2022 EAGE ANNUAL
83RD CONFERENCE & EXHIBITION
MADRID I SPAIN

MACHINE LEARNING FOR TABLE CELL CLASSIFICATION

C.H. Lun[1], T. Hewitt[1], S. Hou[1]

[1] CGG

Summary

Tables are ubiquitous in the geoscience industry, appearing in numerous documents and spreadsheets. They contain a wealth of data in a structured format which can help us understand the subsurface. However, the number of tables created over the years is huge and it requires an enormous manual effort for domain experts to read each table to understand what kind of data is in it. Therefore, it would be more efficient to develop an automated way to do this, but different tables can vary greatly in style and layouts which makes it difficult for a machine to understand tables. For this reason, a first step towards automatic extraction of data from tables and spreadsheets is the identification of the role each cell plays, a task called table cell classification. In this work, we explore machine learning techniques for performing this task.

Machine Learning for Table Cell Classification

Introduction

To gain a thorough understanding and create a holistic model of the subsurface, thereby reducing exploration and development risks, it is necessary to incorporate all sources of data. Each year, the geoscience industry creates huge volumes of documents containing data, research, and plans, which describe valuable subsurface assets. These documents contain a wealth of knowledge in the form of text, figures, and tables, which are intended to be read by humans. Of particular importance are the tables and spreadsheets that contain data in an already structured format. In the geoscience industry, a considerable amount of knowledge is stored in this way. Domain experts use tables to record their experiments, tests, and measurement results, as well as their research findings, and thus, they are a valuable source of information.

However, the number of tables created over the years is huge and requires significant manual effort for domain experts to read each table to understand the included data. Therefore, it would be more efficient to develop an automated way to discover what information is contained in these tables. Achieving this not only requires extracting the content of each individual cell, but also identify the relationship between cells. It is common for humans to be able to interpret a table with ease, yet it is difficult or even impossible to explain the steps behind this interpretation, hence it is difficult to develop an algorithm to process tables automatically. Further adding to the difficulty, is that tables can vary in style and layout, and the same data can be presented in different ways depending on the author's preferences.

Spreadsheets, from for example a Microsoft Excel file, are also ubiquitous in the geoscience industry. These add an extra dimension of difficulty since a single spreadsheet may contain more than one table arranged arbitrarily. In addition, spreadsheets often contain cells that are not a part of any table, for example, titles, notes, and metadata, which may provide context to the table. These individual components of a spreadsheet may vary in shape and layout. For these reasons, a first step towards automatic extraction of data from tables and spreadsheets is the identification of the role each cell plays, a task called *table cell classification*. Knowing that a cell is a table header gives meaning to the data cells in the same column or row and identifying the header of a table allows us to later merge the table's data with existing data sources.

In this work, we discuss how we perform the task of table cell classification using machine learning, introduce the dataset used to train and validate our model, and finally describe the two different machine learning approaches used.

Table Cell Classification

This task is concerned with classifying the cells of a table or spreadsheet into one of a set of predefined labels. We experiment with the DECO dataset (Koci et al., 2019a), which consists of 854 spreadsheets annotated at the cell level. The spreadsheets were extracted from the email archive of a certain corporation and each file is associated with an employee, but it is unknown whether the employee is the author of the file or not. The dataset categorises cells into one of: *Data*, *Header*, *Derived*, *GroupHeader*, *Title*, *Note* and *Other*. "Data" cells contain the actual value of the information to be recorded, "Header" cells represent the label of a column and they can span multiple rows and be hierarchical, "Derived" cells are aggregations of other Data cells, "GroupHeader" cells play the same role as Header cells but for a row. "Title" and "Note" cells provide context to the spreadsheets. Titles are often shorter than notes, which contain more information in one or more full sentences. Finally, "Other" is a label for any other cell types that do not fall into one of the above categories.

Although the dataset described is sourced from spreadsheets, these categories can also apply to tables in a page of a document once we have identified the cells of the table. The cells of a document table may be defined by the table grid lines, or by blank spaces between the contents of the table, or a mixture of both. These cell boundaries can be identified by deterministic methods such as that described by

Nurminen (2013). After identifying the cells of a document table, we can treat them like the cells of a spreadsheet. In our classification approaches, we take care to not use information that are specific to any one data source to ensure that our methods are generally applicable.

We explore two approaches: one involving feature engineering and using traditional machine learning techniques, and a natural language processing approach. We split the dataset into a training set and a validation set randomly but such that files from the same employee are either in the training set or the validation but not both.

Feature Engineering

In this approach, each cell is associated with a set of features that are fed to a model to classify the cell into one of the cell types. Features are computed from the contents of the spreadsheets and can be categorised into four types. First are features that are derived from just the contents of the cell itself, such as the number of characters in the cell, the percentage of the characters that are letters, digits, punctuations, or whitespace, and whether the cell is purely numeric as well as whether the cell is empty. The second type of features are the spatial information about the cell, i.e., its row and column index. Finally, the third and fourth type of features are averages of the above features over cells in the same row and column, respectively. The full list of features is presented in Table 1. A similar approach has been investigated by Koci et al. (2019b), but in their work, they used a non-publicly available dataset and did not use the row and column averages as features. Furthermore, some of their features are specific to Excel spreadsheets, such as whether the cell contains a formula. We decided not to use Excel-specific features since we want to keep our method general enough to also work for tables on a document page.

We tested with a random forest classifier and XGBoost and found that the best result was obtained with random forest.

Cell Features	Row Features	Column Features	Spatial Features
string_length	row_string_length_mean	row_string_length_mean	row_idx
alpha_pct	row_alpha_pct_mean	row_alpha_pct_mean	col_idx
numeric_pct	row_numeric_pct_mean	row_numeric_pct_mean	
space_pct	row_space_pct_mean	row_space_pct_mean	
special_pct	row_special_pct_mean	row_special_pct_mean	
is_numeric	row_is_numeric_mean	row_is_numeric_mean	

Table 1 *The features used in our models separated by type.*

Natural Language Processing

Natural language processing (NLP) is a set of methods that enable machines to process human-written languages. Examples of tasks include text generation, text summarisation, text classification, and token classification, the latter of which is the focus of this work. What constitutes a token in a piece depends on the tokenisation method used, for example, tokens could be words and punctuations if one simply splits on whitespaces and symbols, or they could be parts of a word if using tokenisation methods like WordPiece (Schuster and Nakajima, 2012) or SentencePiece (Kudo and Richardson, 2018). Token classification is a task concerned with classifying the tokens within an input text into one of the predefined labels. An application of this is named entity recognition, where the aim is to identify the spans of text which constitute a named entity, such as a person's name or a location.

In this work, we treat table cell classification as a token classification task. More precisely, the contents in each of the cells are tokenised and the tokens in every cell are fed to a deep neural network where the output is a classification of each token. The final classification of the cell is decided by a majority vote of its constituent tokens.

Enron Entity Contact List for RFI's	nan	nan	nan	nan
(As of 4/29/02)	nan	nan	nan	nan
nan	nan	nan	nan	nan
nan	nan	nan	nan	nan
nan	nan	nan	nan	nan
Enron Entity	Commodity	Information Type	Primary Contact	cc:
EPMI	Physical Power	AR/AP (Settlements)	Rebecca Grace	Juan Camarillo/Zakiyyah McClure
nan	Physical Power	MtM Valuation	Laurel Bolt/Ashley Fay	Sam Leuschen
nan	nan	nan	nan	nan
nan	nan	nan	nan	nan
ENA	Physical Gas	AR/AP (Settlements)	Margaret Dhont	A.K. Matheson
nan	Physical Gas	MtM Valuation	Kulvinder Fowler/Patrick Ryder	Greg Couch/A.K. Matheson
nan	nan	nan	nan	nan

Figure 1 *Example predictions of the table cell classification task. Green – Data, Red – Header, Teal – GroupHead, Orange – Title, Blue – Other.*

State-of-the-art NLP models are now dominated by deep neural networks, particularly transformer (Vaswani et al., 2017)-based language models, such as BERT (Devlin et al., 2019) and GPT-3 (Brown et al., 2020), which contain billions of parameters. In our experiments, we used a language model called LayoutLM (Xu et al., 2020). The reason for this is because a table is an object with a 2D structure and we lose this structure if we treat it as merely a sequence of tokens; however, LayoutLM not only takes tokens as input, but it also uses the bounding box (the top-left and bottom-right coordinates of the box) of each token and so we are able to preserve the 2D nature of the tables. Each token in the spreadsheet does not naturally have a bounding box, so we take the row and column index of the cell as the top-left coordinate and add 1 to the indices for the bottom-right coordinate. These are then normalised so that

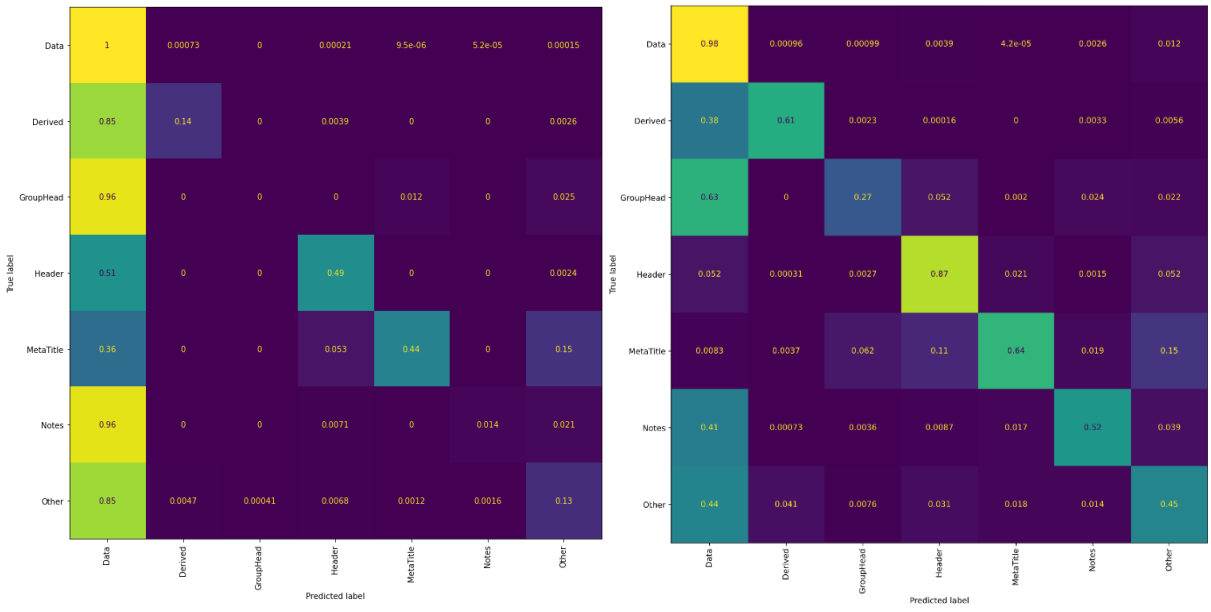

Figure 2 *Confusion matrices generated from predictions on the validation set. The left matrix is for the random forest model and the right matrix is for the LayoutLM model.*

they lie in the range [0, 1000], as required by LayoutLM. In the case of multiple tokens per cell, the bounding box is equally divided horizontally.

Figure 1 shows an example of the prediction by the language model. The confusion matrix for the two methods is shown in Figure 2. As one can see, the NLP approach performs much better than the random forest approach. The best performing classes are Data and Header, while GroupHead is the worst performing. A possible reason for this is because GroupHead is the class with the least training examples.

Conclusions

In this work, we explored two approaches to table cell classification: one approach used features derived from just the content of the cells, while in the other approach, we reframed the problem as a token classification problem in natural language processing where we treat each table as a sequence of tokens, each encoded with positional information in 2D. While the NLP approach works well in identifying the Data and Header cells, which are the two most important classes since they are present in all tables, much work still needs to be done for the other classes.

Acknowledgements

We thank CGG for permission to publish this work. Special thanks to our colleagues in the CGG Data Hub and AI Lab.

References

Brown et al. [2020] Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, **33**, 1877–1901.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. [2019] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT*, 4171–4186.

Koci, E., Thiele, M., Rehak, J., Romero, O. and Lehner, W. [2019] DECO: A Dataset of Annotated Spreadsheets for Layout and Table Recognition. *2019 International Conference on Document Analysis and Recognition*, 1280–1285.

Koci, E., Thiele, M., Romero, O. and Lehner, W. [2019] Cell Classification for Layout Recognition in Spreadsheets. Knowledge Discovery, Knowledge Engineering and Knowledge Management. IC3K 2016. *Communications in Computer and Information Science*, **914**, 78–100.

Kudo, T. and Richardson, J. [2018] SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. *Proceedings of EMNLP*, 66-71.

Nurminen, A. [2013] Algorithmic extraction of data in tables in PDF documents. (Master's thesis, Tampere University of Technology). Retrieved from https://trepo.tuni.fi/handle/123456789/21520

Schuster M. and Nakajima K. [2012] Japanese and Korean voice search. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5149–5152.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I. [2017] Attention is All you Need. *Advances in Neural Information Processing Systems*, **30**

Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., and Zhou, M. [2020] LayoutLM: Pre-training of Text and Layout for Document Image Understanding. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1192–1200.