NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

MSc: DATA SCIENCE AND INFORMATION TECHNOLOGIES

Database Systems

Project Report

Chris Morfopoulos (7115152100011)

George Balaskas (7115152100003)

Supervisor: Georgia Koutrika

Spring Semester

2022 - 2023

Table of Contents

Introduction......3
Summary of the Selected Paper......3
Our Strategy......4
Dataset :......4
Approach of Bert & MCP:......4
Approach of T5 architecture :......5
Bert and Masked Column Prediction......6
T5 model variants......8
Preprocessing & visualisations:......8
Performance of T5 models:......11
Single column annotation Performance:......11
Single column pairwise relationship Performance :......12
Single column Multitask learning Performance :......12
Tablewise annotation Performance:......13
Tablewise relationship Performance:......13
Overall Performance :......14
Conclusion......15

Introduction

In this project our main goal is to explore and investigate all the important findings of our selected paper. The paper that we have selected is “Annotating columns with pre-trained language models” (https://arxiv.org/abs/2104.01785) and its official repository is (https://github.com/megagonlabs/doduo), developed by Facebook. Part of our project is to improve the already pretrained models of the paper as well as to propose alternative ways in order to annotate or predict the relationship between the columns of a table.

Summary of the selected paper

In the paper of “Annotating columns with pre-trained language models” the DoDUO model was first introduced, which is a Bert-based model trained for annotating columns and predicting the relationship between them. Models that are trained for annotating columns are very helpful in many ways and can be used in our daily routine. Missing headers of a table is a common issue in the database world and DoDUO can predict the name of the column based on its context. In the same way, DoDUO can be integrated into a powerful data management tool to propose to the user alternative relations between the respective columns.

In more detail, the DoDUO model is trained both for annotating the columns and for defining the relationship between them via multi task learning. Is a single unified model that can tackle both tasks. DoDUO was trained with the whole table as input to predict the respective column annotations and relationships as sequence classifications.

However a lot of model variations were introduced in this paper such as DoSOLO, which in contrast to DoDUO was trained only with single columns of a table and this have as a result to treat each column independently and not to capture the whole table context.The training datasets for those models are the WikiTable and the Viznet, in which we have serialised column values from many tables.

The authors of this paper mention that only through multitask learning can we achieve the optimal results, implying that only DoDUO has the greatest performance whereas the variations of DoSOLO have lower. They also mention that apart from multi task learning, the table wise model variants can easily outperform the single column model variants.

The performance of the DoDUO model is remarkable, achieving 92% F1-score and 91% F1-score in type prediction and relation prediction respectively. Although the small difference in performance from the DoSOLO can be considered as state of art on Column Type Annotation.

Our Strategy

Dataset

The dataset that we have focused on to train our Models is the WikiTable dataset. In contrast to the VizNet dataset, the WikiTable is much larger .The dataset provides both annotated column types and relations for training and evaluation. For column type prediction, the dataset provides 628.254 columns from 397.098 tables annotated by 255 column types. For column relations, the dataset provides 62.954 column pairs annotated with 121 relation types from 52.943 tables for training. Since we have a sequence multilabel classification task, we should mention that for the column type task we have 255 number of classes, whereas for the column relation task we have 121 number of classes.

Although the WikiTable dataset definitely requires more computational resources than VizNet, it is a very popular dataset for the Column Type Annotation, therefore is crucial for our models to trained with this dataset if not with the whole dataset at least with a partition of it.

Approach of BERT & MCP

Certain DoDUO models are available in the official DoDUO repository and in this way we can implement inference for any dataset. The performance of DoDUO is already extremely high, approximately 90% F1-score, therefore any potential improvement is hard to find.

However, after a lot of research in the Column Annotation field, we encountered a “similar” model called TaBERT, which is also a Bert-based model and it is trained from tabular datasets.

In the training of TaBERT, a certain training technique is first introduced which is called Masked Column Prediction (MCP) and it is applicable to all Masked Language Models (MLM) such as BERT. Instead of masking random words in a sentence, the authors of TaBERT suggest masking random columns of a table. In this way, the model can learn from the table representations and predict the missing values of a column.

In this way, we implemented Masked Column Prediction in the already trained DoDUO model in order to improve its performance. After the DoDUO is finetuned on the WikiTable dataset through MCP, we can make an inference to evaluate the performance of the model.

Approach of T5 architecture

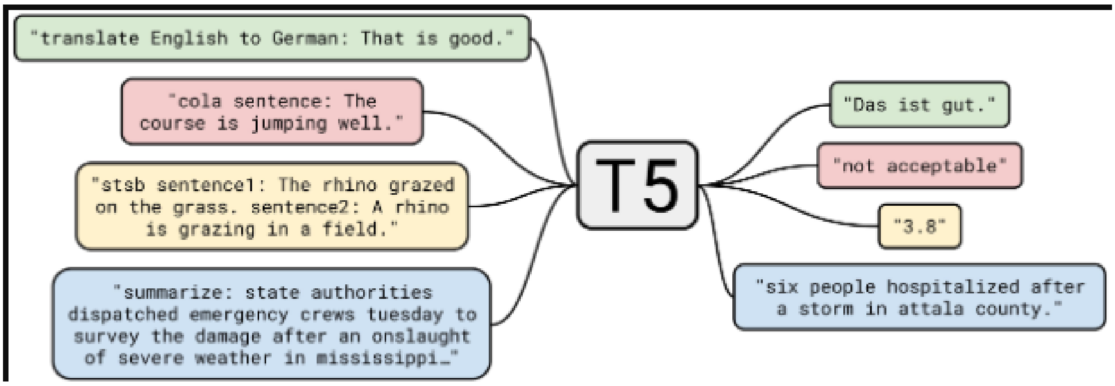
Although the official code of DoDUO is publicly available through its repository we have encountered a lot of issues. **One significant issue was that the default single column settings to train a model led to errors. Therefore we took the initiative to introduce a model, which is based in a different architecture and can be a sequence model**.

In this approach we tried to experiment with the famous T5 (or MT5), which is developed by Google. The T5 is a transformer-based sequence to sequence model, which "models" every problem in text-to-text format, meaning that the input has a text format, and the output has a text format. The applications of the T5 model have a great variety, such as classification, summarization, translation or even regression!

The main difference with the BERT model is that the T5 model uses both the Encoders and the Decoders, as it is implied from the letter "T" (T for Transformers).

A prefix should be applied in the training data in order to define the task that the T5 model should focus on.Due to computational resources, we experimented with the "smallest" version of T5, as the "capacity" of the base model is very high.

Fig. 1 T5 model domains with its prefixes


"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

Despite that we may have a different architecture in our main pretrained model, **we followed the same approach in the training data of WikiTable as DoDUO's**.In the case of tablewise column annotation and relation , we aggregated the data with respect to the tables and concatenated the values adding special token separator of the T5 tokenizer.

Our main goal with this approach is to explore an alternative way of a non Bert-based model and to investigate as well as confirm any similar pattern between the DoDUO and our T5 models.

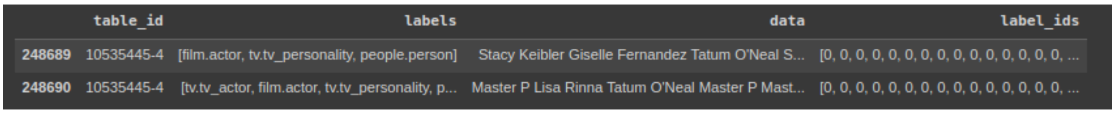
Bert and Masked Column Prediction

As we have discussed above, **we made a significant attempt to improve the performance of the DoDUO model through Masked Column Prediction.** We implemented that particular technique masking a random column so that the DoDUO will learn better and in more depth the representations of each table in the WikiTable Dataset.

Due to the large size of WikiTable Dataset, we defined a cut off point of 250.000 training observations. Only to that particular partition we performed MCP.

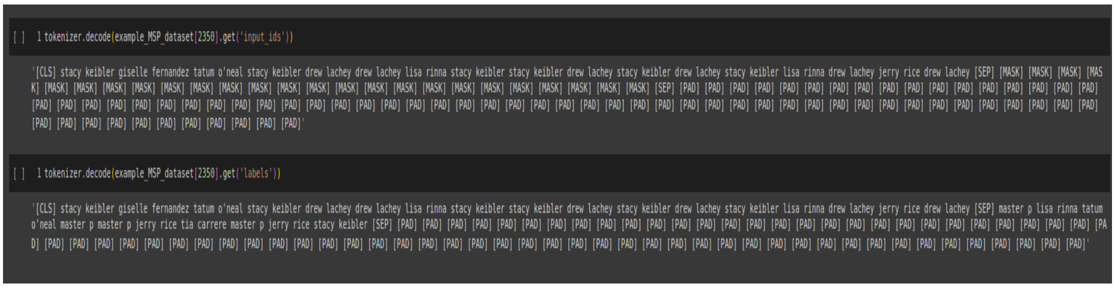
In the figures below, we can visualise a random chosen table from the dataset with its values. Afterwards we mask randomly a column of the table replacing each value with the [MASK] special token. The original values of the tables will be kept as labels in order to calculate the loss in the training procedure.

Fig. 2 Training example of a table from WikiTable dataset with data & labels



	table_id	labels	data	label_ids
248689	10535445-4	[film.actor, tv.tv_personality, people.person]	Stacy Keibler Giselle Fernandez Tatum O'Neal S...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
248690	10535445-4	[tv.tv_actor, film.actor, tv.tv_personality, p...	Master P Lisa Rinna Tatum O'Neal Master P Mast...	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...

Fig. 3 Training example of a table, in which the special tokens (MASK) are applied



```
[ ]   tokenizer.decode(example_MSP_dataset[2350].get('input_ids'))
```

```
'[CLS] stacy keibler giselle fernandez tatum o'neal stacy keibler drew lachey drew lachey lisa rinna stacy keibler stacy keibler drew lachey stacy keibler drew lachey stacy keibler lisa rinna drew lachey jerry rice drew lachey [SEP] [MASK] [MASK] [MASK] [MAS
K] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [MASK] [SEP] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD]
[PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD]
[PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD]'
```

```
[ ]   tokenizer.decode(example_MSP_dataset[2350].get('labels'))
```

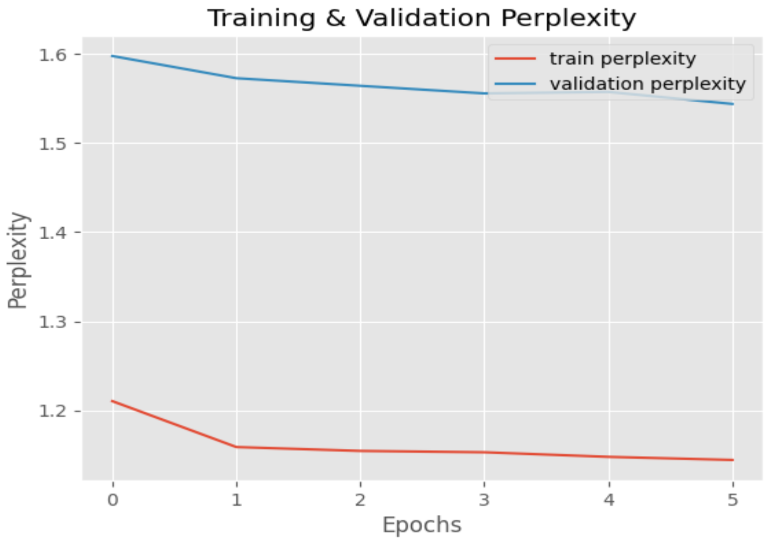
```
'[CLS] stacy keibler giselle fernandez tatum o'neal stacy keibler drew lachey drew lachey lisa rinna stacy keibler stacy keibler drew lachey stacy keibler drew lachey stacy keibler lisa rinna drew lachey jerry rice drew lachey [SEP] master p lisa rinna tatum
o'neal master p master p jerry rice tia carrere master p jerry rice stacy keibler [SEP] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PA
D] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD]'
```

After we have set the mask tokens in the training and the validation dataset, we invoked a vanilla Bert in order to train its encoder.

We train the Bert model for 6 epochs, with a batch size of 64, a learning rate 5e-5 and AdamW as a loss function. A common practice to monitor the performance of Masked Language Models is **to check the training and validation perplexity of the model.** The perplexity formula is just the exponential of the training and validation loss respectively. The lower the perplexity the better the performance of the model.

As we can notice from the figure below, the validation perplexity is much bigger than the training perplexity, but the values of both perplexities were decreasing.

Fig. 4 Visualising the Perplexity of the trained model



After the training is over we replace only the Bert encoder with the DoDUO encoder and we can make an inference on the test dataset of the WikiTable dataset. The evaluation metrics of the fine tuned DoDUO model are micro F1-score of 90,86% and macro F1-score of 73%, whereas the evaluation metrics of the original DoDUO in the test dataset are 90,71% and 72,4% respectively. We can conclude that there is a small increase of 0.15% in the micro F1 metric and 0.6% increase in the macro F1 metric.

To sum up, the improvement of the DoDUO model may be small, but the positive output is that the integration of the Masked Column Prediction technique can be beneficial for any Column Annotation model.

T5 Model Variants

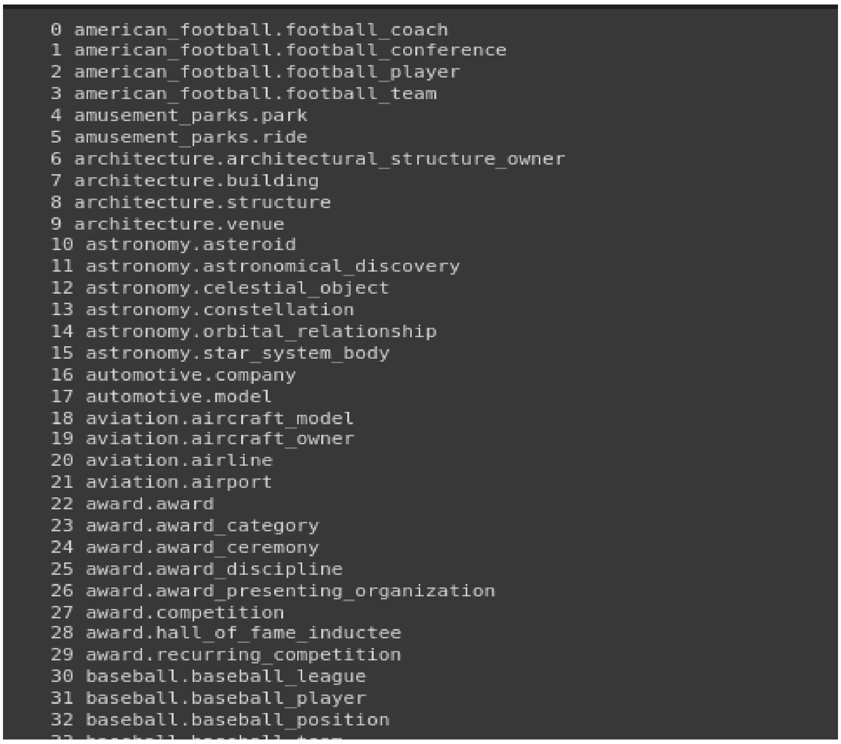
Preprocessing & Visualisations

As we mentioned above, for each T5 model certain requirements and modifications have to be applied in the training and validation datasets. For the T5 single column models, the training data consist of single column values whereas for the T5 multi column models, the training data consist of concatenated column values of a table.It is obvious that the tablewise data are concatenated and connected with a special token separator of T5 tokenizer. In the same way, we concatenate and connect the labels as it is a multilabel classification task.

In order to save computation cost, we defined a training threshold of 150000 observations and a validation threshold of 3000, as the training columns of WikiTable are extremely large, more than half million!

Apart from the necessary preprocessing actions, we applied a respective dictionary for each task, collecting as key-values all the number of classes of each task. We should remind that for the column type we have 255 classes and for annotation type we have 121 classes.Finally, we replace each label with the index of its respective dictionary.**The main reason that we implemented that particular dictionary, is to alleviate the decoder of the T5 to generate a smaller and not so complex string.**

Fig. 5 A partition of the calculated dictionary with its classes



```
0 american_football.football_coach
1 american_football.football_conference
2 american_football.football_player
3 american_football.football_team
4 amusement_parks.park
5 amusement_parks.ride
6 architecture.architectural_structure_owner
7 architecture.building
8 architecture.structure
9 architecture.venue
10 astronomy.asteroid
11 astronomy.astronomical_discovery
12 astronomy.celestial_object
13 astronomy.constellation
14 astronomy.orbital_relationship
15 astronomy.star_system_body
16 automotive.company
17 automotive.model
18 aviation.aircraft_model
19 aviation.aircraft_owner
20 aviation.airline
21 aviation.airport
22 award.award
23 award.award_category
24 award.award_ceremony
25 award.award_discipline
26 award.award_presenting_organization
27 award.competition
28 award.hall_of_fame_inductee
29 award.recurring_competition
30 baseball.baseball_league
31 baseball.baseball_player
32 baseball.baseball_position
```

In order to understand in depth the training data, we employed certain visualisation techniques such as Word Cloud and count of frequencies. From the figures below, we can notice not only the column values of the Word Cloud, but also the frequencies of the top 20 and least common 20 labels in column type and column relations. The duplication of values in the Word Cloud, is a result of the repetition in the column values in certain tables.

Fig. 6 A WordCloud of all the column values

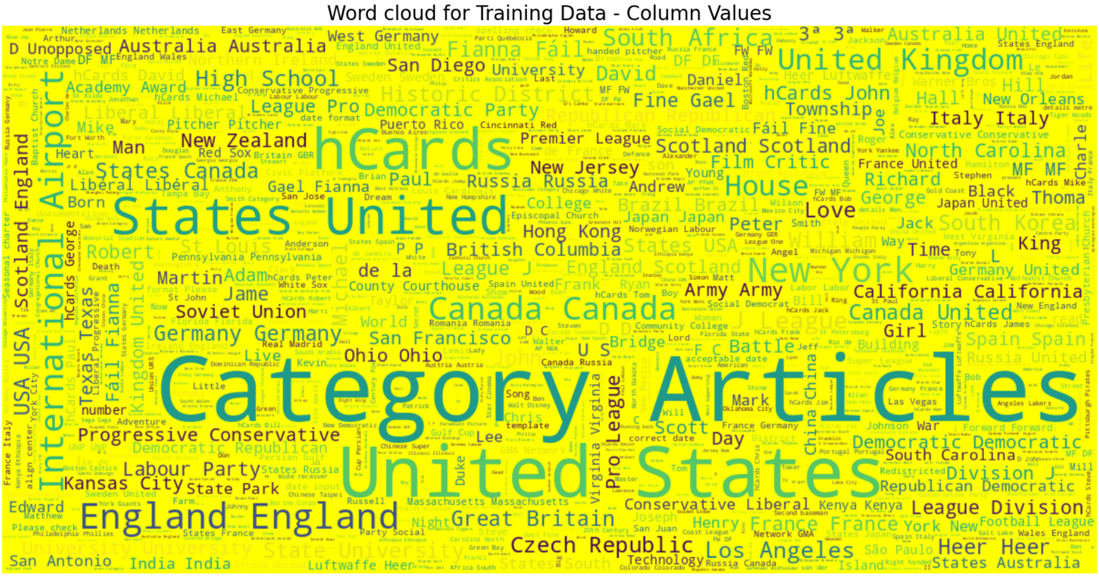
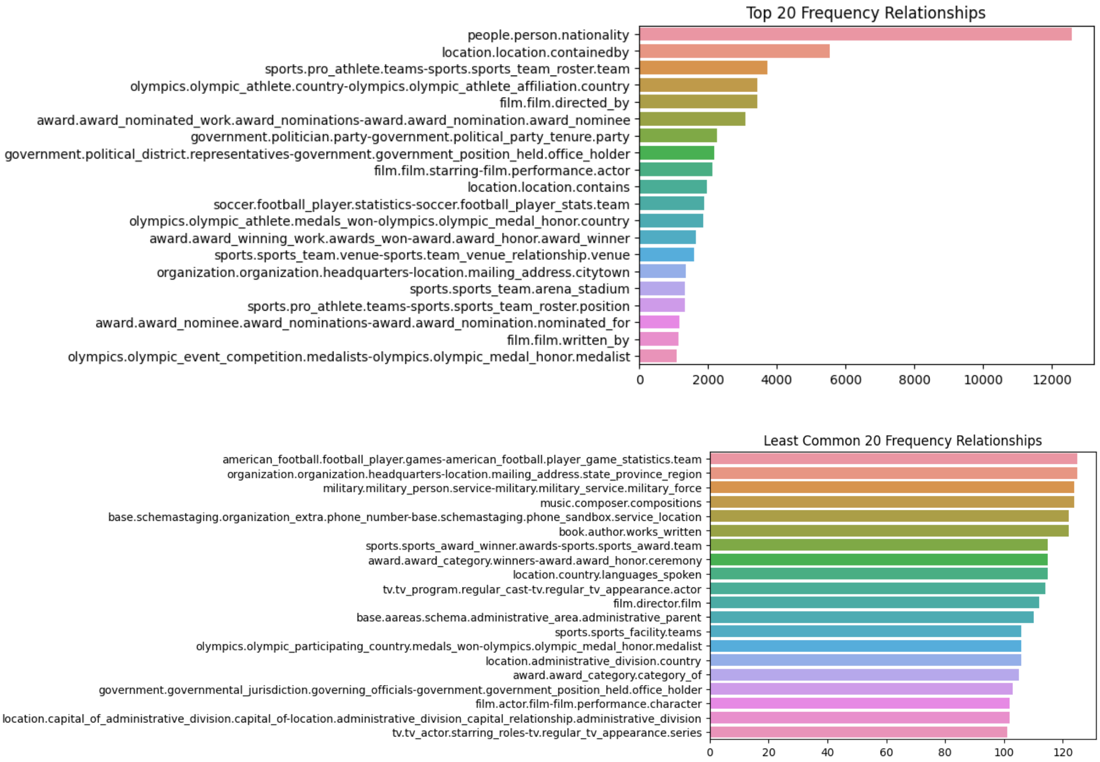
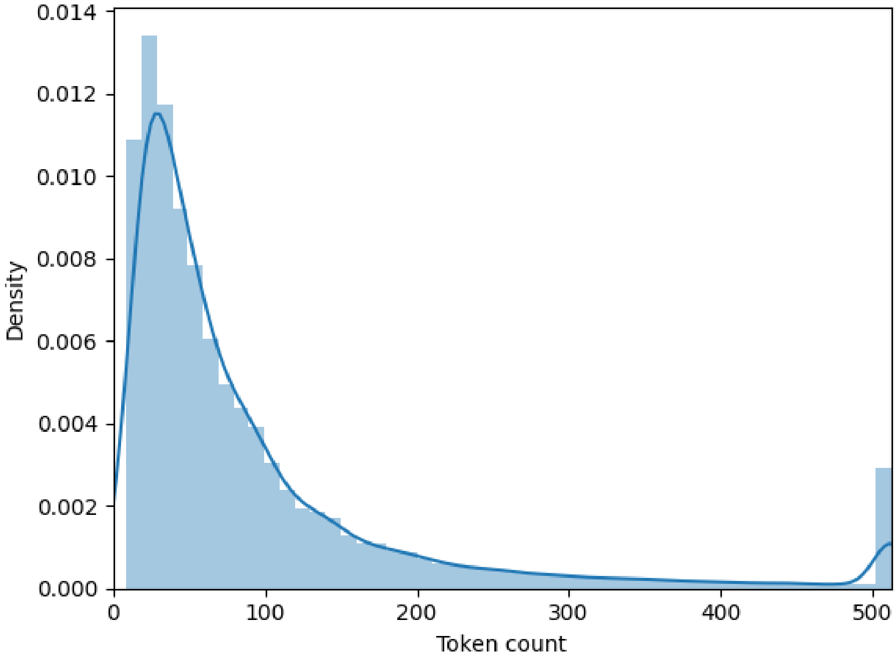


Fig. 7 The Top 20 and Least Common Frequency labels in column relation task



In order to set the optimal source length for the decoder of T5, we should temporarily visualise all the lengths of the tokens. From the distribution graph below we can conclude that the majority of the distribution lies on the left hand side of the graph, therefore we set the length of the T5 source length to 180. **We should mention the bigger is the decoder length, the more computation expensive is the T5 model.**

Fig. 8 The distribution of the length of the tokenized column values



Performance of the T5 models

With the objective to pick the best T5 model in terms of performance, all T5 model variants should be comparable. To achieve that, the training settings for all the experiments were the same. In more detail, we trained every T5 model variant with 10 epochs and the same learning rate of 5e-5 as DoDUO. **This has as a result to perceive homogeneity among the models.**

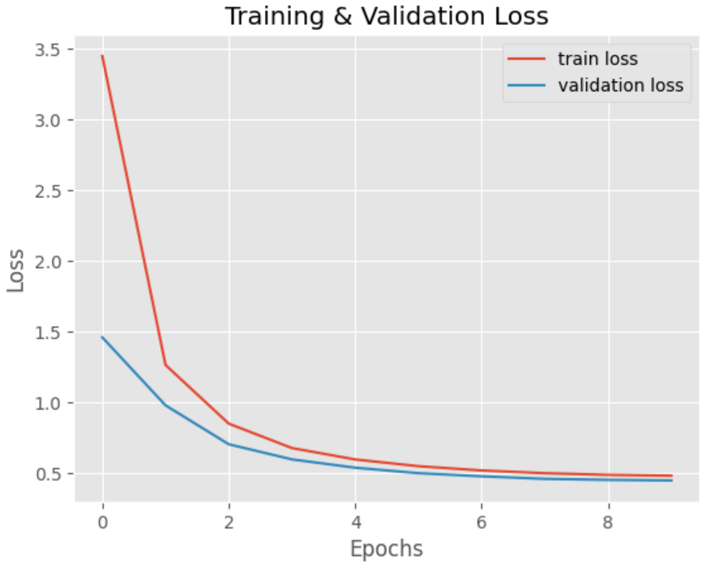
More or less, all the models follow the same approach in their respective pipeline. Only the Multi Task Learning T5 model may differ, in which we have 2 different data loaders one for each task. The training of the MTL model consists of both data loaders across the epoch. Therefore, we can have a single Loss Function, where we can combine the losses of each task with equal weights. We should mention that , the DoDUO utilises multi-tasking learning from a different perspective, by sharing common layers of the BERT model, which is not the conservative way.

Obviously, the best T5 model is that which will reach the best validation evaluation metrics (F1 score) in the validation dataset across the epochs.

Single column annotation Performance

Across the training epochs, the best validation F1 score of the T5-single column annotation model is 74,1% and accuracy of 50%.

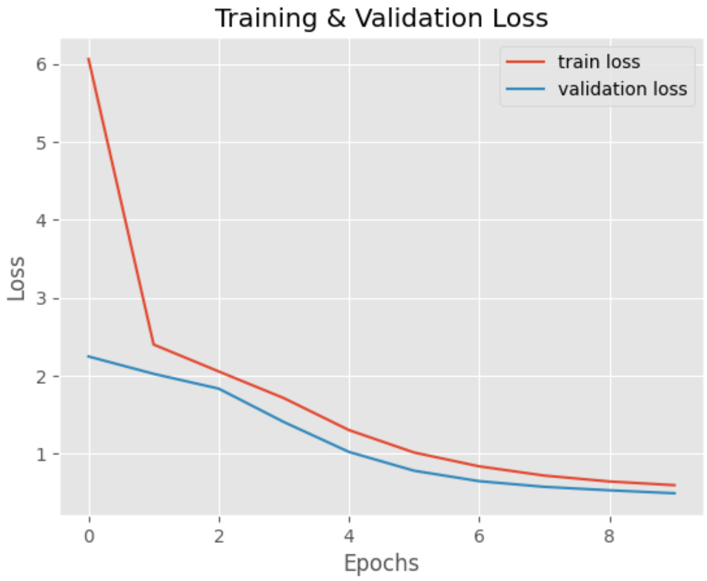
Fig. 9 The training and validation loss of the single col annotation model



Single column pairwise relationship Performance

Across the training epochs, the best validation F1 score of the T5-single column relation model is 74,4% and accuracy of 67%.

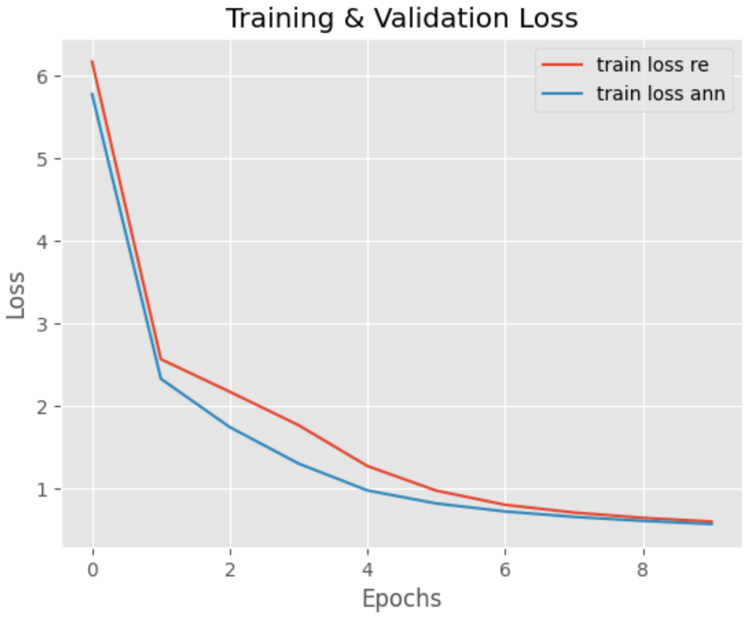
Fig. 10 The training and validation loss of the single col relation model



Single column Multitask learning Performance

Across the training epochs, the best validation F1 score of the T5-single column MTL model is 67% and accuracy of 40% for the column type task, and F1 score of 75,8% and accuracy of 67% for the column relation task

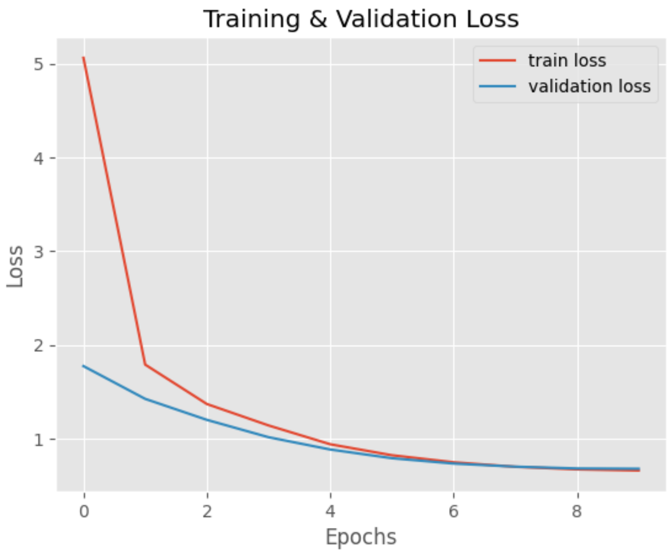
Fig. 11 The training and validation loss of the single col MTL model



Tablewise annotation Performance

Across the training epochs, the best validation F1 score of the T5-multi column annotation model is 47% and accuracy of 11%.

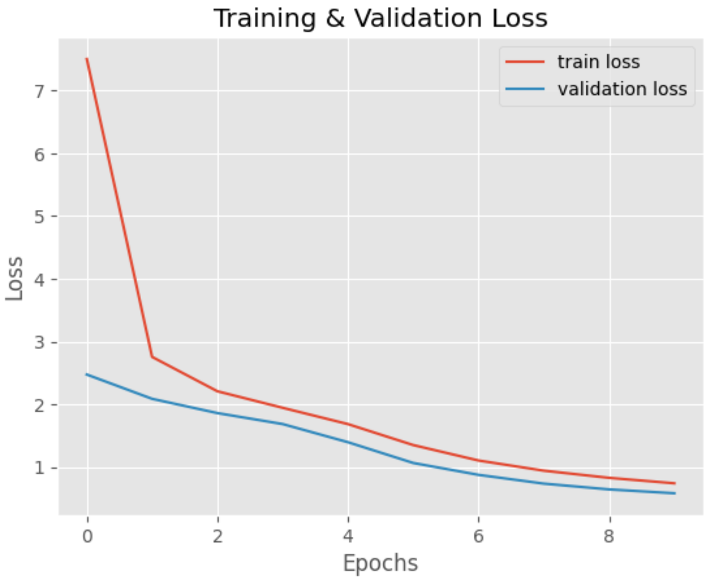
Fig. 12 The training and validation loss of the tablewise annotation model



Tablewise relationship Performance

Across the training epochs, the best validation F1 score of the T5-multi column relation model is 63% and accuracy of 47%.

Fig. 13 The training and validation loss of the tablewise relation model



Overall Performance

At a quick glance, from the figures above of training & validation loss, we can conclude that firstly, **the single column models outperform the tablewise models in every aspect**. The tablewise models mark very low F1 scores in contrast to the single column models.Secondly, the **Multi task single column model is the best in terms of performance**, which **implies that indeed the multi task learning plays a significant role** and the model can learn patterns from both relative tasks. In more detail, the Multi task learning model achieves a remarkable 75.8% F1 score in the column type task, whereas the single column model marks a 74.4%.

Fig. 14 An overall graph with all the F1 validation scores of the T5 models

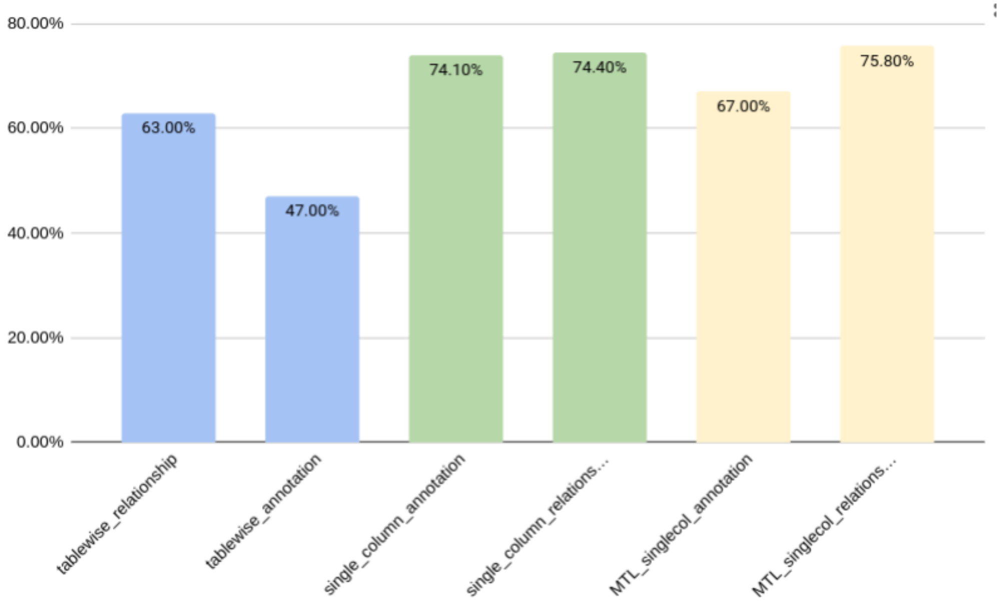


Fig. 15 A table of DoDUO -DoSOLO performance from the original paper

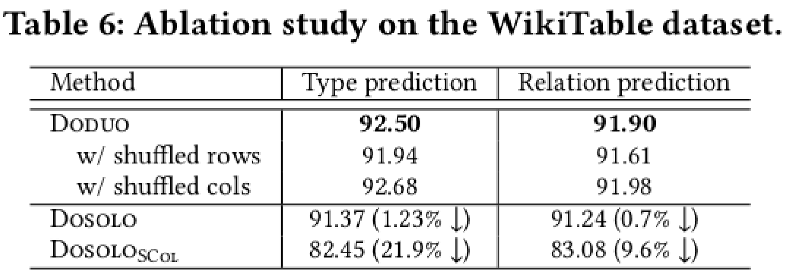


Table 6: Ablation study on the WikiTable dataset.

Method	Type prediction	Relation prediction
Doduo	**92.50**	**91.90**
w/ shuffled rows	91.94	91.61
w/ shuffled cols	92.68	91.98
Dosolo	91.37 (1.23% ↓)	91.24 (0.7% ↓)
Dosolo$_{\text{SCol}}$	82.45 (21.9% ↓)	83.08 (9.6% ↓)

Conclusion

In conclusion, from the T5 experiments we can easily conclude that all the model variants have the “pattern” in their respective training phase, considering the above graphs of training & validation loss. A key point is that the validation loss was decreasing, which implies that each model was continuously learning and had the capacity for even more. Nevertheless, the main novelty that was described in their paper and we verified it as well, is that via Multi Task Learning the model can actually perform better in the tasks of column annotation & relationship and it should be considered as a standard approach for annotation problems.