of the location of header row/columns. Therefore, we take the first non-empty row under the table caption as row headers and the first column except the table caption as column headers. Then, we fine-tune BERT on for the table-classification task.

The table fact verification task is a binary classification task and the input of which is a pair of statement the table. Hence, to adapt this model to our table semantic classification task, we use the flattened table only as input and change the size of output layer from 1 to the number of labels in the ChemTable dataset. Since flattened table are usually longer than the input size limit of BERT which is 512 sub-tokens. We explore the effect of length limit of flattened table on the classification performance and the results of which are presented in 5.2.

4.3 Evaluation Metrics

We use a stratified 60:20:20 split for training, development and test set. We use micro F_1 score across all classes over development set as indicator for performance improvement. All models in this paper are trained with 50 epochs in maximum, and an early stopping is applied if there are no overall micro F_1 score improvement observed after 5 epochs. Macro and weighted average F_1 scores across all labels are also reported.

5 Results

In this section, we present our main experimental results comparing different neural methods on table semantic classification with our baselines (Section 5.1). We then discuss how change in input size impacts the performance of neural models (Section 5.2), and the effect of different pre-processing strategy (Section 5.3). We show that the tables in ChemTable dataset is sufficient to train state-of-the-art machine learning methods. We also provide analysis on error cases and propose possible ways to further improve classification performance (Section 5.5).

5.1 Main Results

Table 3 shows table classification performances for baseline models (Naive Bayes, SVM) and neural network models (TabNet, TBResNet and Table-BERT). The first 9 rows show the micro F_1 score per each semantic type while the last row shows the overall performances in micro-average F_1 scores.

For baseline methods, the SVM significantly outperforms Naive Bayes in every metric. Comparing the baseline approaches with the neural based methods, all neural based methods outperform SVM by at least 2 points in the micro-average F_1 score. Among 3 neural models, Table-BERT achieves the best overall performance, with TBResNet second.

Regarding performances per semantic types, labels with more than 500 instances in the entire dataset achieve nearly 90 Micro F_1 score, except the label `OTHER`. For the least frequent labels `PHYS` and `PROPERTY`, Table-BERT obtains micro F_1 scores of about 80 which is much higher than those of other models.

The label `OTHER` shows a different pattern comparing to the remaining labels. With 1184 instances in the entire dataset, the best Micro average F_1 performance on this label is 76.47, achieved by Table-BERT, which is approximately 15 points less than other labels having 1000+ instances in the dataset.

By comparing the performance per class in Table 3, we found that advantage in performance for Table-BERT mainly comes from under-represented classes which have less than 500 instances in the entire dataset. This suggested that a table classification task can also benefit from the semantic knowledge that the pre-trained language model have learned from massive amount of un-annotated training data.

5.2 Effect of input size

As described in dimension-level statistics of the dataset in section 3.4, tables in our ChemTables dataset are of various sizes and the range of the table size is wide. To adapt neural models from image classification, table instances within the same mini-batch need to be padded to the same size. For pre-trained language model such as Table-BERT, a limitation on input sequence length also applies. Hence, it is important to determine the input size which can help produce the best overall classification performance.

Table 4 shows the test performance of all models when we use no more than the first 3 rows as input (all columns are used in this set of experiments) to explore whether the semantic type of table can be determined by caption and row headers only. The experimental results shows that, with only the first 3 rows, classification performance drops at least 20 F_1 score in all 3 metrics comparing to the models using optimal input size. Hence, although semantic information in captions and headers are denser, including the body of the table can still help significantly improve the understanding of table semantics.

5.2.1 TabNet and TBResNet

Figure 9 shows classification performance in Micro F_1 score under different combination of input height (number of rows) and width (number of columns) from the set [5, 10, 15, 20, 25, 32]. Figure 9(a) shows that for TabNet the best performance is achieved when using 15 rows and 25 columns, while as shown in Figure 9(b) for TBResNet, the optimal performance is achieved when using 20 rows and 32 columns. For models taking 2-dimensional table data as inputs, the classification performance does benefit from increasing model complexity as TBResNet outperforms TabNet in all 3 metrics on the overall performance. From the observation from Figure 9(a) and (b), with increased model complexity, TBResNet is able to

learn more knowledge of table semantics from larger input sizes and hence results in higher classification accuracy.

As shown in Figure 9(a) and (b), for TBResNet, the difference between the highest and lowest F_1 score is larger than that of in TabNet. We can also observe that the performance of model drops faster than TabNet when as input size deviating from the optimal value. This shows that as the model complexity increases, the performance of model is more sensitive to the change in input size as complex models generally need more data to be trained sufficiently.

Results in Table 4 show that the model performance when only the first few rows are used for training. We can observe that when the size of the input table is ≤ 3 rows, TabNet outperforms both TBResNet and Table-BERT. This confirms that the less complex model TabNet is not as sensitive to input size comparing to TBResNet and Table-BERT.

5.2.2 Table-BERT

The maximum length of input for Table-BERT is 512 sub-word tokens. However, the number of words within a table can easily exceed the length limit. Therefore, we also explore the effect of input length on classification performance. As shown in Figure 10, Table-BERT using both Naturalization and Linearization strategies achieve best performance in all 3 metrics when input size is limited to 320. Further increasing the input length does not help producing better results as more noise is introduced.

5.3 Effect of flattening strategy

The complexity in table structure makes it difficult to accurately determine the header rows and columns for chemical patent tables. For the naturalization approach, we aim to convert table into a more human readable format, which is closer to the training data of BERT. However, when the headers cannot be identified correctly, incorporating wrong headers might introduces extra noise to the input which has negative effect on the classification performance.

As shown in Figure 10, the Linearization approach outperforms the Naturalization approach by at least 0.5 F_1 score at all input sizes, showing different pattern comparing to the table text entailment task in which the Naturalization approach does significantly better [17]. This reconfirms that it is difficult to naturalize tables accurately by using heuristic-based methods when the table structure is complex.

5.4 Effect of train set size

To investigate whether the size of ChemTable is enough for developing classifiers based on deep neural networks, we plot the micro F_1 score of the 3 state-of-the-art machine learning models trained on dataset of different sizes. In this experiment, we created a 20-fold stratified split (5% of entire dataset per fold) and use the evaluation and test set in the standard split (i.e. 20% of the entire dataset, respectively). As shown in Figure 11 (a), for all 3 models, improvement in micro F_1 score slows substantially after 35% of the entire dataset has been used for training. This suggests that the data in ChemTable dataset is sufficient to develop more complex deep learning models. Note that the 10% subset evaluated here corresponds to the dataset that we have made publicly available, as mentioned in 3.6.

Figure 11 (b) shows per label performance of Table-BERT in F_1 scores. We can observe that label `SPECT`, `SPECT|PHYS IDE`, `PHARM` and `COMPOSITION` are easier to identify than other labels as Table-BERT can achieve ˜80 F_1 score when only 5% of entire dataset is used for training. Where as the learning curve of the out of scheme labels `PROPERTY` and `OTHER` are much steeper. Especially for `PROPERTY`, the F_1 is 40 when 5% data is used for training, which is 2 times lower than the aforementioned easier labels.

5.5 Error Analysis

In this section, we analyze the error pattern of different models using the confusion matrix of each category. For most classes, we can observe that large portions (> 50%) of confusions are between the label `OTHER`. This is because `OTHER` is only assigned when there is no other label within or without the Reaxys scheme matches with the content of the table. Hence, content of tables labelled as `OTHER` may have content which is partially close to other labels, but not sufficient to qualify as one of them. For label `SPECT|PHYS`, most confusions happens between it and its two components `SPECT` and `PHYS` as tables with label `SPECT|PHYS` have data of both type, but the proportion of these two types of contents may not be balanced.

We also find that there is considerably large amount of `PHARM` tables being classified as `OTHER`. Especially in the case of the latter the dis-accord is due to the consideration of the context in the annotated samples. The information provided directly in the table or its attached subsets is often somewhat cryptic and requires additional explanations. These are provided in the surrounding text paragraphs and are thus not considered by Table-BERT and TBResNet. The label `OTHER` has a very low inner-annotator agreement score, showing that human experts are also struggling on making the decision of whether there is content doesn't belong to any label. Hence it is not surprising that most of the confusions produced by our model are between `OTHER` and rest of label set.

The table in Figure 12 is a typical example: Whilst the table already lists the required species, chemical compound and effect, the text in the paragraph above the table clearly outlines that the experiment evolves around pharmacokinetics.

6 Conclusion and Future work

In this work, we have presented a new dataset ChemTables of tables extracted from chemical patents, with gold standard annotations on the semantic type of tables, which enables research on applications of deep learning methods on the task of classifying tables based on semantics. We also established strong baselines on this dataset with various machine learning methods, including non-neural methods and 3 neural networks models TabNet, TBResNet and Table-BERT. The experimental results show that all neural models outperform non-neural baselines. This indicates that machine learning models trained on ChemTables dataset can help identifying tables of interest to researchers automatically, hence reducing the amount of human effort needed in chemical patent excerption. Furthermore, this dataset can also be used for table representation learning, such as training unsupervised language models like BERT.

The best performing model Table-BERT achieves a strong performance of nearly 89 micro average F_1 score across all classes and shows a significant advantage on

relatively under-represented classes in the dataset. The 2-dimensional models show competitive performance given that they have not been pre-trained on any data other than the ChemTables dataset itself.

The error analysis showed that even the best performing neural models still fail to tell whether the pharmacological data in tables is relevant or not due to the lack of information from the surrounding text in the model input. Therefore, to solve this problem, it is also important to develop models which learn from both the table and its context.

Although the image classification models can learn local features from the convolutional layers, the relationships between cells (e.g. a row header that defines the data type of subsequent cells) are not captured. Hence, it may be interesting to combine sequential models with CNNs to mitigate this problem. The advantage of pre-trained language models observed in this task also motivates us to test their incorporation into models taking 2-dimensional data as inputs, such as TabNet and TBResNet, in future work.

Availability of data and materials
We release the ChemTables Sample Dataset on Mendeley data (http://dx.doi.org/10.17632/g7tjh7tbrj.1) which contains 10% of the entire dataset. This sample dataset is created by taking two folds from a 20-fold stratified split.

Competing interests
SA, CD, and CT are employed by Elsevier and have direct involvement with the development of the Reaxys® system referred to in this manuscript.

Funding
Funding for the ChEMU project is provided by an Australian Research Council Linkage Project, project number LP160101469, and Elsevier.

Author's contributions
ZZ: Pre-processed data and prepared code, performed the study, generated all figures, and wrote the first version of the manuscript. KV, TC: Conceived and designed the study and revised the manuscript. DN: Generated baseline results, pre-processed data. CD: Wrote annotation guidelines, prepared data, contributed to the background of the study. SA: Coordinated the annotation process and contributed to revising the manuscript. CT: Contributed to revising the manuscript. KV: Project leadership, obtained funding. All authors reviewed, revised and approved the final manuscript.

Acknowledgements
We acknowledge the work of Panagiotis Eustratiadis who contributed to the preparation of the ChemTables dataset while employed at Elsevier.

Author details
[1]School of Computing and Information Systems, The University of Melbourne, Melbourne, Australia. [2]Elsevier - Data Science, Life Science. [3]VinAI Research, Vietnam.

References

1. Senger, S., Bartek, L., Papadatos, G., Gaulton, A.: Managing expectations: assessment of chemistry databases generated by automated extraction of chemical structures from patents. Journal of cheminformatics **7**(1), 49 (2015)
2. Akhondi, S.A., Rey, H., Schwörer, M., Maier, M., Toomey, J.P., Nau, H., Ilchmann, G., Sheehan, M., Irmer, M., Bobach, C., Doornenbal, M.A., Gregory, M., Kors, J.A.: Automatic identification of relevant chemical compounds from patents. Database **2019**, 001 (2019)
3. Lowe, D.M., Corbett, P.T., Murray-Rust, P., Glen, R.C.: Chemical name to structure: OPSIN, an open source solution. Journal of chemical information and modeling **51**(3), 739–753 (2011)
4. TorchVision. https://chemaxon.com/products/marvin. Accessed: 08/09/2020
5. Milosevic, N., Gregson, C., Hernandez, R., Nenadic, G.: Disentangling the structure of tables in scientific literature. In: International Conference on Applications of Natural Language to Information Systems, pp. 162–174 (2016). Springer
6. Muresan, S., Petrov, P., Southan, C., Kjellberg, M.J., Kogej, T., Tyrchan, C., Varkonyi, P., Xie, P.H.: Making every SAR point count: the development of chemistry connect for the large-scale integration of structure and bioactivity data. Drug Discovery Today **16**(23-24), 1019–1030 (2011)
7. Akhondi, S.A., Klenner, A.G., Tyrchan, C., Manchala, A.K., Boppana, K., Lowe, D., Zimmermann, M., Jagarlapudi, S.A., Sayle, R., Kors, J.A., *et al.*: Annotated chemical patent corpus: a gold standard for text mining. PLoS One **9**(9), 107477 (2014)

8. Krallinger, M., Rabal, O., Lourenço, A., Perez, M.P., Rodriguez, G.P., Vazquez, M., Leitner, F., Oyarzabal, J., Valencia, A.: Overview of the chemdner patents task. In: Proceedings of the Fifth BioCreative Challenge Evaluation Workshop, pp. 63–75 (2015)
9. Wei, C.-H., Peng, Y., Leaman, R., Davis, A.P., Mattingly, C.J., Li, J., Wiegers, T.C., Lu, Z.: Overview of the biocreative v chemical disease relation (CDR) task. In: Proceedings of the Fifth BioCreative Challenge Evaluation Workshop, vol. 14 (2015)
10. Habibi, M., Weber, L., Neves, M., Wiegandt, D.L., Leser, U.: Deep learning with word embeddings improves biomedical named entity recognition. Bioinformatics **33**(14), 37–48 (2017)
11. Zhai, Z., Nguyen, D.Q., Akhondi, S., Thorne, C., Druckenbrodt, C., Cohn, T., Gregory, M., Verspoor, K.: Improving chemical named entity recognition in patents with contextualized word embeddings. In: Proceedings of the 18th BioNLP Workshop and Shared Task, pp. 328–338 (2019)
12. He, J., Nguyen, D.Q., Akhondi, S.A., Druckenbrodt, C., Thorne, C., Hoessel, R., Afzal, Z., Zhai, Z., Fang, B., Yoshikawa, H., Albahem, A., Cavedon, L., Cohn, T., Baldwin, T., Verspoor, K.: Overview of ChEMU 2020: Named entity recognition and event extraction of chemical reactions from patents. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Eleventh International Conference of the CLEF Association (CLEF 2020) vol. 12260. Lecture Notes in Computer Science, ??? (2020)
13. Zhai, Z., Druckenbrodt, C., Eustratiadis, P., Thorne, C., Akhondi, S.A., Nguyen, D.Q., Cohn, T., Verspoor, K.: ChemTables Sample: dataset for table classification in chemical patents. Mendeley Data (2020). doi:10.17632/g7tjh7tbrj.1
14. Lehmberg, O., Ritze, D., Meusel, R., Bizer, C.: A large public corpus of web tables containing time and context metadata. In: Proceedings of the 25th International Conference Companion on World Wide Web, pp. 75–76 (2016). International World Wide Web Conferences Steering Committee
15. Nishida, K., Sadamitsu, K., Higashinaka, R., Matsuo, Y.: Understanding the semantic structures of tables with a hybrid deep neural network architecture. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
17. Chen, W., Wang, H., Chen, J., Zhang, Y., Wang, H., Li, S., Zhou, X., Wang, W.Y.: Tabfact: A large-scale dataset for table-based fact verification. arXiv preprint arXiv:1909.02164 (2019)
18. Crestan, E., Pantel, P.: Web-scale table census and classification. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 545–554 (2011). ACM
19. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
20. April 2016 Common Crawl Archive. https://commoncrawl.org/2016/05/april-2016-crawl-archive-now-available/. Accessed: 08/09/2020
21. Cafarella, M.J., Halevy, A.Y., Zhang, Y., Wang, D.Z., Wu, E.: Uncovering the relational web. In: WebDB (2008)
22. Eberius, J., Braunschweig, K., Hentsch, M., Thiele, M., Ahmadov, A., Lehner, W.: Building the dresden web table corpus: A classification approach. In: 2015 IEEE/ACM 2nd International Symposium on Big Data Computing (BDC), pp. 41–50 (2015). IEEE
23. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1480–1489 (2016)
24. Ghasemi-Gol, M., Szekely, P.: TabVec: Table vectors for classification of web tables. arXiv preprint arXiv:1802.06290 (2018)
25. Kanerva, P.: Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. Cognitive computation **1**(2), 139–159 (2009)
26. Zhang, L., Zhang, S., Balog, K.: 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, pp. 1029–1032 (2019)
27. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a Meeting Held December 5-8, 2013, Lake Tahoe, Nevada, United States., pp. 3111–3119 (2013)
28. Pasupat, P., Liang, P.: Compositional semantic parsing on semi-structured tables. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 1470–1480 (2015)
29. Haug, T., Ganea, O.-E., Grnarova, P.: Neural multi-step reasoning for question answering on semi-structured tables. In: European Conference on Information Retrieval, pp. 611–617 (2018). Springer
30. Krishnamurthy, J., Dasigi, P., Gardner, M.: Neural semantic parsing with type constraints for semi-structured tables. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 1516–1526 (2017)
31. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: 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: Human Language Technologies, pp. 4171–4186 (2019)
32. Liang, C., Berant, J., Le, Q., Forbus, K.D., Lao, N.: Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 23–33 (2017)
33. Ibrahim, Y., Weikum, G.: Exquisite: Explaining quantities in text. In: The World Wide Web Conference, pp. 3541–3544 (2019). ACM
34. Ibrahim, Y., Riedewald, M., Weikum, G., Zeinalipour-Yazti, D.: Bridging quantities in tables and text. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE), pp. 1010–1021 (2019). IEEE
35. Shmanina, T., Zukerman, I., Cheam, A.L., Bochynek, T., Cavedon, L.: A corpus of tables in full-text biomedical research publications. In: Proceedings of the Fifth Workshop on Building and Evaluating Resources

for Biomedical Text Mining (BioTxtM2016), pp. 70–79 (2016)
36. Elsevier: Reaxys features and capabilities. https://www.elsevier.com/solutions/reaxys/how-reaxys-works. Accessed: 08/09/2020
37. Goodman, J.: Computer software review: Reaxys. J. Chem. Inf. Model. **49**(12), 2897–2898 (2009). doi:10.1021/ci900437n
38. Lawson, A.J., Swienty-Busch, J., Géoui, T., Evans, D.: The making of Reaxys – Towards unobstructed access to relevant chemistry information. In: The Future of the History of Chemical Information, pp. 127–148. American Chemical Society Publications, ??? (2014)
39. Jessop, D.M., Adams, S.E., Willighagen, E.L., Hawizy, L., Murray-Rust, P.: OSCAR4: a flexible architecture for chemical text-mining. Journal of cheminformatics **3**(1), 41 (2011)
40. Di Eugenio, B., Glass, M.: The Kappa statistic: A second look. Computational Linguistics **30**(1), 95–101 (2004)
41. Ma, X., Hovy, E.: End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1064–1074 (2016)

Figures

Figure 1 Example of tables in chemical patents that are of high interest to researchers.
(a) Example of table which contains solubility data of compounds. *(EP2949316A1 Table 2)*
(b) Example of table which shows reaction related data. (US09194041B2 Table 2)

Figure 2 Examples of showing the heterogeneity of tables in chemical patent documents.
(a) Example of a self-contained table describing spectroscopic data of compounds. Columns in this table are organized by data format (i.e. images, texts). *(EP1769286B1 Table 1)*
(b) Example of a pharmacological table containing only pointers to contents in the body of the patent documents. Columns in this table are organized by data type (i.e. different activity range). *(EP2049474B1 Table 2)*

Figure 3 Statistics on annotations per semantic category
(a) Statistics on total number of annotations per semantic category.
(b) Statistics on category label overlaps

Figure 4 Statistics on table size per semantic category. Table size is measured as the product of number of rows and columns. The whiskers show a 95% confidence interval.

Figure 5 Statistics on size of tables in the dataset (y-axis on the right side shows percentage of instances with number of rows/columns less than certain range.)
(a) Statistics on number of rows
(b) Statistics on number of columns

Figure 6 Statistics on length of cells within tables in the dataset
(a) Statistics on number of tokens in cells
(b) Average number of tokens in cells per table category. (Estimator shows 95% confidence interval)

Figure 7 Model Architecture of TabNet
(a) Embedder architecture
(b) TabNet architecture

Figure 8 Illustration of different pre-processing approach used in Table-BERT

Figure 9 Effect of input length on classification performance of TabNet and TBResNet
(a) TabNet
(b) TBResNet

Figure 10 Effect of input length on classification performance of Table-BERT
(a) Table-BERT with linearization approach
(b) Table-BERT with naturalization approach

Figure 11 Effect of training set size on (a) overall performance of 3 deep learning models (b) per label performance for Table-Bert
(a) Overall learning curve for 3 deep learning models
(b) Learning curve per label for Table-BERT

Figure 12 Example of `PHARM` table which is predicted as `OTHER` by both Table-BERT and TBResNet. *(US20150259353A1 TABLE 6)*

Tables

Table 1 Table categories IDs within the Reaxys scheme and examples of expected content.

Label	Description	Examples
`SPECT`	Spectroscopic data	*mass spectrometry, IR/NMR spectroscopy*
`PHYS`	Physical data	*melting point, quantum chemical calculations*
`IDE`	Identification of compounds	*chemical names, structure, formula, label*
`RX`	All properties of reactions	*starting materials, products, yields*
`PHARM`	Pharmacological data	*pharmacological usage of chemicals*
`CHEM`	Chemical data	*catalysis, electrochemical reactions*
`COMPOSITION`	Compositions of mixtures	*compositions made up by multiple ingredients*
`PROPERTY`	Properties of chemicals	*the time of resistance of a photoresis*
`OTHER`	Other tables	-

Table 2 Inner annotator agreement between annotator group 1, 2 and gold set in precision ($\mathcal{P}$), recall ($\mathcal{R}$) and F_1 score ($\mathcal{F}_1$). "Random" refers to randomly sampled label from the label distribution in the final gold standard dataset.

Label	Annotator 1			Annotator 2			Random		
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}_1$
SPEC	97.20	97.74	97.47	96.77	96.67	97.47	24.27	26.14	25.17
PHYS	79.02	89.13	83.77	88.43	89.15	88.79	9.43	3.79	5.41
IDE	88.47	94.51	91.39	94.04	85.54	89.59	16.64	18.87	17.68
RX	62.32	82.45	70.98	80.68	83.72	82.17	7.14	4.84	5.77
PHARM	80.94	93.67	86.84	87.55	91.69	89.57	14.76	17.42	15.98
CHEM	23.81	41.67	30.30	83.33	38.46	52.63	0.00	0.00	0.00
COMPOSITION	85.71	85.41	85.56	79.20	79.94	79.57	6.97	8.04	7.47
PROPERTY	45.56	61.89	52.49	36.68	46.35	40.95	3.13	3.45	3.28
OTHER	69.92	25.05	36.88	62.21	58.37	60.23	13.43	15.32	14.31
Overall	81.36	82.38	81.87	83.81	84.00	83.90	16.73	15.41	15.95

Category	NB	SVM	TabNet	ResNet	Table-BERT	Count
SPECT	85.57	90.77	93.66	**95.13**	94.64	1383
PHYS	79.69	79.72	76.39	71.43	**80.29**	371
SPECT—PHYS	78.48	93.58	92.70	**93.77**	93.73	669
IDE	85.93	87.52	93.24	**92.42**	90.75	1311
RX	84.69	85.28	82.35	81.91	**88.77**	490
PHARM	85.76	87.44	88.51	90.97	**91.80**	1430
COMPOSITION	83.10	86.02	88.30	89.05	**91.45**	689
PROPERTY	66.67	69.33	70.97	73.28	**79.14**	351
OTHER	64.71	72.28	75.26	75.30	**76.47**	1184
Micro Avg.	80.95	84.89	86.92	87.30	**88.70**	-

Table 3 Table classification baseline results in F_1 measure. "Count" denoted number of instances in the entire ChemTables dataset.

Metric	Micro F_1			Macro F_1			Weighted Avg. F_1		
# of Rows	1	2	3	1	2	3	1	2	3
NB	**24.51**	**43.17**	51.11	**21.70**	**43.94**	50.24	**23.65**	**43.58**	50.64
SVM	22.03	42.98	**51.94**	20.33	42.04	**50.74**	21.00	42.69	**53.11**
TabNet	**46.29**	**56.44**	**65.21**	**39.64**	**53.43**	**62.22**	**44.56**	**56.33**	**65.15**
ResNet	43.68	53.97	63.87	34.23	48.09	58.50	40.87	51.69	63.19
Table-BERT	38.48	52.51	61.65	33.51	51.49	62.21	36.95	51.99	62.48

Table 4 Micro Avg. F_1 scores of models with only table caption and row headers as inputs. "# of Rows" denotes the number of rows included as input for the models (1 = only the header row used as input).

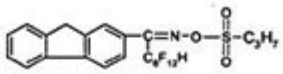
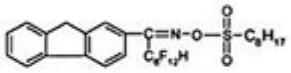
Figures

Menstruum name	Solubility (mg/ml)		
	Ropivacaine free base	Ropivacaine methansulfonate	Ropivacaine hydrochloride
anhydrous ethanol	155	320	95
benzyl alcohol	225	550	140
glyceryl monoacetate	18	50	30
benzyl benzoate	70	2	2.5
ethyl lactate	80	198	16.5
Soybean oil	9.6	2.5	2
Sesame oil	11.7	1.7	2

		Reaction yield (%)	Composition ratio of product (%)			
	Imine conversion (%)	t-butyl (methyl) amine (desired product)	t-butyl (methyl) amine (desired product)	t-butyl (dimethyl) amine (by-product)	t-butyl amine (raw material)	methanol (by-product)
Example L-1	97	84	90.3	1.4	2.7	0.3
Example L-2	97	83	89.4	1.4	2.4	0.1
Example L-3	98	77	91.6	3.0	1.0	1.0
Comparative Example L-1	70	63	87.8	6.2	3.0	0.2
Comparative Example L-2	64	37	70.5	0.6	0.6	0.1

Figure 1

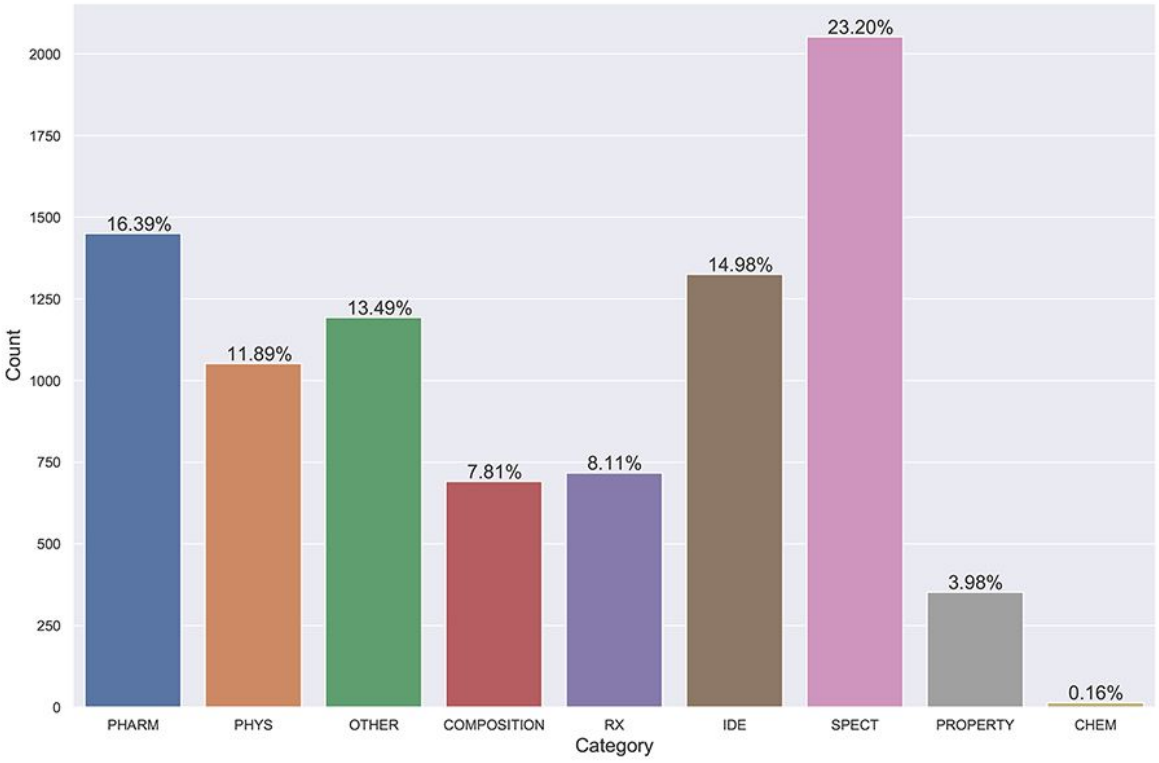
Example of tables in chemical patents that are of high interest to researchers. (a) Example of table which contains solubility data of compounds. (EP2949316A1 Table 2) (b) Example of table which shows reaction related data. (US09194041B2 Table 2)

Ex.	Structure	Purification, Physical properties
3		Recrystallization from 2-propanol ^{1}H-NMR and ^{19}F-NMR ($CDCl_3$) δ [ppm]: 1.10 (t, 3H), 1.87-1.98 (m, 2H), 3.39 (t, 2H), 3.98 (s, 2H), 6.05 (tt, 1 H), 7.33-7.43 (m, 3H), 7.54-7.62 (m, 2H), 7.84 (d, 1 H), 7.88 (d, 1H), -137.40 (d, 2F), - 129.74 (s, 2F), -123.80 (s, 2F), -121.43 (s, 2F), -120.55 (s, 2F), -109.83 (s, 2F), tentatively assigned as E-configuration White solid, mp: 66-68°C
4		Recrystallization from 2-propanol ^{1}H-NMR and ^{19}F-NMR ($CDCl_3$). δ [ppm]: 0.89 (t, 3H), 1.20-1.50 (m, 10H), 1.83-1.96 (m, 2H), 3.40 (t, 2H), 3.98 (s, 2H), 6.05 (tt, 1H), 7.33-7.48 (m, 3H), 7.53-7.63 (m, 2H), 7.88 (d, 1H), 7.88 (d, 1H), - 137.47 (d, 2F), -129.75 (s, 2F), -123.81 (s, 2F), -121.45 (s, 2F), -120.02 (s, 2F), - 109.81 (s, 2F), tentatively assigned as E-configuration White solid, mp: 78-79°C

Example Number	Compound Number	Activity Range (IC50)	Activity Range (EC50)
1	1	D	B
2	2	D	B
3	3	D	C
4	4	D	B

Figure 2

Examples of showing the heterogeneity of tables in chemical patent documents. (a) Example of a self-contained table describing spectroscopic data of compounds. Columns in this table are organized by data format (i.e. images, texts). (EP1769286B1 Table 1) (b) Example of a pharmacological table containing only pointers to contents in the body of the patent documents. Columns in this table are organized by data type (i.e. different activity range). (EP2049474B1 Table 2)



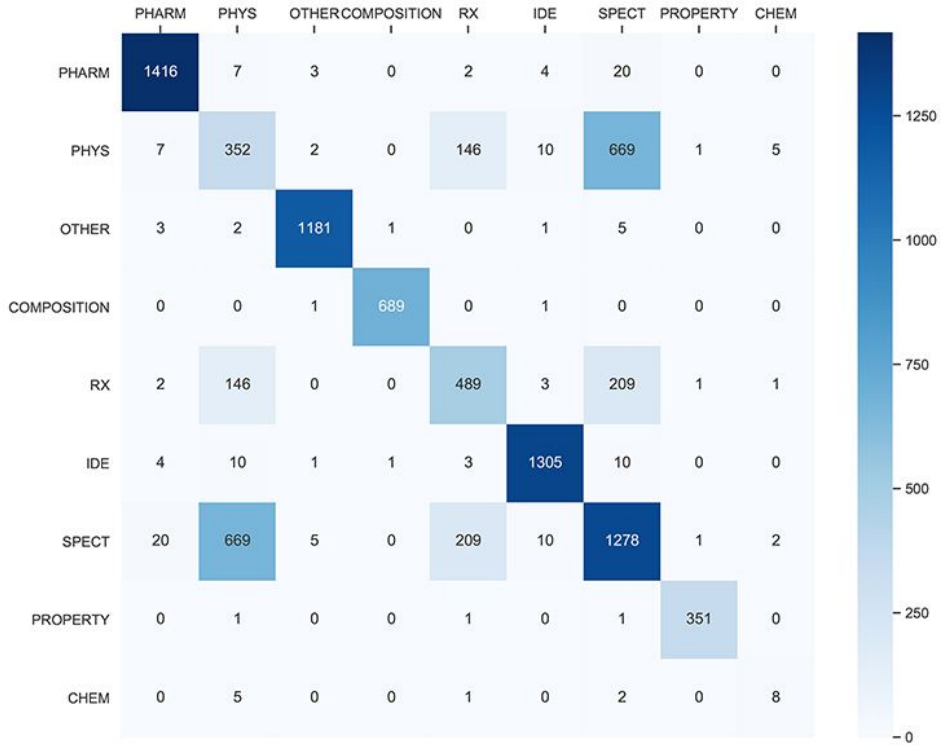


Figure 3

Statistics on annotations per semantic category (a) Statistics on total number of annotations per semantic category. (b) Statistics on category label overlaps

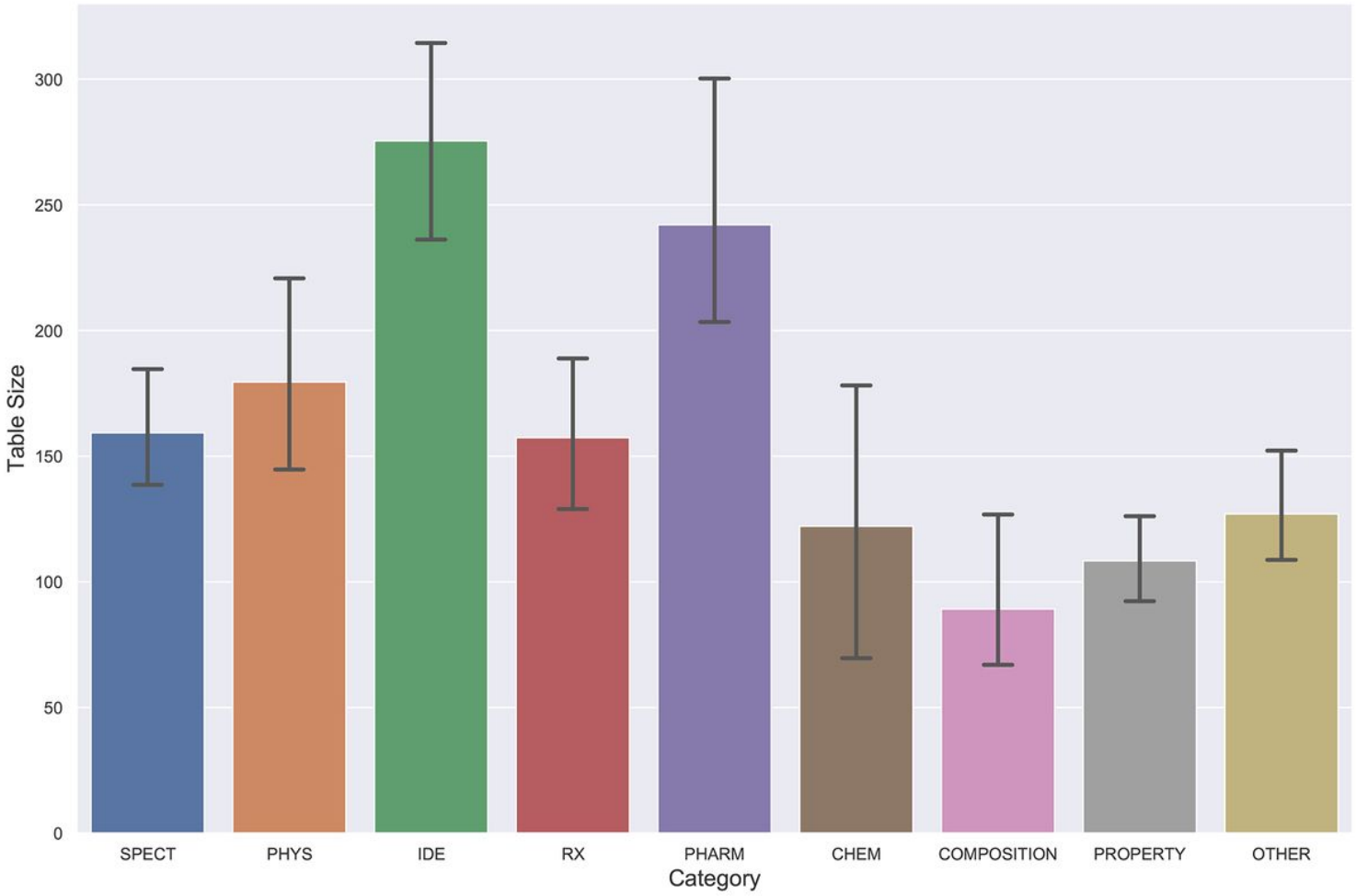


Figure 4

Statistics on table size per semantic category. Table size is measured as the product of number of rows and columns. The whiskers show a 95% condence interval.

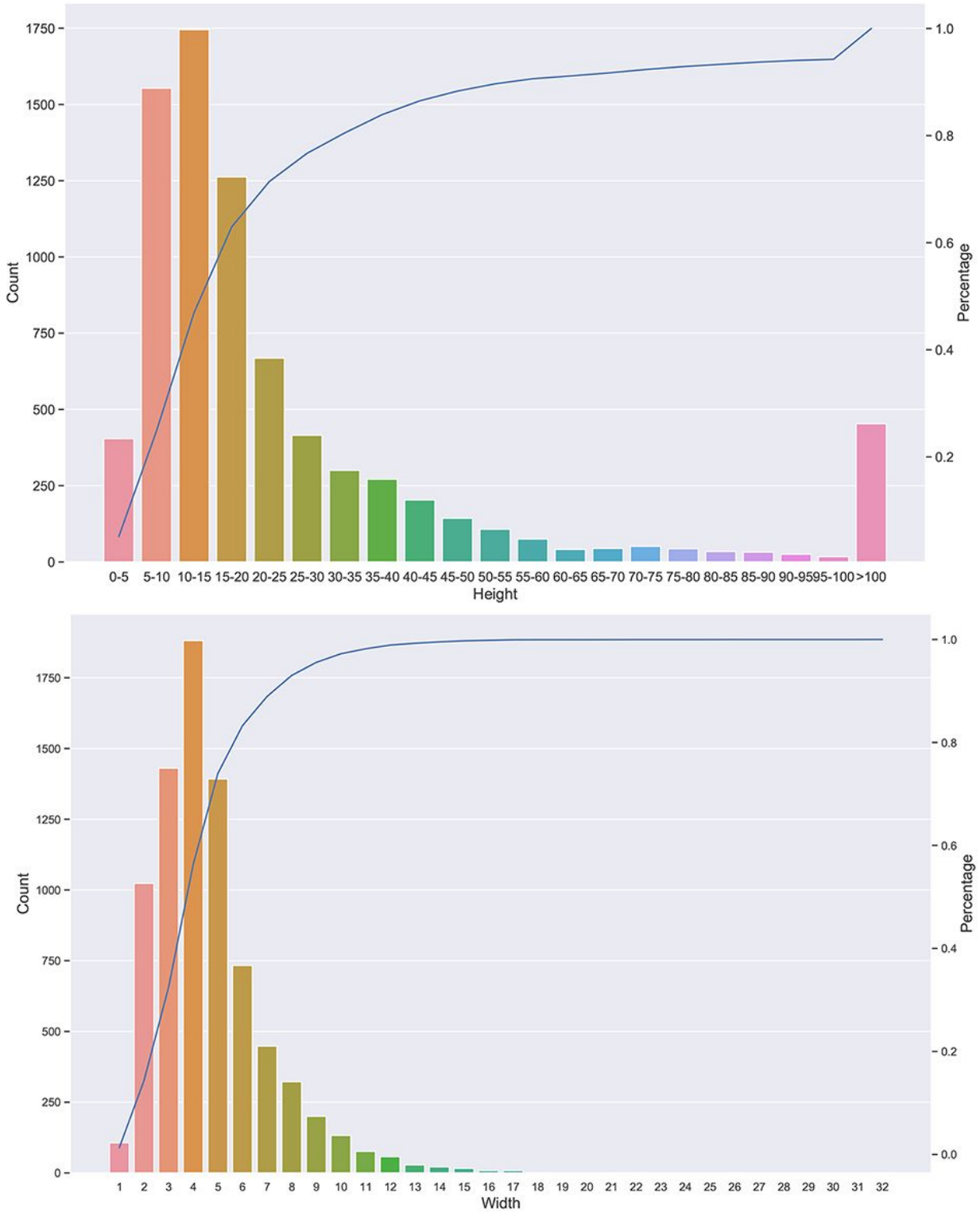


Figure 5

Statistics on size of tables in the dataset (y-axis on the right side shows percentage of instances with number of rows/columns less than certain range.) (a) Statistics on number of rows (b) Statistics on number of columns

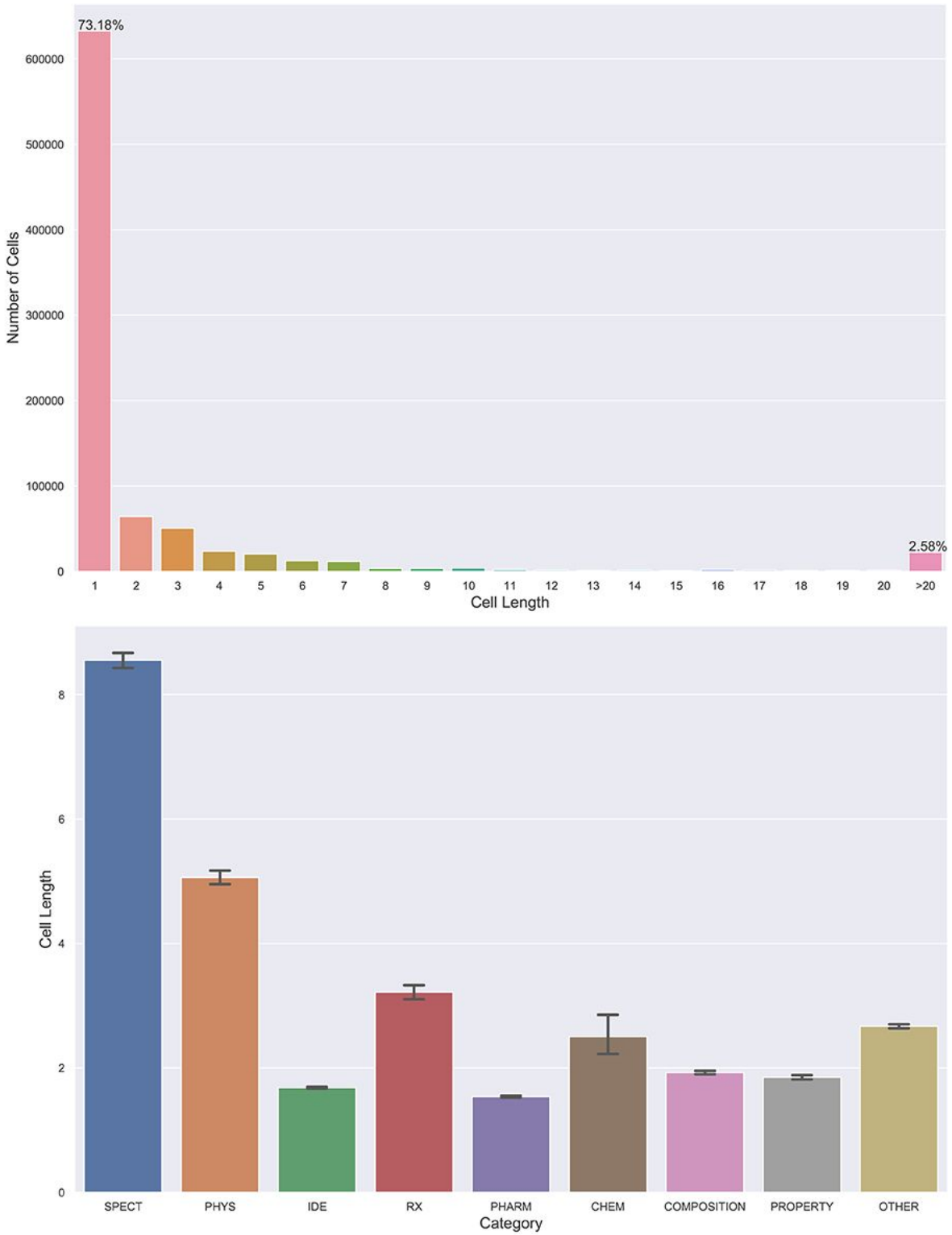


Figure 6

Statistics on length of cells within tables in the dataset (a) Statistics on number of tokens in cells (b) Average number of tokens in cells per table category. (Estimator shows 95% condence interval)

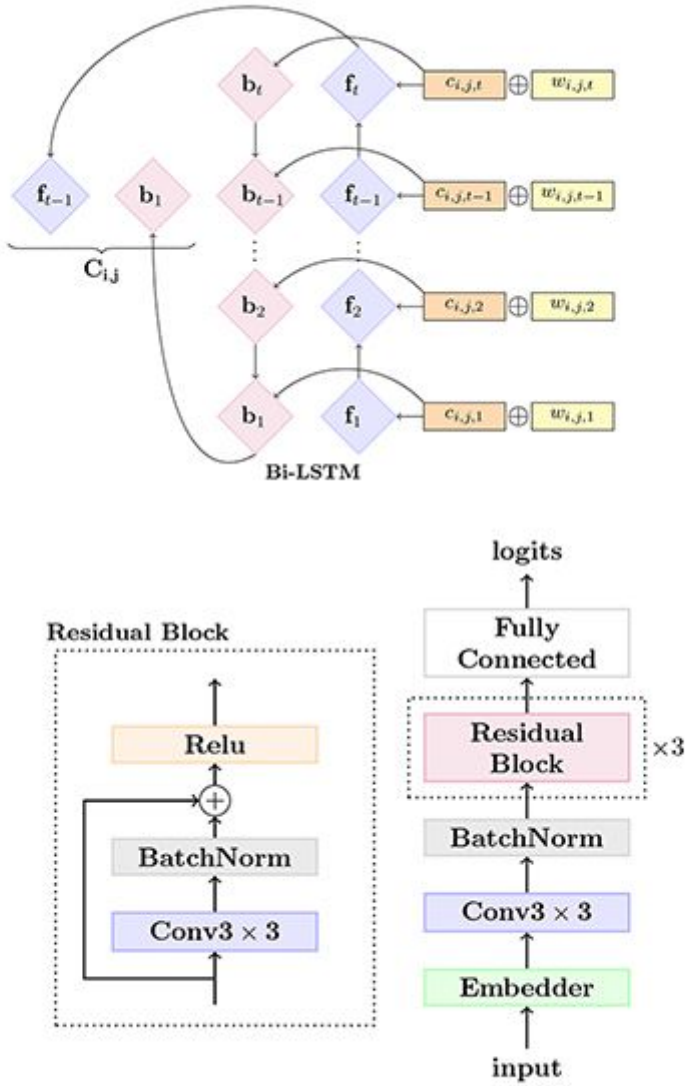


Figure 7

Model Architecture of TabNet (a) Embedder architecture (b) TabNet architecture

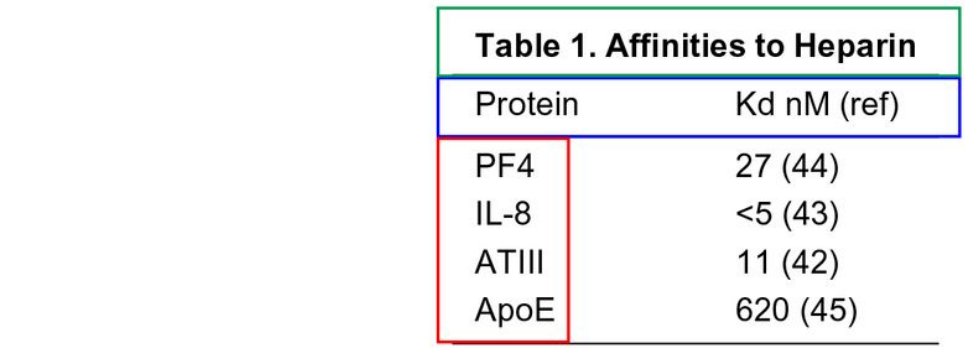


Table 1. Affinities to Heparin

Protein	Kd nM (ref)
PF4	27 (44)
IL-8	<5 (43)
ATIII	11 (42)
ApoE	620 (45)

[**EOS**]: End of sentence

[**SEP**]: End of paragraph

Green: Caption, **Blue: Row headers**, **Red: Column headers**

Linearization

[**CLS**] Table, 1, ., Affinities, to, Heparin, [**EOS**] Protein, Kd, nM, (, ref,) [**EOS**] PF4, 27, (, 44,), [**EOS**] IL, -, 8, <, 5, (, 43,) [**EOS**] ATIII, 11, (, 42,) [**EOS**] ApoE, 620, (, 45,) [**SEP**]

Naturalization

[**CLS**] Table, 1, ., Affinities, to, Heparin, [**EOS**] Protein, PF4, Kd, nM, (, ref,), is, 27, (, 44,), [**EOS**] Protein, IL, -, 8, Kd, nM, (, ref,), is, <, 5, (, 43,) [**EOS**] Protein, ATIII, Kd, nM, (, ref,), is, 11, (, 42,), [**EOS**] Protein, ApoE, Kd, nM, (, ref,), is, 620, (, 45,) [**SEP**]

Figure 8

Illustration of different pre-processing approach used in Table-BERT

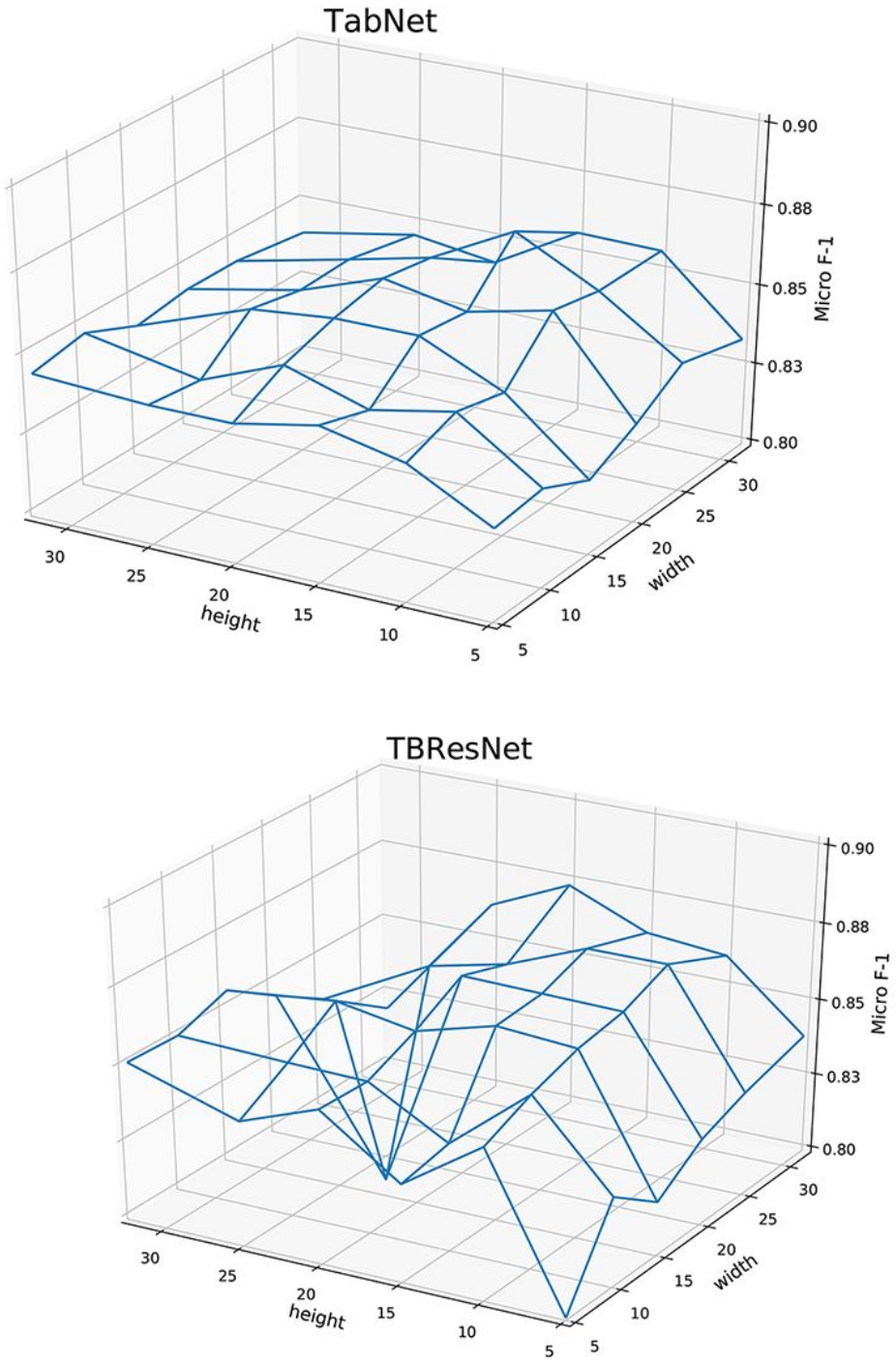


Figure 9

Effect of input length on classification performance of TabNet and TBResNet (a) TabNet (b) TBResNet

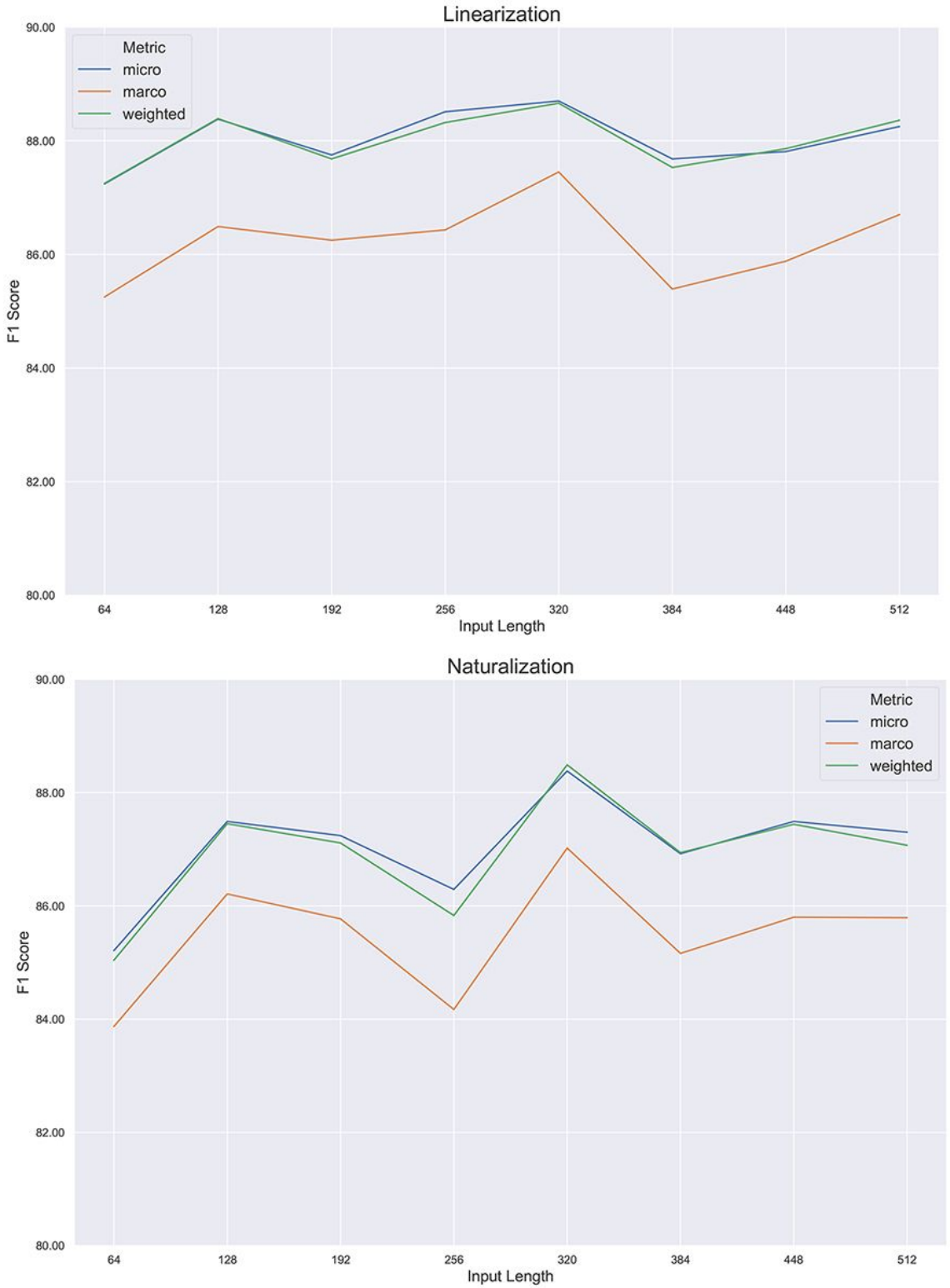


Figure 10

Effect of input length on classification performance of Table-BERT (a) Table-BERT with linearization approach (b) Table-BERT with naturalization approach

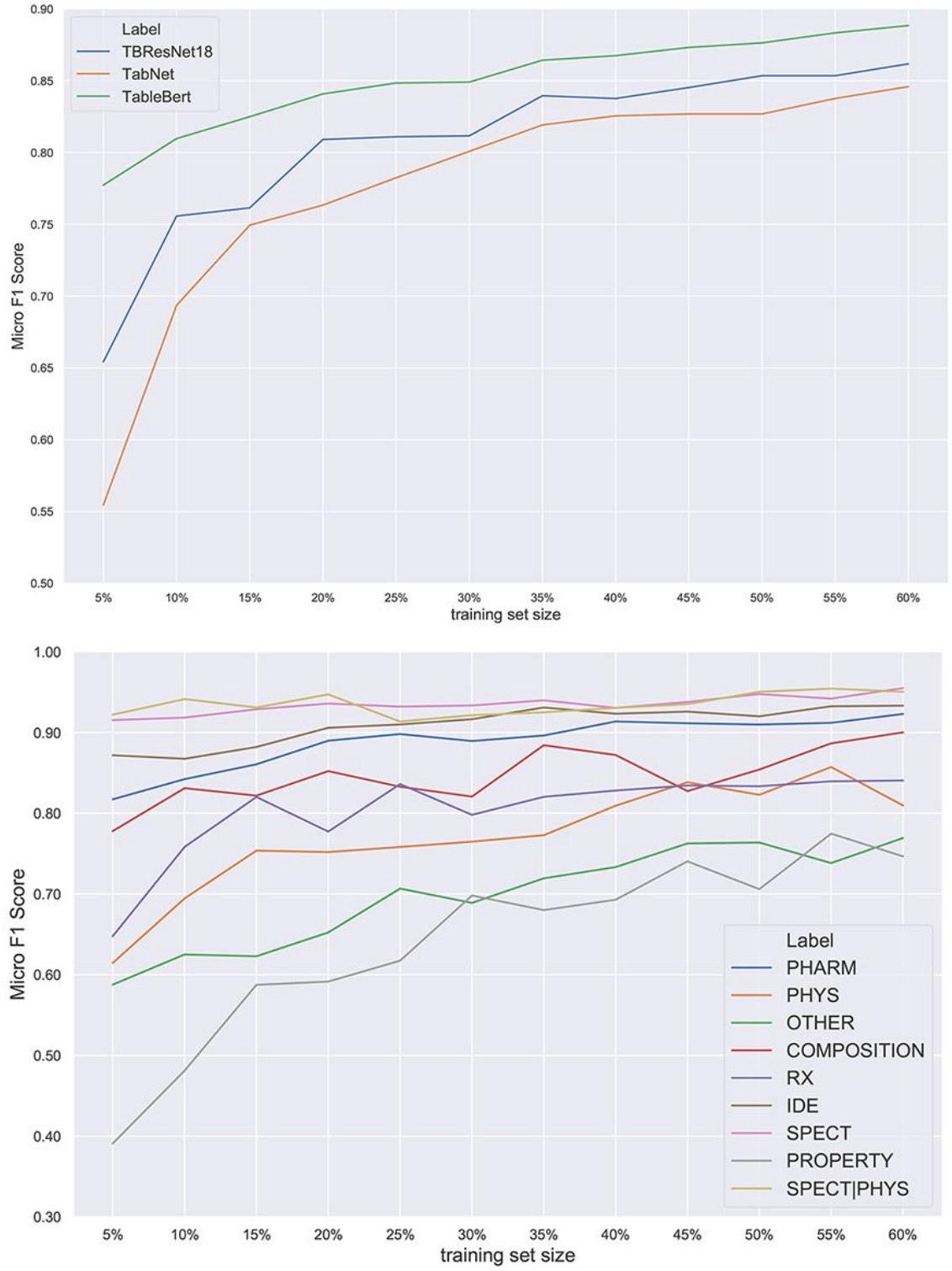


Figure 11

Effect of training set size on (a) overall performance of 3 deep learning models (b) per label performance for Table-Bert (a) Overall learning curve for 3 deep learning models (b) Learning curve per label for Table-BERT

	Route								
	Oral Gavage Phosphate Salt					Oral Capsule Phosphare Salt			
Dog Number	M-1001	M-1002	F-1101	F-1102	Mean	M-2001	M-2002	F-2102	Mean
E Half-life (hr)	7.8	12.0	6.6	5.5	8.7	4.6	7.8	6.2	7.3
Cmax (obs) (ng/mL)	918	1519	1083	1356	1158	2431	1529	1470	1331
Tmax (obs) (hr)	8	4	4	8	4	2	2	8	2
AUC¥ (area) (ng-hr/mL)	16864	31593	16208	21323	21300	13161	21861	20547	18742

Figure 12

Example of PHARM table which is predicted as OTHER by both Table-BERT and TBResNet. (US20150259353A1 TABLE 6)