Talk to me: Summarization for Database Vocalisation with Bert and ProphetNet

Ankit Mukherjee, Apoorva Rani, Nishitha Nancy Lima, Oliver Nitin Watson, Subhajit Mondal, Surabhi Katti

***Abstract*—This paper proposes and evaluates a transformer-based summarization model on a generated dataset, with the intent of applying it to summarise query results. These summaries can then be vocalised to use on devices without screens or for visually impaired users. For our study we compared two different transformer models, a traditional Bert model adapted for summarization, and a multi-stream based model. First we conducted a scoped study on the impact of the different model parameters for our task. We also generated a dataset based on WikiSQL and conducted early experiments with our fine tuned models on this dataset, validating the potential of applying transformer models to database summarization for vocalization.**

***Index Terms*—Summarization, Transformers, Database Vocalization, Bert, ProphetNet, ROUGE**

I. Introduction

Today, there is a need for specialised experts on SQL or similar languages to understand and query large databases and extract valuable information from these systems. Then, we need another set of experts to understand the data extracted and summarise it in simpler terms for the end user. However, the way we interact with data is constantly changing and with the large amount of data being generated everyday, there is a strong need to bridge this knowledge gap for Data Analytics through the use of simple tools that can work without complex schema knowledge or SQL. In addition, with new devices coming up everyday which may not have a screen for interaction, we will eventually have to find a effective Data Management system for such devices. Both aspects suggest the importance of working on techniques to support voice-based interfaces.

There have been many breakthroughs and ongoing research in converting Natural Language to SQL Queries to help non-technical users to make use of available data. What we do not have though is a vocalisation system that would effectively summarise the query results and read them out to the end user. This would not only reduce dependency on experts but also help visually impaired users to work and interact with the available data.

Achieving this goal requires us to look at some key concerns and to find effective solutions for them: How to vocalise results of SQL Queries? How to effectively describe a table of thousands of records to the user?

To approach this task we propose to use state of the art NLP transformers, which are employed for data summarization. To this end the models are first trained on an example dataset, generated by us. Our resulting trained transformer models should summarise query results in a way that they can be vocalised and comprehended without visual aid. In this paper, we have looked at traditional as well as multi-stream state of the art summarization models and compared their performances for our task.

Our contributions are as follows:

1) We offer a scoped study on the impact of the different parameters on 2 transformer models (Bert and ProphetNet) for our summarization task.
2) We propose a novel dataset generation method for the summarization task for database queries, based on the WikiSQL dataset. The dataset is availble on our github repository [1]
3) We show early results on how these transformer models work on our dataset.

The remainder of the paper is structured as follows: In Sec. II we cover the background for our work, describing work on summarization and on transformers. In Sec.III we describe the research questions of our study and our method of dataset generation. In Sec. IV we cover the implementation of our work, describing our datasets and model details. In Sec. V we present the results of our study. We conclude in Sec.VI with the conclusion and suggestions for future work.

II. Background

A. State of the Art

This sub-section explains the current techniques in the area of voice-based text summarization. Text Summarization is indeed a complicated task for a computer because we as humans tend to read the entire content and develop a understanding to paraphrase it in short sentences. Automatic text summarization is the task of producing a concise and fluent summary while preserving key information content and overall meaning.[8]. Since computers lack the understanding that humans possess automatic text summarization gained attraction as early as the 1950s. An important research of these days was [9] for summarizing scientific documents. Luhn et al. [9] introduced a method to extract salient sentences from the text using features such as word and phrase frequency. They proposed to weight the sentences of a document as a function of high frequency words, ignoring very high frequency common words. Data vocalization is the process of summarizing data via voice output. All the research of the database community has been extensively focused on visual output. Until recently there has been a shift towards vocalised output. First voice-based query interfaces have appeared quite recently [11, 10]

[1]https://github.com/0liverWatson/Summarization-and-Database-Vocalization

but they targeted smaller datasets. Another research was Data Vocalization: Optimizing Voice Output of Relational Data [3]. The goal of this paper was to present relational data in the most efficient way as voice output. The focus was on to have voice generation as an approximation problem: minimize speaking time while approximation of relational table to user and keeping the constraints - precision of transmitted data and cognitive load placed on the listener.

B. Transformers:

This sub-section deals with the basic idea of transformers and its applications. Transformers as a basic idea is an architecture for transforming one sequence of characters into another, with the help Encoder and Decoder,and employs attention mechanism instead of any RNN networks(LSTM,GRU). Transformers are the current state-of-the-art type of model for dealing with ML tasks related to sequence data. The most common application of these models is machine translation. There are pre-trained models that can be fine-tuned to be used in projects. The main idea behind the transformer models is self-attention. Self-attention, in simple words, is an operation on sets. In this type of attention mechanism, the model relates different positions of the single sequence, to compute a representation of the same sequence. Text summarization is usually done though extractive summarization and abstractive summarization [12]. Extractive summarization deals with extracting keywords,clauses, sentences or paragraphs, which are then put together to form a summary. Another approach is understanding the original text and retelling it in few words. Most of the transformer models adopted to the task focus on abstractive summarization.

C. Choosen Transformer Models

The main motivation was to go with the model that could understand the context given to it and then summarize rather than dealing with extracting keywords and building statistics on it ,Abstractive summarization was our focus and the models chosen for the summarization task are *Bert* and *ProphetNet*.

1) ***Bert*** *: BertSUM* is an extended form of a state of the art Transformer model BERT which handles long input and word to word relationships.*BertSUM* is the first specialized text summarization model using BERT as encoder.

The key Idea of *BertSumExtAbs* is the two-stage fine-tuning approach :

- Fine-tune the encoder on an extractive summarization task, then
- Fine-tune it on an abstractive summarization task.

a) ***Extractive Summarization*** *:* It deals with task of assigning a label 0 or 1 to each sentence, indicating whether the sentence should be included in the summary. With *BertSUM*, the vector of the specific [CLS] symbol from the top layer can be used as the representation for each sentence. Several inter-sentence Transformer layers are then stacked on top of Bert outputs, to capture document-level features for extracting summaries.

b) ***Abstractive Summarization*** *:* Consists of a Standard Encoder-decoder Framework. The encoder is the pretrained *BertSUM* and the decoder is a 6-layered Transformer, initialized randomly. To avoid unstable fine tuning, authors designed a new fine-tuning schedule which separates the optimizers of the encoder and the decoder
BertSUM uses 2 training strategies:
Masked LM (MLM): 15 percent of the words in the sequence are replaced with a [MASK] token, before feeding word sequences into BERT. Then the model attempts to predict the original value of the masked words, based on the remaining non-masked, words in the sequence[5].
NSP (Next Sentence Prediction) : Here, the model receives pairs of sentences as input and learns to predict the order, ie if the second sentence in the pair is the subsequent sentence in the original document. During training, the inputs are mixed up such that 50 percent of the inputs are a pair in which the second sentence is the subsequent sentence in the original document, while in the other 50 percent a random sentence from the corpus is chosen as the second sentence, with the assumption that the random sentence will be different from the first sentence [5].

2) ***Prophet-Net****: Prophet-Net* is a transformer based summarization model that introduces a novel self-supervised objective named future n-gram prediction and a proposed n-stream self-attention mechanism [6].

a) N-gram prediction: Traditional language models and Seq2Seq models are trained by teacher forcing. The models are optimised to predict the next token given all previous context tokens at each time step. However in addition to the this, one step ahead prediction, Prophet-Net also learns n-step ahead prediction. This future n-gram prediction is served as extra guidance that explicitly encourages the model to plan for future tokens and prevents over-fitting on strong local correlations.

b) N-stream self attention: Prophet- Net contains a main stream self-attention, which is the same as the self-attention in the original Transformer. Besides, they introduced n extra self-attention predicting streams for future n-gram prediction, respectively. During training, the i-th predicting stream attends to the main stream's hidden states to predict the next i-th future token, which guarantees every n continuous tokens in the target sequence are trained to predict at one time step.

c) Datasets used: Prophet-Net is pretrained using the base scale database of 16GB (as used in BERT) and the large-scale dataset of 160gb (similar to BART) respectively. It is then experimented on CNN/DailyMail, Gigaword and SQuAD 1.1 benchmarks for abstractive summarization and question generation tasks.

d) Performance: Prophet-Net achieves the best performance on both abstractive summarization and question generation tasks. Furthermore, Prophet-Net achieves new state-of-the-art results on CNN/DailyMail and Gigaword using only about 1/3 of the pre-training epochs used for the previous model.

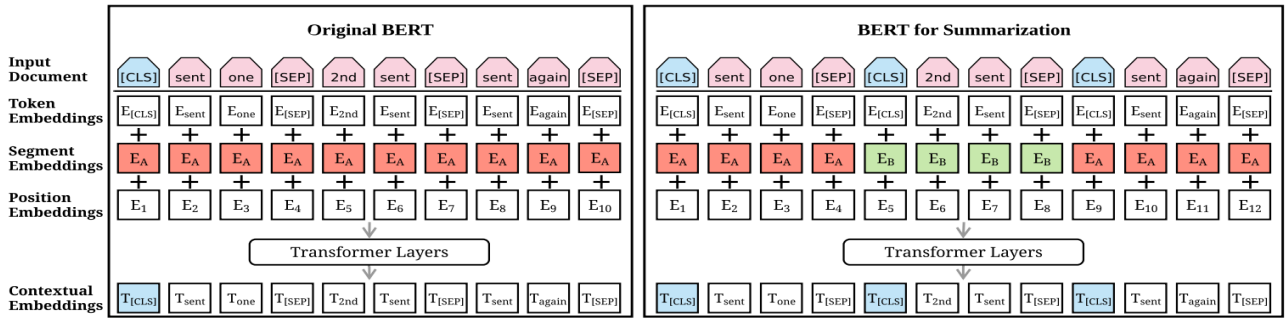

Fig. 1: Differences in Bert and BertSUM[5]

III. Prototype Design

For our project, the research questions concentrate primarily on the degree to which the parameters of the transformers influence the fine-tuning of the data set used.

A. Research Questions

1) How do the selected models (Bert and ProphetNet) compare when fine-tuning on an unseen summarization text dataset, and what parameters have an influence on the model performance?
2) How do the selected models compare on their best parameters, when fine-tuned on our generated dataset for the database summarization task?

Initially, the fine-tuning of the parameters is to be performed on a dataset sample, and then, depending on observations and tests, it is extended to the dataset as a whole.

B. Structure/Design

The key principle of summarizing database query results is to understand how optimal the transformer models operate on datasets consisting of relational database outputs.T he complete flow diagram indicating each step involved in generating database summaries is shown in figure 2. The architecture is accompanied by a collection of transformer models *Bert* and *ProphetNet*. Initially, both the models are used for summarization of text and later on the basis of their performance, the best training configurations of these models is used for summarizing the actual database. Experiments were performed by training both the models on the Reddit TIFU dataset. Parameters were fine-tuned depending on the training loss and cost function values. The evaluation metrics used for the decision purpose are discussed in the following section. The model which gave considerably good results on basis of quality and accuracy was selected. The dataset for our second research question is generated from the WikiSql database with predefined principles. Based on rules summaries were formed which later were used as target values for the transformer model. The detailed explanation of generating datasets is given in a further segment. The selected Bert model with fine-tuned parameters is trained using the generated WikiSQL dataset.

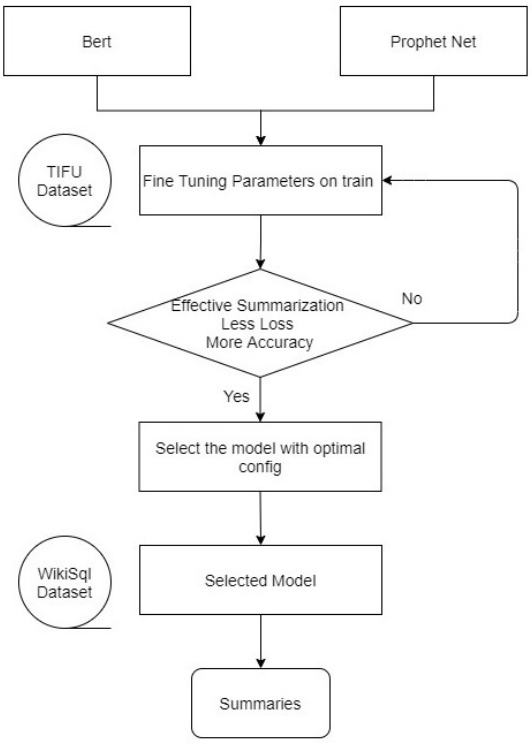

Fig. 2: Flow Diagram stating steps involved in generating Summaries from TIFU and WikiSQL dataset using BERT and Prophet-Net transformer models

C. Dataset Generation (for Database Summarization)

The dataset is generated from the WikiSQL database by applying some set of production rules for the cases of each of cell, row or column or table summary. The process is split into two parts, viz the input variable generation and the output variable generation.

1) Input Variable Generation: The general structure of the input variable is as follows <Table Name>,<Query Type>,<row>,<col>,<table data>,<stats>

a) Table Name: Represents the table name. The interplay of Query Type, row and col is indicated in the II.

b) Table Data: To illustrate the encoding used for table data we may consider the (partial)table named **Zakspeed** [I]

For each of the 4 scenarios of cell, row, col or table summarization, the following are the rules.
Cell data is encoded as col:data;
Row Representation: col:data;col:data;
Column data is encoded as col:data;data;. . . .more data;
Table data is encoded as col:data;; where ; marks the end of a cell data and the other ; marks the end of a row
Examples of above encoded data format is given below.

cell data of row1	Chassis:Zakspeed 841;
row summary of row1	Year:1985.0;Chassis:Zakspeed 841;
col summary of Year	Year:1985;1986;
table summary Zakspeed	Year:1985.0;Chassis:Zakspeed841;; Year:1986.0;Chassis:Zakspeed 861;;

c) Stat: : Finally the stat consist of 3 subparts, Percentage, mean, median, which are put in in a ; separated fashion and only for column summary. For the data that is encoded, please check column target data generation.

Year	Chassis
1985	Zakspeed 841
1986	Zakspeed 861

TABLE I: Partial table named Zakspeed from WikiSQL

2) Target Data Generation: For the target variable we have a set of rules. III underlines the rules that were followed to generate the target summary

D. Evaluation Metrics

Evaluation metrics are generally used to evaluate the performance of model on an unseen dataset. Since our task here deals with the generation of summaries one of the best choices was to use the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) score. This measures the quality of a summary created by comparing it to the target summaries [13]. This score counts the number of overlapping units such as n-gram, word sequences, and word pairs between the generated summary which has to be evaluated and the target summaries. (1). Higher Rouge scores indicate better performance.
ROUGE-N — measures the n-gram overlaps like unigram(n=1), bigram, (n=2),trigram(n=3) and so on[13].
ROUGE-L — measures longest matching sequence of words using Longest Common Subsequence(LCS). An advantage of using LCS is that it does not require consecutive matches but in-sequence matches that reflect sentence level word order [13].
ROUGE-S — measures any pair of words in a sentence in order, with the allowance for arbitrary gaps. This can also be called skip-gram concurrence[13].
For example, ROUGE-1 refers to overlap of unigrams . ROUGE-2 refers to the overlap of bigrams and so on.

Query type value	Meaning	Row	Col
0	Table summary	-1	-1
1	Row summary	Row id of the row	-1
2	Column summary	-1	Column name of the column
3	Cell summary	Row id of the row	Column name of the column

TABLE II: Mapping of Query Type and Row and Col variable

Summary Of	Rules
Cell	The cell contains more than 50 characters then read
Row	Read first column
	Out of the rest, sort the columns by entropy and half of the columns that achieved the highest entropy are chosen and their data is read
	If a cell has a lot of text we don't read it
	Replace column name containing punctuations (underscores) with spaces and remove the parenthesis and text inside the parenthesis
Column	If a column is textual and has low entropy Read out the probability distribution of these values
	If probability less than 10% - assign it to others
	If a column is numerical and has high entropy, If small precision - Read a number of unique values, If large precision - Read low, high, and median values
	Replace column name that has underscores to spaces and don't read data in parenthesis
	If a cell has a lot of text we don't read it
Table	3 columns of the table are selected.
	Select the first column
	For the last two columns, sort the columns by entropy and select the column with the highest entropy, such the selected columns are unique.
	The selected columns are described as per the strategy to describe columns

TABLE III: Rules to generate summarization target

IV. Implementation

A. Datasets

The evaluation of transformer models were done by using two competitive datasets TIFU and our generated dataset based on WikiSql.

1) TIFU: TIFU dataset is collection of web postings from the online discussion forum Reddit [1]. These are derived from 120K posts online and consist of discussions over diverse topics. We deal with only part of these conversations collected for limited period and thus it is also called as TIFU SubReddit. It is divided into TIFU-short and TIFU-long datasets on the basis of the length of summaries. An example of summaries present in TIFU is shown in Figure 4. For the fine-tuning of parameters on transformers models Bert and ProphetNet, we consider the TIFU-long summaries dataset.

2) WikiSQL: This is a large crowd sourced dataset which deals with natural language interfaces on relational database [2]. Data is represented as collection of SQL queries and SQL tables. The tables present in the dataset are extracted from HTML tables from Wikipedia. The magnitude of the dataset is enormous as it contains 26,531 tables and 8,654 hand-annotated examples. The SQL queries are distributed over the dataset. An example of WikiSQL table is shown in figure 5. For our data generation we did not use the queries, but instead

[Short Summary] (16 words)
TIFU by forgetting my chemistry textbook and all of my notes in a city five hours away

[Long Summary] (29 words)
TL;DR I forgot my chemistry textbook and binder full of notes in Windsor, which is five hour drive away and I am now screwed for the rest of the semester.

[Source Text] (282 words)
(…) So the past three days I was at a sporting event in Windsor. I live pretty far from Windsor, around a 5 hour drive. (…)
A five hour drive later, I finally got back home. I was ready to start catching up on some homework when I realized I left my binder (which has all of my assignments, homework etc.) in it, and my chemistry textbook back in Windsor.
I also have a math and chem test next week which I am now so completely screwed for. (…)

Fig. 3: TIFU SubReddit Example for long and short memories

attempted to do summaries over sections from the tables under study.

3) Generated Dataset: The WikiSQL Dataset is customized to be adapted for our proposed models. The dataset is generated from the rules described in Section III C. A script was built to randomly select table, row, column or cell summary and then picking the tables form the WikiSQL dataset. The rules are then applied to generate a valid record in the form of a $\langle x, y \rangle$ pair where x represents the data points or indexes of the relational tables combined with query type and statistics and y represents the target summaries formed using the scripts. For the experiments a dataset with 10000 records was generated.

Number	Artist	Album	_1st_week_sales	_1st_week_position
1.0	Kanye West	Graduation	957,000	#1
2.0	50 Cent	Curtis	697,000	#2
3.0	T.I	T.I vs. T.I.P.	468,000	#1
4.0	Jay-Z	American Gangster	426,000	#1
5.0	Fabolous	From Nothin' to Somethin'	159,000	#2
6.0	Common	Finding Forever	158,000	#1
7.0	Lupe Fiasco	The Cool	143,000	#14
8.0	Young Buck	Buck the World	141,000	#3
9.0	Timbaland	Shock Value	138,000	#1
10.0	Bone Thugs-N-Harmony	Strength & Loyalty	119,000	#2

Fig. 4: Table showing data of hip-hop music in 2007 from WikiSQL

B. Models

The existing models needed to be fine tuned for our task. We tuned the following parameters to test our models and eventually make them suitable for summarising on our generated dataset:

1) Min and Max length: The minimum and maximum number of words the summary should contain.

2) Temperature: The randomness of predictions. If the temperature is more, it is less likely to sample from unlikely candidates. As the temperature values decrease, it is more likely to sample from all candidates.

3) Number of Beams: The n most probable next words as candidates to be sampled as opposed to greedy search which returns the most probable next word.

4) Length Penalty: Exponential penalty to the length of the sentences of the summary generated. A value of 1 indicates no penalty. As the value increases, longer sentences are less likely to be selected for summarising.

5) Repetition Penalty: Exponential penalty to the repetition of words of the summary generated. A value of 1 indicates no penalty.

6) Diversity: Encourages the summaries to be more diverse. The higher the value, more diverse are the outputs.

V. Results and Discussion

A. Best models for Fine tuning on a general summarizing sample dataset

To understand the impact of the hyperparameters, we picked a small sub-slice of the dataset of 1000 records and trained both the models for 10 epochs for each of the different hyper parameter settings. The different values of the parameters in these experiments are listed in Table IV. The min and max length for the Reddit TIFU Dataset was set to 20 and 130 respectively, considering the characters in the target summaries

Parameters	Values
Temperature	1.0 , 0.6 , 0.3
Length Penalty	1.5 , 1.2 , 1.0
Repetition Penalty	1.0 , 1.2
Number of Beams	4 , 1
Diversity Penalty	0.0 , 0.2

TABLE IV: Parameter Values used for tuning the models

The final ROUGE-1, -2, and -L scores of the models with the best scores for the values of the parameter are shown in Table V with the selected hyperparameter values shown in table VI for both models. Increasing the number of beams did not seem to have an impact on the BertSum model, while results improved for the ProphetNet model. We theorize this is due to the ProphetNet model already having a multi-stream prediction architecture, and its inclusion of further branching possibilities seems likely to improve the performance of the model.

Model	ROUGE-1	ROUGE-2	ROUGE-L
ProphetNet	22.3762	5.1160	17.2089
BertSum	17.2280	2.9850	12.9210

TABLE V: ROUGE-1, -2 and -L scores for the models on the Reddit TIFU slice

The ProphetNet model performed better than the BertSum model in most of the experiments, and as as seen from Table V performed almost 30% better on the best configurations of both the models.

Parameter value	ProphetNet	BertSum
Temperature	0.6	0.6
Length Penalty	1.2	1.2
Repetition Penalty	1.0	1.0
Number of Beams	4	1
Diversity Penalty	0.0	0.0

TABLE VI: Values of Hyper Parameters used in the experiments

B. Best model config for the complete Reddit TIFU dataset

The hyper parameter values selected from the previous experiments (Table VI) were then used to train the models on the entire dataset, that is the Reddit TIFU Long dataset consisting of 42139 records. The models were trained for 100 epochs. As observed in the previous experiments, we hypothesise that the ProphetNet model would perform better on the entire dataset as well.

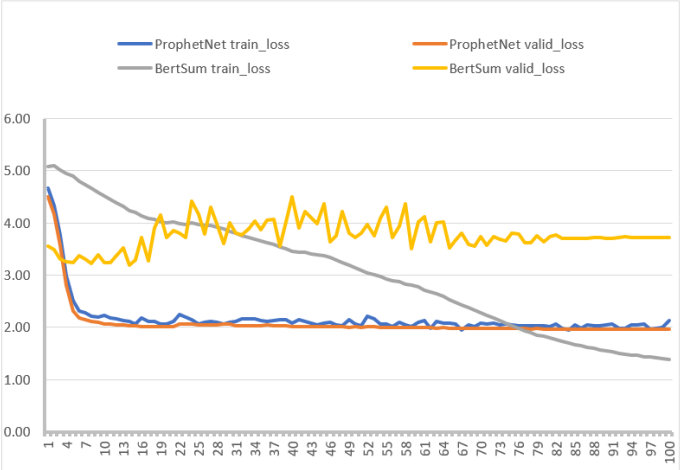

Fig. 5: Training and Validation loss curves for ProphetNet and BertSum trained on the Reddit TIFU Dataset

The training and validation losses for the ProphetNet and BertSum models are shown in Figure 5. As seen from the figure the BertSum model seems to overfit on the dataset. This was also observed in the predicted summaries where the same predicted summary of a random set of words were generated for all input records, shown in Appendix **??**. Further experiments need to be conducted to analyze this behaviour with the traditional BertSum model.

Model	ROUGE-1	ROUGE-2	ROUGE-L
ProphetNet	22.3762	5.1160	17.2089
BertSum	17.2280	2.9850	12.9210

TABLE VII: ROUGE-1, -2 and -L scores for the models on the entire Reddit TIFU dataset

The ROUGE scores for both the models are shown in Table VII. The ProphetNet model has obtained better scores on all three of the metrics compared to the BertSum, confirming our hypothesis.

C. Fine-tuning best model on our Generated Dataset

The best parameters for each model were picked, shown in Table VI and then trained on the generated dataset for 25 epochs considering the dataset size of 10000 records. Two percent of the records (2000) were used as the validation split. The max length and min length parameters were changed as the text is the target summaries was considerably longer than the Reddit TIFU Dataset. min length of 30 and max length of 300 was used in the below experiments. Our hypothesis is that the ProphetNet model would perform better as observed on the Reddit TIFU dataset.

The training and validation loss curves are shown in Figure 6, the progression of the ROUGE scores for the ProphetNet model are shown in Figure 7, and some of the examples of predictions are shown in Appendix **??**

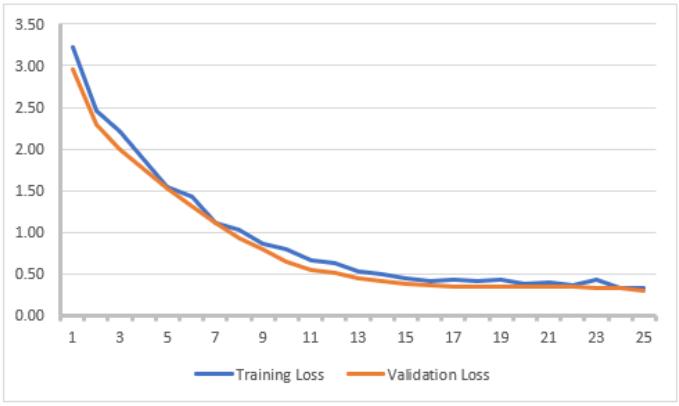

Fig. 6: Training and Validation loss curves for ProphetNet trained on our Generated Dataset

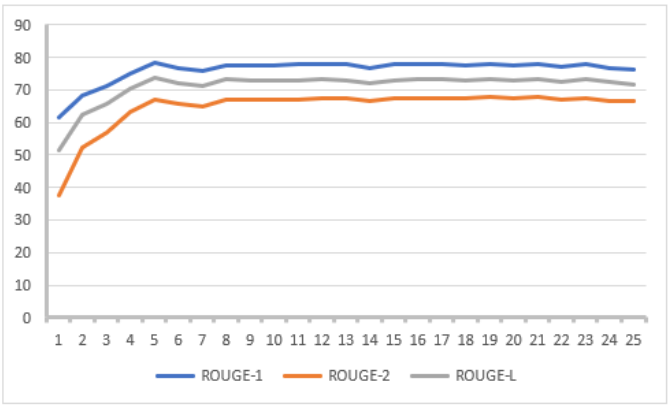

Fig. 7: Progression of the ROUGE-1, -2, -L scores for ProphetNet while training on our Generated Dataset

Model	ROUGE-1	ROUGE-2	ROUGE-L
ProphetNet	76.279	66.5158	71.7507
BertSum	16.7173	1.2581	15.6245

TABLE VIII: ROUGE-1, -2 and -L scores for the models on our Generated dataset

The BertSum model, as expected, was not able to learn on this data as the loss increased and worsened and hence these results have not been included here. The final rouge scores are shown in Table VIII. The predicted summaries showed that some sort of overfitting or collapse into the same repeated pattern for all examples. This would have to be further analyzed in future experiments to understand the cause.

Since the ProphetNet model showed promising results we further trained it on data augmented with the statistics from the stat column in the dataset along with the table names, all separated with a special separator token. The format of the

augmented data is: ⟨ Table ⟩ ⟨ SEP ⟩ ⟨ Query Type ⟩ ⟨ SEP ⟩ ⟨ Data ⟩ ⟨ SEP ⟩ ⟨ Stats ⟩.

The test and validation curves were similar to the unaugmented training, with the loss reaching its lowest point at around epoch 15. The final rouge scores are shown in Table IX and some samples are shown in Appendix **??**

ROUGE-1	ROUGE-2	ROUGE-L
78.6919	68.3018	73.7813

TABLE IX: Final ROUGE-1, -2 and -L scores for ProphetNet trained on the augmented data

While there appears to some of the template structure emerging in training with the limited amount of training in this scoped study, the results have inconsistencies in the numbers of statistics of the table. We believe this could be resolved with further longer training times and fine tuning of the "x" set of the generated dataset. These experiments show promising results on using Transformer based models for the Database Summarization task and could be further investigated in a larger study.

VI. Conclusion

In this paper we conduct an early scoped study on the use of Transformer-based models to summarize database query results in a way that can be vocalised and comprehended without visual aid or screen based systems. We also present a novel generated dataset for this task from the WikiSQL Dataset based on rules for the summarization task. Our focus was mainly on the table, row, column and cell summaries. Two different Transformer models were compared, a traditional Bert model adapted for summarization (BertSum) and multistream based model (ProphetNet). We conducted experiments first on a general summarization task and we compared these models on the Reddit TIFU Dataset. As shown in the results in Section V, the ProphetNet model performed better on the ROUGE metrics. We then trained this model on our generated dataset for database summarization. While the traditional BertSum model did not perform well and seemed to overfit on the data, the ProphetNet model showed promising results that show potential for further investigation and experiments. We would like to study specially introspection and interpretability applications over these models, to help us to tune them better. Future work can also be done to compare these results with existing database vocalization approaches such as CiceroDB and a user study can be conducted to see how they perform with human evaluators

VII. Acknowledgement

The paper on "Talk to me: Database Vocalization Summaries using BertSUM and Prophet-Net" and the research behind it would not have been possible without the exceptional support of our supervisor, M.Sc. Gabriel Campero Durand. His enthusiasm, knowledge and careful attention to detail have been an inspiration and kept our work on track from first encounter. We would like to thank each and every member of our team who contributed successfully for the completion of this project and offered insights into writing all of the interpretations/conclusions of this paper.

References

[1] Byeongchang Kim, Hyunwoo Kim and Gunhee Kim. Abstractive Summarization of Reddit Posts with Multi-level Memory Networks. In NAACL-HLT (oral), 2019.

[2] Victor Zhong, Caiming Xiong, Richard Socher: Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. CoRR abs/1709.00103 (2017)

[3] Trummer, I., Zhu, J. and Bryan, M., 2017. Data vocalization: optimizing voice output of relational data. Proceedings of the VLDB Endowment, 10(11), pp.1574-1585.

[4] Jo, S., Trummer, I., Yu, W., Wang, X., Yu, C., Liu, D. and Mehta, N., 2019, June. Verifying text summaries of relational data sets. In Proceedings of the 2019 International Conference on Management of Data (pp. 299-316).

[5] Liu, Y. and Lapata, M., 2019. Text summarization with pretrained encoders. arXiv preprint arXiv:1908.08345.

[6] Yan, Y., Qi, W., Gong, Y., Liu, D., Duan, N., Chen, J., Zhang, R. and Zhou, M., 2020. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. arXiv preprint arXiv:2001.04063.

[7] Trummer, I., 2020. Demonstrating the voice-based exploration of large data sets with CiceroDB-zero. Proceedings of the VLDB Endowment, 13(12), pp.2869-2872.

[8] Mehdi Allahyari and Seyedamin Pouriyeh and Mehdi Assefi and Saeid Safaei and Elizabeth D. Trippe and Juan B. Gutierrez and Krys Kochut, (2017) arXiv:1707.02268

[9] H. P. Luhn,"The Automatic Creation of Literature Abstracts," in IBM Journal of Research and Development, vol. 2, no. 2, pp. 159-165, Apr. 1958, doi: 10.1147/rd.22.0159.

[10] Nenkova, Ani Bagga, Amit. (2003). Facilitating email thread access by extractive summary generation. 287-296. 10.1075/cilt.260.32nen.

[11] Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saied Safaei, Elizabeth D. Trippe, Juan B. Gutierrez, Krys Kochut - A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques(2017) arXiv:1707.02919v2

[12] Gupta, Vishal Lehal, Gurpreet. (2010). A Survey of Text Summarization Extractive Techniques. Journal of Emerging Technologies in Web Intelligence. 2. 10.4304/jetwi.2.3.258-268.

[13] Lin, Chin-Yew. (2004). ROUGE: A Package for Automatic Evaluation of summaries. Proceedings of the ACL Workshop: Text Summarization Braches Out 2004. 10.

[14] Francia, Matteo, Enrico Gallinucci, and Matteo Golfarelli. "Towards Conversational OLAP." DOLAP. 2020.

APPENDIX

A - BertSum was highly overfitting on the dataset and producing invalidsummary sequence

Target	Prediction
got fed up with the neighbours gossiping, accidentally kicked her door in. police cam while I spoke to landlord. everything super awkdward. I am a jerk.	, a. my to i and ' the in of for got it was with
left keys in coat. had to trek to get brothers keys. bring them back to him. miss class and test. while freezing	, a. my to i and ' the in of for got it was with
there was a mixup in the interview time, but i don't know who made the mistake (me or employer). i don't think i'll get the job.	, a. my to i and ' the in of for got it was with

B - Predictions from the ProphetNet models on the Generated Dataset

Data	name : gareth abraham ; nationality : wales ; years : 1987 – 1993 ; appearances : 109. 0 ; goals : 5. 0 ; position : cd ; ; name : neil alexander ; nationality : scotland ; years : 2001 – 2007 ; appearances : 234. 0 ; goals : 0. 0 ; position : gk ; ; name : ivor allchurch ; nationality : wales ; years : 1962 – 1965 ; appearances : 112. 0 ; goals : 39. 0 ; position : if ; ; name : willie anderson ; nationality : england ; years : 1973 – 1977 ; appearances : 142. 0 ; goals : 12. 0 ; position : w	name : gareth abraham ; nationality : wales ; years : 1987 – 1993 ; appearances : 109. 0 ; goals : 5. 0 ; position : cd ; ; name : neil alexander ; nationality : scotland ; years : 2001 – 2007 ; appearances : 234. 0 ; goals : 0. 0 ; position : gk ; ; name : ivor allchurch ; nationality : wales ; years : 1962 – 1965 ; appearances : 112. 0 ; goals : 39. 0 ; position : if ; ; name : willie anderson ; nationality : england ; years : 1973 – 1977 ; appearances : 142. 0 ; goals : 12. 0 ; position : w
Target	table 2007 melbourne cup has 23 rows and column saddle cloth has entries with unique values ranging between 23. 0 and 1. 0 and column weight kg has entries with uniques value between 57 and 51 and column horse has entries with uniques value between tawqeet usa and princess coup	table list of cardiff city f c players has 167 rows and column name has entries with uniques value between gareth abraham and scott young and column nationality has entries with uniques value between wales and wales and column years has entries with uniques value between 1987 – 1993 and 1993 – 2004
Prediction	table 2007 melbourne cup has 16 rows and column saddle cloth has entries with unique values ranging between 3 . 0 and 1 . 0	table 1968 european european cup has 16 rows and column name has entries with uniques value between gareth abraham and the position of w and column nationality is wales and column years is 1987 – 1993 , and with a appearances of 109 . 0 , and the goals is 5 . 0

C - Predictions from the ProphetNet models trained on the augmented dataset

Target	Prediction
table list of tampa bay lightning draft picks has 204 rows and column draft has entries with unique values ranging between 2013. 0 and 1992. 0 and column nationality has entries with uniques value between czech republic and switzerland and column player has entries with uniques value between roman hamrlik category articles with hcards and joel vermin category articles with hcards	table list of tampa bay lightning draft picks has 20 rows and column draft has entries with unique values ranging between 1992 . 0 and 1992 . 1 and column nationality has entries reaching 5 . 0 percent probability of czech republic and column player has entries dating between roman hamrlik category : articles with hcards and 1 . 0 as maximum value 1 . 5 as median and 1 as minimum value
table indian national rally championship has 25 rows and column season has entries with unique values ranging between 2012. 0 and 1988. 0 and column governing body has entries with 64. 0 percent probability of fmsci 36. 0 percent probability of mai and column co driver has entries with uniques value between ashwin naik and raj bagri	table indian national rally championship has 12 rows and column season has entries with uniques value between 2012 . 0 and 2010 . 1 and column governing body has entries reaching 100 . 0 percent probability of fmsci and column co driver has entries dating between ashwin naik and mai
table 1979 world figure skating championships 2 has 31 rows and column rank has entries with unique values ranging between 31. 0 and 1. 0 and column nation has entries with uniques value between united states and spain and column name has entries with uniques value between linda fratianne and gloria mas	table 1979 world figure skating championships 2 has 8 rows and column rank has entries with unique values ranging between 5 . 0 and 1 . 0 before column nation has entries and column name has entries offering uniques value between linda fratianne and john biellmann