TableTidier: A UMLS powered data extraction tool

Journal:	*Journal of the American Medical Informatics Association*
Manuscript ID	Draft
Article Type:	Research and Applications
Keywords:	Machine Learning, UMLS, Classifiers

Research and Applications

TableTidier: A UMLS powered data extraction tool

Jesus A. Rodriguez Perez[1], Elaine Butterly, [1] Lili Wei,[1] Avirup Chowdhury,[2] Peter Hanlon[1] and David McAllister[1]

[1]MVLS, The University of Glasgow, UK,
[2]Public Health Suffolk, Suffolk County Council, UK

Corresponding Author:
Jesus A. Rodriguez Perez, PhD,
Institute of Health & Wellbeing,
Room 207, 1 Lilybank Gardens,
University of Glasgow,
G12 8QQ,
jesus.RodriguezPerez@glasgow.ac.uk
07516881104.

Word count: 3958

Keywords: machine learning; UMLS; classifiers

Issue Section: Research and Applications

ABSTRACT

Objective: To assess the performance of UMLS powered classifiers in the context of our semi-automated table data extraction tool, TableTidier.

Materials and Methods: A dataset involving human annotations of 57k+ concepts extracted from 1900 tables was utilised to test whether UMLS-powered classifiers could outperform its non-UMLS counterpart by correctly labelling each of the concepts. Multiple classifiers and feature sets combinations are compared to ascertain the effects of UMLS features against non-UMLS and evaluated in terms of popular metrics such as Precision and Recall.

Results: The results show that the inclusion of UMLS features such as Concept Unique Id (CUIs) and Semantic Types (SemTypes) can have a significantly positive effect on classification. Moreover, the improved behaviour of the classifier is transferred to our Auto-Annotator thus consistently confirming the positive effects of relying on UMLS features.

Discussion: Our evaluation results demonstrate that including UMLS-based features consistently improves the performance of our classifiers, thus optimising the behaviour of TableTidier, which in turn reduces the labour needed by humans when undertaking a systematic review within our tool.

Conclusion: UMLS features can be used effectively in the context of data extraction from tables, as it provides better linkage of text strings to concepts extracted from tables. The better performance achieved can reduce the time and labour spent by users on tasks such as systematic reviews for which TableTidier was firstly designed.

OBJECTIVE

This paper is to evaluate the use of UMLS features in an informed classifier which is a fundamental element of our semi-automatic data extraction tool (TableTidier). The aim of our tool is to reduce the human effort when extracting data from tables contained in published documents, when content is not readily available in a machine-readable format.

BACKGROUND AND SIGNIFICANCE

Systematic reviews are highly influential in clinical decision making,[1]. Following a pre-specified protocol, systematic reviews seek to obtain and extract all available relevant information on a specific clinical question, with the largest and most influential reviews involving many hours of work for highly-trained researchers,[2].

Consequently, a number of software tools,[3] have been developed to assist researchers with the numerous labour-intensive tasks involved in systematic reviewing such as screening published papers for relevance, assessing studies for their risk of bias and extracting data from published results,[4-6]. There are also a number of more generic software tools which may help with some aspects of systematic reviewing. For example, there are extensions available for programming languages commonly used in data analysis that provide functions for processing tabular data (E.g. Unpivotr,[7] and Databaker,[8]).

However, none of these tools are designed to assist in the (semi)-automatic extraction and standardisation of tabular results from published papers. Standardising such tables is not a trivial task; in the medical literature table design is highly idiosyncratic. Even where there are established reporting guidelines,[9] there are no standards for table design, and aesthetic or branding considerations appear to be at least as important as consistency and accessibility. Indeed, features such as multi-level headers are common, and descriptions (labelling) of data-containing cells must often be inferred from ambiguous features such as formatting and the relative position of labels (E.g. Top center table in Figure 1). Consequently, it requires both time and expertise to extract results from tables in the published literature and such data extraction is potentially error-prone.

In traditional systematic reviews of clinical trials, analyses might be based on only a few numbers for each trial. However, advances in the field of clinical trial meta-analysis mean that richer data are now needed,[10] while systematic reviews of epidemiological studies often involve the extraction of large quantities of data covering a range of exposures and outcomes,[11]. Consequently, it has become more important to find ways to improve the efficiency of tabular data extraction.

To address this issue, we developed TableTidier, a software tool which assists the conversion of tabular data to standard formats where each value is unambiguously labelled. Developed using medical journals articles reporting clinical trial findings, TableTidier uses informed classifiers to help estimate the table structure and content, based on the position, formatting and actual text of the table content.

We hypothesised that the use of the Unified Medical Language System (UMLS), which has been shown to improve such diverse tasks as information retrieval, as,[12] and,[13], would improve the precision and recall of this classification task.

Research Problem

Extracting data from tables into a machine-readable format can be extremely useful albeit an extraordinarily challenging task. The main hurdle in automated table data extraction is the preservation of the data cells relationships towards the headings that describe them.

Figure 1. Data Extraction Diagram

Consider the top center table in Figure 1 as an example. We can easily see that the data cell 200 is related to the Gender and Placebo headings. Similarly, 140 is related to the Female and Gender as well as Aspirin and Participants concepts. We can tell because the cells containing numbers are aligned with the headings, but also because of the indented formatting which aligns related concepts such as Gender to Male and Female. Following these cues we can compile the data into a machine-readable form as shown in the bottom right table of Figure 1. While, these visual cues are easily interpreted by our brains, no matter how complex the tables are, this behaviour is not so easily translated into algorithmic solutions. Additionally, we associate terms semantically, independently of their position in the table. Finally, although there are patterns, the data can be organised in any arbitrary way resulting in potentially infinite possible ways to structure them.

However, we believe there are basic elements that can be effectively exploited to enable semi-automated data extraction. These features can be derived from the position of concepts within the table and their textual formatting. Consequently, we believe that by describing the structure of a table in terms of these features, we can use this description as a proxy to extract the data from cells whilst keeping the relationships to the headings. Furthermore, we believe that the inclusion of semantic features capable of linking concepts through associated metadata can improve the fidelity of this semi-automated process. Since UMLS contains the vocabulary utilised in our domain and is already organised as an ontology capable of linking concepts, it is a promising resource for our purposes. Consequently, we aim to focus our investigation in answering the following research question:

Research Question
Can UMLS features enhance the performance and behaviour of the automated annotations provided by TableTidier?

Figure 2. TableTidier: Table to be annotated and editor

MATERIALS AND METHODS

System Architecture

Our TableTidier system is a web application composed by a number of modules organised within a classic Server / User Interface (UI) architecture.

Server written in Node.js combines all sub-modules that provide the data to the User Interface (UI):

- *Automatic Annotation Module*: Predicts the structure of a table from the position and formatting of the different concepts.
- *Data Extraction*: Module written in R which generates the machine-readable data, from the table annotations.
- *Database*: Provided by PostgreSQL, and holding data for results, user interactions, look-up tables, etc.
- *MetaMap Interface*: Communicates with a "*dockerised*"[1] version of MetaMap in order to assign concepts to the strings extracted from cells.

User Interface (UI) built with React.js, comprises the following components:

- *Live Table Viewer/Editor*: Allows the user to view and edit tables on the fly in order to fix OCR or transcription issues. (Figure 2)
- *Table Annotator*: Allows the users to manually and/or automatically annotate the structure of a table, in preparation for data extraction. (Figure 3)
- *Live Data Previewer:* Shows the data extracted on the fly, as users make changes to the table annotations. Helps users to make corrections if needed. (Figure 3)

Annotation and Data Extraction

In this section we describe the steps involved in annotating a table in TableTidier's UI, with the aim of extracting the data and concepts within cells:

[1] https://metamap.nlm.nih.gov/DockerImage.shtml

1. For every table to be annotated (Figure 2) the user will be presented with two options within the annotation section of the UI (Figure 3).

   A. Provide a manual "annotation" which describes the structure of the table in terms of its row/column content and format. Annotations are described by assigning a label to each relevant row and column which relates to its content, and by tagging any relevant formatting options as shown in (Figure 3).

   B. Utilise the "**Automatic Annotation Module**" which attempts to simulate a human annotation and edit the annotation if needed. This study presents approaches to improve the behaviour of this module. ("Auto Add" button in Figure 3).

2. In both cases, the "Live Data Previewer" (Bottom panel in Figure 3) attempts to extract the data given the previous annotation, and display a preview of the machine readable format.

3. Once all tables are annotated, the user can obtain all extracted data in a machine-readable format directly from TableTidier.

Figure 3. TableTidier: Annotation and Data extraction Screenshots[2]

Data sources

Tables Dataset: The 1900 tables in our dataset originate from the "denominator trials" shown in Figure 1 of our recent publication,[14]. Briefly, eligible trials were registered via the US Clinical Trials Register (clinicaltriasl.gov), started on or after 1st January 1990, were phase 3 or 4, recruited at least 300 participants, had an upper age of at least 60 years and evaluated drugs

[2] For a demonstration of the tool visit: https://vimeo.com/365161796

for a selected set of chronic conditions.

UMLS: The UMLS,[15], or Unified Medical Language System, is a set of medical ontologies and interfacing software, which allows the interaction between many biomedical vocabularies, thus enabling the interoperability of computer systems. Its Metathesaurus includes popular vocabularies such as ICD-10, MeSH and SNOMED, linked together by means of a semantic ontology which associates concepts through semantic relationships.

MetaMap: MetaMap,[16] is a software developed by Dr. Alan (Lan) Aronson at the National Library of Medicine (NLM) which allows the processing of text utilising NLP[3] techniques, and its subsequent linkage to the UMLS metathesaurus. We make extensive use of MetaMap in our evaluation to derive our UMLS-based features, in particular CUIs[4] and SemanticTypes,[17].

Features

- **pos_start**: Whether the concept appears in the first row of the table.
- **pos_middle**: Whether the concept appears between the first and last rows of the table.
- **pos_end**: Whether the concept appears in the last row of the table.
- **is_bold**: Whether the concept is formatted as bold
- **is_italic**: Whether the concept is formatted as bold
- **is_indent**: Whether the concept is formatted as indented
- **is_empty_row**: Whether the concept is in the only populated cell of its row

[3] Natural Language Processing
[4] Concept Unique Ids

- **is_empty_row_p**: Whether the concept appears in a row, where the only other populated cell contains a "P value"
- **semanticTypes**: The UMLS semantic groups assigned by MetaMap to the text on each cell. (e.g. "inpo" semanticType code for "Injury or Poisoning")
- **cuis**: The Concept Unique Identifiers (CUIs) assigned by MetaMap to each of the text strings in the table cells. (e.g. C0001779 which represents the concept "Age")

Note: Multiple CUIs and semanticTypes can be associated with a single string or concept.

Dataset

Our starting evaluation dataset comprises 1900 tables and their manual annotations mentioned above. During the annotation process each row and column contained in each table's headings was labelled using one of the following: *arms, characteristic_level, characteristic_name, measures, other, outcomes, p-interaction, time/period*. Furthermore, concepts within columns and rows were grouped depending on their common formatting options: "bold, indentation and italics" and positional cue's: "empty_row" and "empty_row_with_p_value". All these formatting and positional cues refer directly to the features defined above.

We performed a feature extraction over all concepts belonging to either annotated rows or columns, and all features described in the previous Section, resulting on 59175 concepts. Each of these concepts was then processed using the Metamap API to extract the associated CUI and Semantic Types.

Feature Sets

The features introduced in the previous Section were combined to produce four different feature sets, in order to evaluate the effect of UMLS features:

- **Basic**: Set without UMLS features (Features 1-8).
- **UMLS-SemTypes**: Extension of Basic adding the **semanticTypes** from UMLS (Features 1-9)
- **UMLS-CUIs**: Same as Basic but adding the **cuis** feature from UMLS (Features 1-8, 10)
- **UMLS-Full**: Includes all features (Features 1-10)

Classifiers

In order to ensure that results are independent from the classifier of choice, we have chosen a number of classifiers implemented in the "*scikit-learn*[5]" python library. The classifiers belong to mostly different families to ensure significantly different implementations:

- Tree: **DecTree** (sklearn.tree.DecisionTreeClassifier)
- Naive-Bayes: **MultinomialNB** (sklearn.naive_bayes.MultinomialNB)
- Logistic Regression: **LogReg** (sklearn.linear_model.LogisticRegression)
- Ensemble: **RandomForest** (sklearn.ensemble.RandomForestClassifier)
- Ensemble: **AdaBoost** (sklearn.ensemble.AdaBoostClassifier)

All classifiers were run utilising their default parameters and were not optimised in any way. This is unimportant, as we only want to investigate whether UMLS features can improve classification performance.

[5] https://scikit-learn.org/

Evaluation Metrics

In our evaluation we have two tasks. The first task is involved in the classification of single concepts associated to table cells which are part of the table headings, namely **Multi-label Classification task**. The second task, which utilises a trained classifier as part of TableTidier's auto-annotation module, is denominated **Auto-Annotation Task**.

In the **Multi-label Classification task**, the classifiers have been trained with the capability to produce multiple labels for a single prediction. For example, a concept may be identified as both a "*characteristic_name* and *characteristic_level*". Therefore, we define Precision and Recall as follows:

- **Precision**: The number of correctly assigned labels as a proportion of the total number of labels predicted for a concept. (Normalised between 0 and 1)
- **Recall**: The number of correctly assigned labels as a proportion of the total number of labels which should have been assigned to a concept. (Normalised between 0 and 1)

Example of **Precision** and **Recall**:

Concept	**Expected Labels**	**Predicted Labels**	**Precision**	**Recall**
"Gender Female"	"Characteristic_name, Characteristic_level"	"Characteristic_level"	1.0	0.5

In this case the classifier correctly predicted a single label for the concept thus obtained a precision of 1, however it missed the other expected label ("Characteristic_name"), thus recall is 0.5.

On the other hand, during the **Auto-Annotation Task**, the performance is evaluated using a similarity metric between the human annotation, and the prediction produced by the auto-annotation module in TableTidier, namely "**Similarity Score"**. Specifically, we compute the normalised edit distance between the elements of the human annotation and the automatic annotation. Then we combine these distances as an average, and invert it, to produce the final **"Similarity Score"**

Example of **Similarity Score**:

Human annotations		**Automatic annotations**		**Edit distance**[6]
Row 1	Arms; P-Interaction	Row 1	Arms	0.333
Col 1	Characteristic_name	Col 1	Characteristic_level; Arms	0.667

In this case, the Similarity Score is computed averaging the normalised edit distances and taking the inverse of the result, as follows:

$$\textit{Similarity Score} = 1 - (0.667 + 0.333) / 2 = 0.5,$$

whereas a result of 1 would indicate a perfect fit and 0 complete disagreement, between the human annotation and the attempted prediction.

Note: *We firstly compute the edit distance of every human annotation to every automatic annotation, then we pair them by their shortest distance. This is important as otherwise comparisons would be arbitrary.*

[6] Normalised by the total number of possible steps between the compared Annotations.

RESULTS

The experimental results are showcased throughout Table 1, Table 2 and Figure 1. Tables Table 1 and Table 2 hold the evaluation results for the first task **Multi-label Classification**, whereas Figure 1 presents the results for the **Auto-**

Classifier	Feature Set	Precision	Recall
RandomForest	Basic	**0.611**	**0.602**
	UMLS-SemTypes	**0.612**	**0.602**
	UMLS-CUIs	**0.614***	**0.602**
	UMLS-Full	**0.614***	**0.603**
LogReg	Basic	0.595	0.573
	UMLS-SemTypes	0.596*	0.575*
	UMLS-CUIs	0.606*	0.590*
	UMLS-Full	0.607*	0.592*
AdaBoost	Basic	0.590	0.570
	UMLS-SemTypes	0.590	0.570
	UMLS-CUIs	0.593*	0.573*
	UMLS-Full	0.592*	0.572*
MultinomialNB	Basic	0.587	0.541
	UMLS-SemTypes	0.587	0.541
	UMLS-CUIs	**0.614***	0.588*
	UMLS-Full	**0.614***	0.589*
DecTree	Basic	0.584	0.571
	UMLS-SemTypes	0.585	0.572
	UMLS-CUIs	0.593*	0.579*
	UMLS-Full	0.593*	0.579*

Table 1. Summary statistics for all feature sets and classifiers when predicting single labels. (p < 0.01 w.r.t. Basic Feature Set. Bold denotes best performance)*

Annotation Task.

Table 1 results presents Precision and Recall values for the classification of the 59175 concepts involved in training the classifiers already introduced in previous Section. In order to reduce any type of bias all values are obtained by averaging over 5-fold cross-validation runs. Each cross-validation run takes 70% of the dataset to train and 30% for testing. Before the cross-validated runs we also shuffle the whole dataset, so avoid any groupings due to similar concepts appearing close as they are extracted from similar tables. All combinations of Feature set and Classifier are run over the same folds, so that we ensure results are reliably comparable across the table.

				(T-Test)	Differences		Total cases		
Feature Set 1	**Feature Set 2**	**Precision**		**p-value**	**Mean**	**SD**	**Better**	**Worse**	**Diff.**
Basic	UMLS-SemTypes	0.608	0.610	0.342	0.002	0.318	1693	1651	42
Basic	UMLS-CUIs	0.608	0.615	0.007	0.007	0.393	2359	2197	162
Basic	UMLS-Full	0.608	0.617	0.001	0.009	0.399	2450	2220	230
UMLS-SemTypes	UMLS-CUIs	0.610	0.615	0.059	0.005	0.400	2417	2292	125
UMLS-SemTypes	UMLS-Full	0.610	0.617	0.011	0.007	0.398	2417	2257	160
UMLS-CUIs	UMLS-Full	0.615	0.617	0.402	0.002	0.311	1574	1530	44
		Recall							
Basic	UMLS-SemTypes	0.609	0.610	0.910	0.001	0.328	1393	1386	7
Basic	UMLS-CUIs	0.609	0.612	0.430	0.002	0.393	2056	1921	135
Basic	UMLS-Full	0.609	0.613	0.196	0.004	0.398	2095	1956	139
UMLS-SemTypes	UMLS-CUIs	0.610	0.612	0.494	0.002	0.400	2111	1975	136
UMLS-SemTypes	UMLS-Full	0.610	0.613	0.230	0.003	0.398	2103	1966	137
UMLS-CUIs	UMLS-Full	0.612	0.613	0.524	0.001	0.321	1320	1321	-1

Table 2. Exploring 20569 cases where any differences were found for the RandomForest classifier.

Additionally, Table 2 goes into details for the best performing classifier, RandomForest. RandomForest achieves the highest Precision and Recall values for all feature sets, closely followed by MultinomialNB in terms of Precision. In this table, we add P-values for a paired t-test which compares every feature set to each other. Moreover, we introduce the “Differences” column which focuses only in those cases where any difference is observed. In other words, we exclude all cases where regardless of the feature set choice the same performance is achieved. Specifically, we include the mean of subtracting Precision (or Recall) of “Feature Set 1” from “Feature Set 2”. Thus, a positive value shows a better performance in average by “Feature Set 2”. The number counts such cases is also presented in the “Total cases” column. This column presents the number of cases were “Feature Set 2” performed “Better” and “Worse” than “Feature Set 1” along with the difference “Diff”.

*Figure 4. Difference scores comparing Automatic Annotations vs Human annotations. (Statistical significance computed by Paired T-Test and denoted by *, and not significant denoted by "ns")*

Finally, Figure 1 introduces the results obtained from employing the trained RandomForest classifiers as part of the TableTidier's Auto-Annotation module, during the **Auto-Annotation Task.** The classifiers were similarly trained to the previous case, however this time we removed 30% of the Tables (not the concepts) from the dataset, leaving 70% of the tables to extract the concepts from and perform the 5-fold cross validation learning. The removed 30% of tables (511) were then used for testing the performance of the classifiers under the **Auto-Annotation Task**, showcased in this figure. As we can observe, we compare each feature set combination by means of a paired T-test, indicating statistical significance with the "*" symbol, and non-significance with "ns".

DISCUSSION

Multi-Label Classification Task

Table 1 presents the evaluation results for all chosen classifiers and all feature set combinations. Firstly, we compare the performance of the Basic feature set against UMLS-SemTypes. The inclusion of the "*semanticTypes*" feature seems to have little effect. Significantly better performance in terms of Precision and Recall is only obtained for LogReg. Clearly the inclusion of this feature does not result in any definitive advantage, as the rest of classifiers do not show any significant changes over the Basic baseline feature set. On the other hand, the inclusion of "cuis" in the *UMLS-CUIs* feature set has a much more positive impact. Statistically significant performance increases are consistently achieved for all classifiers in terms of Precision and Recall, except for the Recall obtained for RandomForest. Finally, the *UMLS-Full* feature set seems to produce a behaviour that very closely resembles that of the *UMLS-CUIs* set, as results are very similar. This connects with the very little effect that the "*semanticTypes*" feature seems to have over the performance. Finally, the best performance is achieved by the RandomForest classifier equipped with the UMLS-Full feature set, with MultinomialNB being very tight contender, albeit with lower Recall.

Moreover in Table 2 we investigate the best performing classifier, RandomForest, in more detail. In order to focus on the effects, the results included in this table are only those where any differences were observed when considering all metrics and all feature sets. The resulting subset is composed of 20569 cases out of 59175. The table displays the results for each feature combination indicated by "Feature Set 1" and "Feature Set 2". Generally, a positive trend can be observed when UMLS features are included. Statistically significant evidence of this improvement is found for UMLS-CUIs and UMLS-Full when compared to Basic. However, whilst UMLS-SemTypes is slightly better in average, it does not improve significantly. Interestingly UMLS-Full, which includes *semanticTypes* and *cuis* features, seems to obtain a better p-value than UMLS-CUIs alone. This suggests that the combination of *semanticTypes* and *cuis,* does improve the statistical evidence of the improved results obtained when UMLS features are included. Furthermore, the column named *"differences"* introduces the mean and standard deviation of the difference produced when subtracting the performance of "Feature Set 1" from "Feature Set 2". For example, the first row of results shows the mean of subtracting the Precision values of Basic from UMLS-SemTypes at each data point. As we can see, whilst positive, the average difference is very small, which connects with how cases are affected by the difference feature set configurations in the last column. The "Total cases" column is sub-divided into three columns where *Better* shows the total number of cases where "Feature Set 2" improved over "Feature Set 1", *Worse* shows the opposite case, and *Diff* is the subtraction

"*Better - Worse*". As we can observe there is a consistent positive trend, from using UMLS features, albeit with limited impact, as it also negatively affects many cases. In the following Section we examine what is the effect of UMLS features when utilised as part of the Auto-Annotator module.

Auto Annotation Task

In previous results we focused on the performance of the classifier when predicting labels for a single concept. In this section, we measure the classifiers' performance as part of the Auto-Annotation module within our TableTidier platform. To this end, Figure 4 shows a comparison of all feature sets when applied to our automatic annotation, measured by the above-mentioned "Difference Score". Our Auto-Annotation module takes all predictions and aggregates them into rows and columns utilising a voting mechanism to choose the most commonly occurring labels for each row or column. However, since we are only interested in exploring the effects of the different feature sets, we treat it as a black box, and focus on whether the effects of UMLS can still be significantly perceived in the end result.

Similarly, to the results obtained for the previous task, we can observe statistically significantly improved performance for all UMLS configurations with respect to the Basic, as denoted by the stars on top of the lines connecting the results for each feature set. However, the change in performance within the different UMLS versions is not that clear, as we find no statistically significant effects. There is a slight reduction on the variability when comparing UMLS-Full to UMLS-SemTypes.

Regardless of the choice of UMLS-abled version, this experiment serves as another piece of evidence to demonstrate the positive effect of utilising UMLS features, since the positive effects obtained by the classifiers ripple through the auto annotation module process of our system. Hence, we can conclude that utilising UMLS features significantly improves performance of our classifier, which in turn significantly improves the automatic annotation module of TableTidier, thus confirming our initial hypothesis. These results translate to significantly less manual labour involved when annotating tables, and less time required for data extraction in tasks such as systematic reviews.

CONCLUSION

Systematic reviews are highly influential in clinical decision making, and increasingly require the extraction of large quantities of data from tables published in scholarly journals, which is a challenging and labour-intensive task. We found that adding UMLS features to software designed to aid this task markedly improved precision and recall.

We built a system that exploits the structure of tables in order to convert them into a machine-readable format, in a semi-automated manner, by means of a classifier. The basic software relied upon the position of concepts within the tables, as well as the existence of formatting cues. However, we also found that adding UMLS-based features into the classifier - CUI codes and SemanticTypes assigned by MetaMap which represent features with different levels of specificity - markedly improved prediction. Most importantly, we showed how this finding is robust to the choice of classifier (e.g. random forest, neural network), dataset and feature sets, as the rich contextual knowledge embedded in UMLS and MetaMap helps to significantly increase performance of our automatic annotator.

MetaMap is a complex software with many options and functions. We elected to use the out-of-the-box defaults, and alternative options may have led to better or worse performance. However, our decision to do so does mean that over-fitting is a highly unlikely explanation for the improved performance we observed, since we made a very small number of modelling choices (i.e. the basic model versus two different sources of information from MetaMap using default settings.)

A strength of this study includes the fact that the allocation of labels by the manual reviewers was done blinded to (indeed prior to) the mapping of strings to UMLS concepts. However, there are number of weaknesses in our study. First, we only examined tables for one subject area, clinical trials, and we do not know if the use of UMLS will increase performance similarly in closely related fields such as pharmaco-epidemiology, or in moderately-related fields such as genetic, biomarker, clinical and social epidemiology. Nonetheless, the influential role of clinical trials within evidence-based medicine mean that our findings are of some importance despite our selection of a single field.

Finally, it may be concluded that mapping strings from published tables of clinical trial reports to UMLS using MetaMap resulted in significant improvements in our ability to automatically classify labels. This was consistent across the choice of classifier (e.g. random forest, neural network) and when used as part of the Auto-Annotation module in TableTidier. Thus, the integration of UMLS and MetaMap into our (and perhaps other's) tools have the potential to significantly reduce the manual work involved in conducting systematic reviews.

Acknowledgements

We would like to thank Hebe Shedden and Yifan Shen for their help in compiling a substantial part of the training data used in this study, as well as the Wellcome Trust for funding the development of this project.

REFERENCES

[1] M. H. Murad, N. Asi, M. Alsawas and F. Alahdab, “New evidence pyramid,” *BMJ Evidence-Based Medicine,* vol. 21, p. 125–127, 2016.

[2] J. P. T. Higgins and S. Green, Cochrane handbook for systematic reviews of interventions, vol. 4, John Wiley & Sons, 2011.

[3] C. Marshall and A. Sutton, “Systematic Review Tools,” 2020 (accessed February 9, 2020).

[4] S. R. Jonnalagadda, P. Goyal and M. D. Huffman, “Automating data extraction in systematic reviews: a systematic review,” *Systematic reviews,* vol. 4, p. 78, 2015.

[5] L. A. C. Millard, P. A. Flach and J. P. T. Higgins, “Machine learning to assist risk-of-bias assessments in systematic reviews,” *International Journal of Epidemiology,* vol. 45, pp. 266-277, 12 2015.

[6] G. Tsafnat, P. Glasziou, M. K. Choong, A. Dunn, F. Galgani and E. Coiera, “Systematic review automation technologies,” *Systematic Reviews,* vol. 3, p. 74, 09 7 2014.

[7] D. Garmonsway, “Unpivotr: Unpivot Complex and Irregular Data Layouts,” 2019 (accessed February 9, 2020).

[8] T. S. C. Company, “Databaker,” 2018 (accessed February 9, 2020).

[9] Z.-x. Bian and H.-c. Shang, “CONSORT 2010 statement: updated guidelines for reporting parallel group randomized trials,” *Annals of internal medicine,* vol. 154, p. 290–291, 2011.

[10] D. M. Phillippo, S. Dias, A. Elsada, A. E. Ades and N. J. Welton, “Population Adjustment Methods for Indirect Comparisons: A Review of National Institute for Health and Care Excellence Technology Appraisals,” *International Journal of Technology Assessment in Health Care,* vol. 35, p. 221–228, 2019.

[11] A. S. V. Shah, K. K. Lee, D. A. McAllister, A. Hunter, H. Nair, W. Whiteley, J. P. Langrish, D. E. Newby and N. L. Mills, “Short term exposure to air pollution and stroke: systematic review and meta-analysis,” *bmj,* vol. 350, p. h1295, 2015.

[12] Y. Zhang, P. K. Srimani and J. Z. Wang, “Combining MeSH Thesaurus with UMLS in pseudo relevance feedback to improve biomedical information retrieval,” in *2016 IEEE International Conference on Knowledge Engineering and Applications (ICKEA)*, 2016.

[13] H. Gurulingappa, L. Toldo, C. Schepers, A. Bauer and G. Megaro, “Semi-Supervised Information Retrieval System for Clinical Decision Support.,” in *TREC*, 2016.

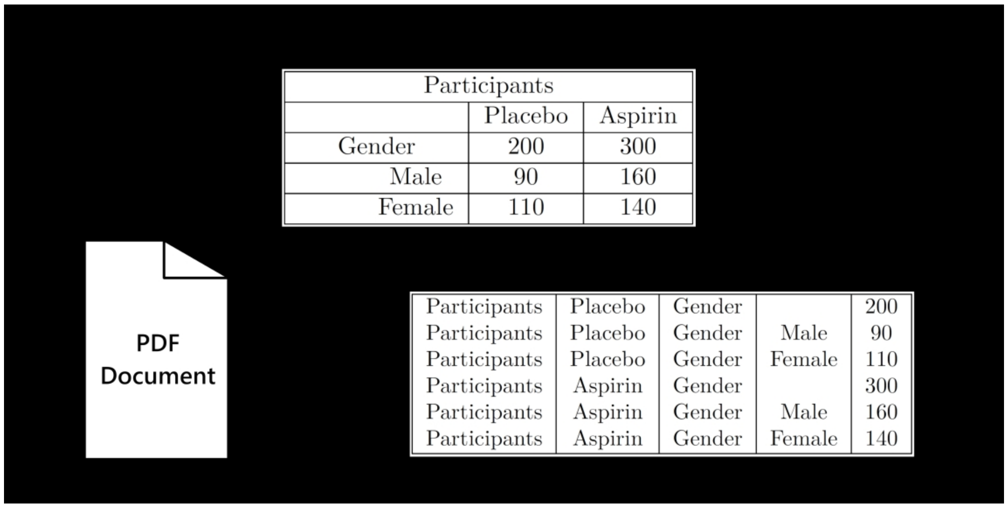
[14] P. Hanlon, L. Hannigan, J. Rodriguez-Perez, C. Fischbacher, N. J. Welton, S. Dias, F. S. Mair, B. Guthrie, S. Wild and D. A. McAllister, “Representation of people with comorbidity and multimorbidity in clinical trials of novel drug therapies: an individual-

level participant data analysis," *BMC Medicine,* vol. 17, p. 201, 12 11 2019.

[15] O. Bodenreider, "The Unified Medical Language System (UMLS): integrating biomedical terminology," *Nucleic Acids Research,* vol. 32, pp. D267-D270, 1 2004.

[16] A. R. Aronson and F.-M. Lang, "An overview of MetaMap: historical perspective and recent advances," *Journal of the American Medical Informatics Association,* vol. 17, pp. 229-236, 5 2010.

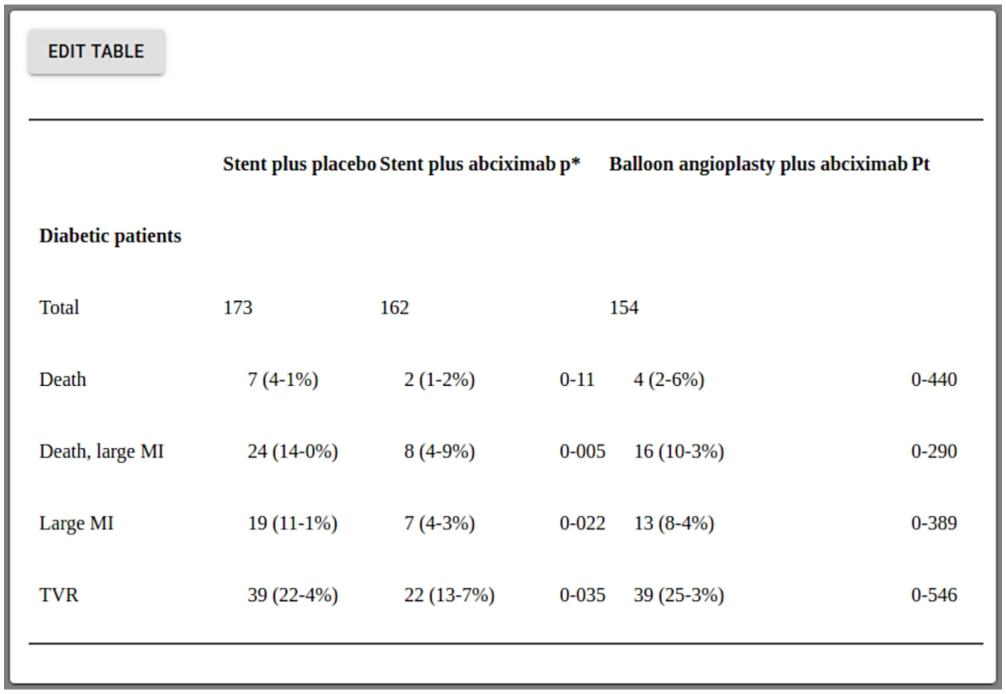
[17] NLM, "NLM: Semantic Types," 2016 (accessed February 9, 2020).



	Participants	
	Placebo	Aspirin
Gender	200	300
Male	90	160
Female	110	140

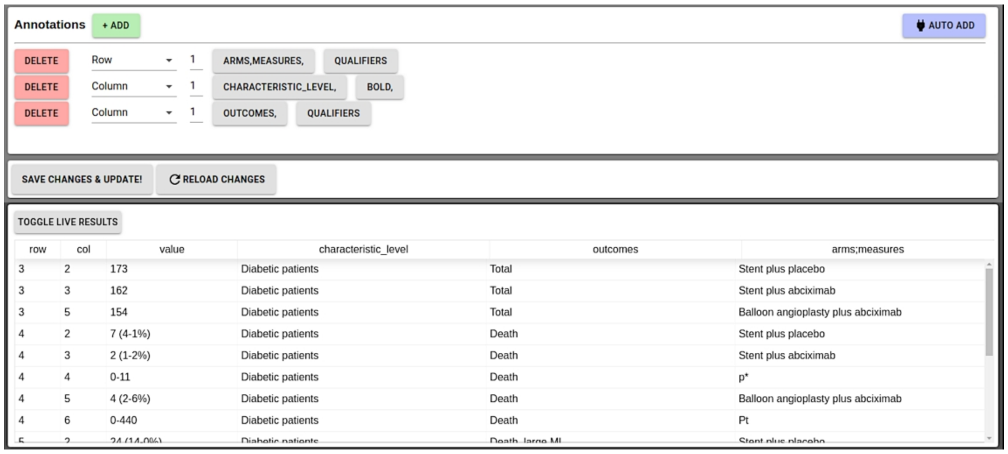
Participants	Placebo	Gender		200
Participants	Placebo	Gender	Male	90
Participants	Placebo	Gender	Female	110
Participants	Aspirin	Gender		300
Participants	Aspirin	Gender	Male	160
Participants	Aspirin	Gender	Female	140

Figure 1. Data Extraction Diagram



	Stent plus placebo	Stent plus abciximab	p*	Balloon angioplasty plus abciximab	Pt
Diabetic patients					
Total	173	162		154	
Death	7 (4-1%)	2 (1-2%)	0-11	4 (2-6%)	0-440
Death, large MI	24 (14-0%)	8 (4-9%)	0-005	16 (10-3%)	0-290
Large MI	19 (11-1%)	7 (4-3%)	0-022	13 (8-4%)	0-389
TVR	39 (22-4%)	22 (13-7%)	0-035	39 (25-3%)	0-546

Figure 2. TableTidier: Table to be annotated and editor



Row	1	ARMS,MEASURES,	QUALIFIERS
Column	1	CHARACTERISTIC_LEVEL,	BOLD,
Column	1	OUTCOMES,	QUALIFIERS

row	col	value	characteristic_level	outcomes	arms;measures
3	2	173	Diabetic patients	Total	Stent plus placebo
3	3	162	Diabetic patients	Total	Stent plus abciximab
3	5	154	Diabetic patients	Total	Balloon angioplasty plus abciximab
4	2	7 (4-1%)	Diabetic patients	Death	Stent plus placebo
4	3	2 (1-2%)	Diabetic patients	Death	Stent plus abciximab
4	4	0-11	Diabetic patients	Death	p*
4	5	4 (2-6%)	Diabetic patients	Death	Balloon angioplasty plus abciximab
4	6	0-440	Diabetic patients	Death	Pt
5	2	24 (14.0%)	Diabetic patients	Death, large MI	Stent plus placebo

Figure 3. TableTidier: Annotation and Data extraction Screenshots

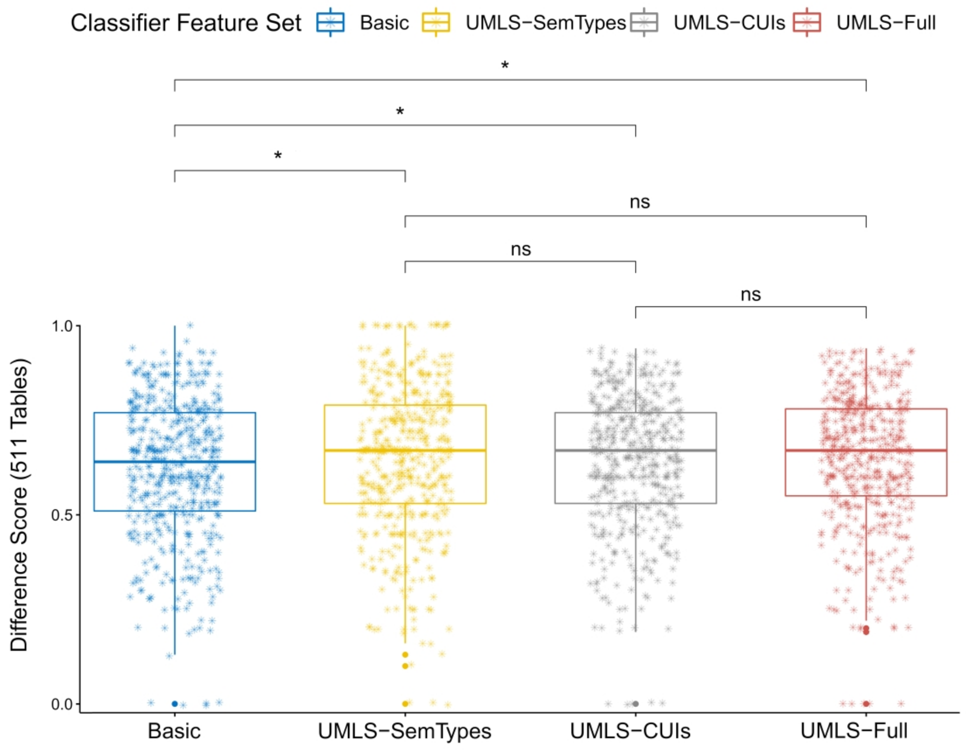


Figure 4. Difference scores comparing Automatic Annotations vs Human annotations. (Statistical significance computed by Paired T-Test and denoted by *, and not significant denoted by "ns")