Vol. 08, No. 02, July 2024, pp. 11-21

INTERNATIONAL JOURNAL OF INNOVATION IN ENTERPRISE SYSTEM

Journal homepage: https://ijies.sie.telkomuniversity.ac.id
e-ISSN: 2580-3050, DOI: https://doi.org/10.25124/ijies.v8i02.236

Enhancing Table-to-Text Generation with Numerical Reasoning Using Graph2Seq Models

Sulisetyo Puji Widodo[1*], Adila Alfa Krisnadhi[2]

[1] Faculty of Computer Science
Universitas Indonesia, Depok, Indonesia
sulisetyo.puji@ui.ac.id

[2] Faculty of Computer Science
Universitas Indonesia, Depok, Indonesia
adila@cs.ui.ac.id

[*]sulisetyo.puji@ui.ac.id

ARTICLE INFO

Article history:
Received 14 June 2023
Accepted 24 July 2024
Published 31 July 2024

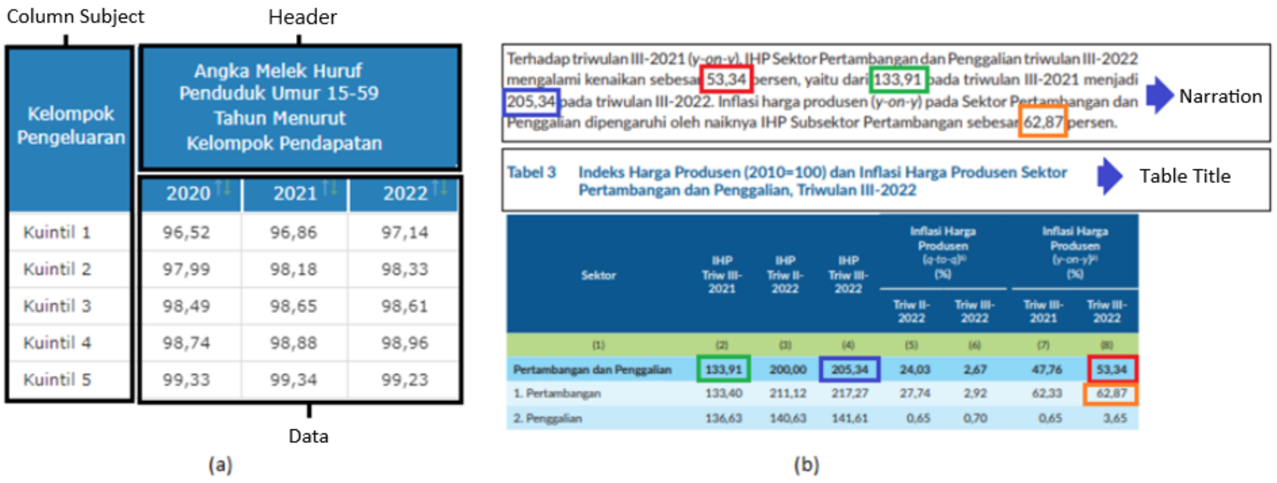
Keywords:
Table-to-Text Generation;
Graph; Graphsage-RNN;
GCN-RNN; Hallucination;

ABSTRACT IN ENGLISH

Interpreting data in tables into narratives is necessary because tables cannot explain their own data. Additionally, there is a need to produce more analytic narratives from the results of numerical reasoning on data from tables. The sequence-to-sequence (Seq2Seq) encoder-decoder structure is the most widely used in table-to-text generation (T2XG). However, Seq2Seq requires the linearization of tables, which can omit structural information and create hallucination problems. Alternatively, the graph-to-sequence (Graph2Seq) encoder-decoder structure utilizes a graph encoder to better capture important data information. Several studies have shown that Graph2Seq outperforms Seq2Seq. Thus, this study applies Graph2Seq to T2XG, leveraging the structured nature of tables that can be represented by graphs. This research initiates the use of Graph2Seq in T2XG with GCN-RNN and GraphSage-RNN, aiming to improve narrative generation from tables through enhanced numerical reasoning. Based on the automatic evaluation of the application of Graph2Seq on the T2XG task, it has the same performance as the baseline model. Meanwhile, in human evaluation, Graphsage-RNN is better able to reduce the possibility of hallucinations in text compared to the baseline model and GCN-RNN.

1. INTRODUCTION

Data volume is currently growing very rapidly and the most commonly used form of data presentation is tables. Tables can arrange data in a standard structure so that it is easier to find and compare information [1]. Nonetheless, it is important to interpret data in tables [2] because tables cannot explain the data themselves. Tables accompanied by text can help interpret the information in tables more easily. As an illustration, Figure 1(a) shows a table that only has column subjects, headers, data, and without a narrative explaining the table. Figure 1(b) shows a table equipped with a narrative explaining the data in the table. Figure 1: Illustration of (a) tables without narration and (b) tables with narration explaining the data in the table. Boxes with the same color show the association between narrative and data. Generally, text is made manually by humans by making direct observations on tables commonly found in scientific journals, creating NBA news, statistical reports, financial reports, medical reports, etc. Therefore, automation of text generation from tables is necessary due to the increasing number of tables. In this regard, Table-to-Text Generation (T2XG) can be applied, where T2XG aims to produce descriptive text that is factual or contextually aligned based on the data in the table [3].



Column Subject
Header
Data
(a)
Narration
Table Title
(b)

Kelompok Pengeluaran	Angka Melek Huruf Penduduk Umur 15-59 Tahun Menurut Kelompok Pendapatan		
	2020	2021	2022
Kuintil 1	96,52	96,86	97,14
Kuintil 2	97,99	98,18	98,33
Kuintil 3	98,49	98,65	98,61
Kuintil 4	98,74	98,88	98,96
Kuintil 5	99,33	99,34	99,23

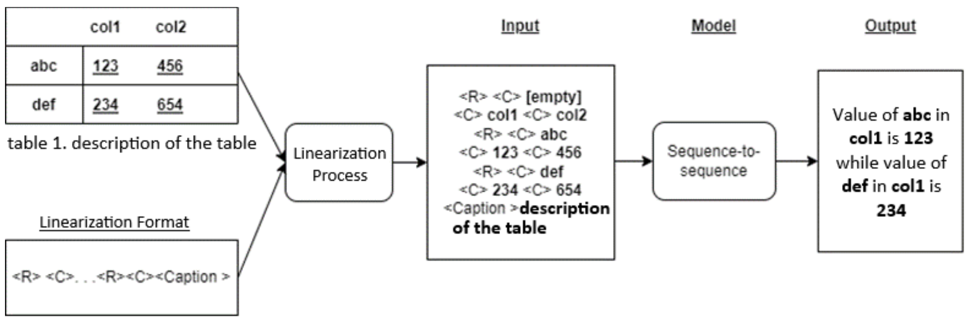
Terhadap triwulan III-2021 (y-on-y), IHP Sektor Pertambangan dan Penggalian triwulan III-2022 mengalami kenaikan sebesar 53,34 persen, yaitu dari 133,91 pada triwulan III-2021 menjadi 205,34 pada triwulan III-2022. Inflasi harga produsen (y-on-y) pada Sektor Pertambangan dan Penggalian dipengaruhi oleh naiknya IHP Subsektor Pertambangan sebesar 62,87 persen.

Tabel 3 Indeks Harga Produsen (2010=100) dan Inflasi Harga Produsen Sektor Pertambangan dan Penggalian, Triwulan III-2022

Sektor	IHP Triw III-2021	IHP Triw II-2022	IHP Triw III-2022	Inflasi Harga Produsen (q-to-q)[1] (%)		Inflasi Harga Produsen (y-on-y)[1] (%)	
				Triw II-2022	Triw III-2022	Triw III-2021	Triw III-2022
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Pertambangan dan Penggalian	133,91	200,00	205,34	24,03	2,67	47,76	53,34
1. Pertambangan	133,40	211,12	217,27	27,74	2,92	62,33	62,87
2. Penggalian	136,63	140,63	141,61	0,65	0,70	0,65	3,65

Figure 1: Illustration of (a) tables without narration and (b) tables with narration explaining the data in the table. Boxes of the same color indicate associations between the narration and the data in the table

In general, T2XG can be divided into two categories: pipe patterns and end-to-end models [4]. Pipeline patterning is a traditional method based on rules and templates that consists of three successive processes, namely content planning, sentence planning, and linguistic realization. In the pipeline pattern, facts and conclusions in the text can be interpreted and controlled more. However, the solution for this approach is done manually because it is necessary to select rules and templates that match the table format [5]. Meanwhile, the end-to-end model is a neural network model that relies more on data without manual intervention and uses more encoder-decoder structures, namely sequence-to-sequence (Seq2Seq) [6]. Seq2Seq uses input in the form of a sequence, so the table needs to be linearized following a certain format so that it can be processed in the model, as shown in Figure 2.



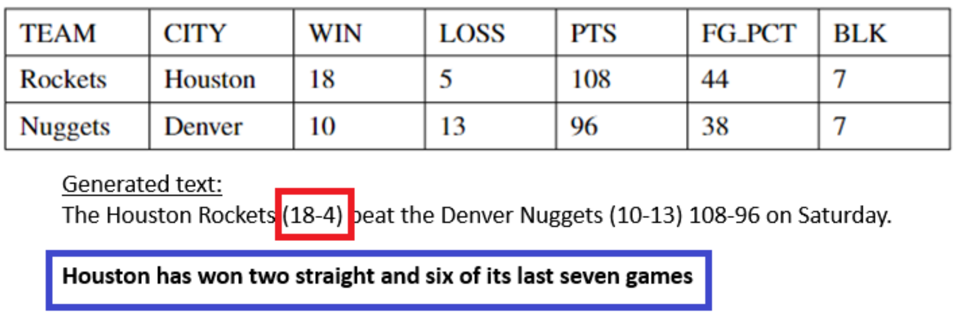
Input
Model
Output
table 1. description of the table
Linearization Format
<R> <C>. . .<R><C><Caption >
Linearization Process
<R> <C> [empty] <C> col1 <C> col2 <R> <C> abc <C> 123 <C> 456 <R> <C> def <C> 234 <C> 654 <Caption >description of the table
Sequence-to-sequence
Value of abc in col1 is 123 while value of def in col1 is 234

	col1	col2
abc	123	456
def	234	654

Figure 2 - Illustration of making table-to-text using sequence-to-sequence architecture

However, maintaining harmony between descriptive text and facts in tables is not enough because it is necessary to produce texts that are more analytical [7]. In general, analytical texts are made from the results of numerical reasoning as indicated by conclusions ("better", "decreased", and "increased by") and calculated data ("2.46 percent" and "5.58 percent") in the form table. Therefore, many researchers use the pre-trained language model (PLM), which is the Seq2Seq architecture, to generate narratives from tables with numerical reasoning. [8] trained BART [9] and T5 [10] using table linearization as input. Linearization follows the <R>, <C>, and <CAP> formats that represent the start of new rows, cells, and table headings. The evaluation results show that T5 is better than BART at T2XG. [11] calculates the maximum, minimum, and difference values for each column in the table. The information is then fed into a template adopted from [12] to retrain GPT2 [13] and T5. The results showed that T5 with an encoder-decoder architecture was better at producing text than GPT2, which was only a decoder model. [14] used the dataset and format used by [8] on BART and T5.

However, the Seq2Seq model often fails to function as expected because the table linearization process cannot maintain the data structure in the table, causing loss of structural information and hallucination problems [15], [16]. Hallucinations are an anomaly that can cause the text to have conflicting data from the table and the text can also have conclusions that are not in the table. Of course, these hallucinations can lead the resulting text to wrong conclusions and understanding of the tables. As an illustration, Figure 3. demonstrated two types of hallucinations on the T2XG. Text marked with a red box is a fact that contradicts the data in the table, while text in a blue box is text whose information is irrelevant or not mentioned in the table.



TEAM	CITY	WIN	LOSS	PTS	FG_PCT	BLK
Rockets	Houston	18	5	108	44	7
Nuggets	Denver	10	13	96	38	7

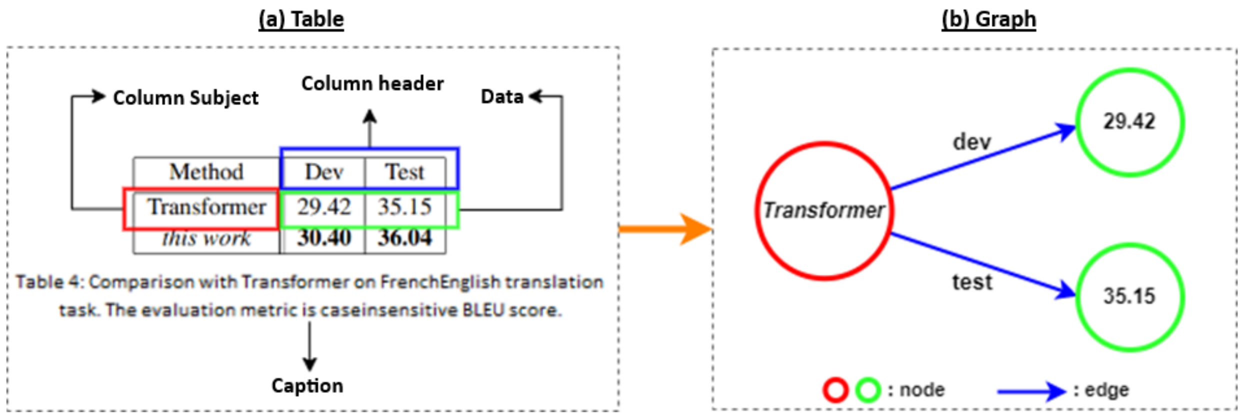
Generated text:
The Houston Rockets (18-4) beat the Denver Nuggets (10-13) 108-96 on Saturday.

Houston has won two straight and six of its last seven games

Figure 3 - Examples of text in the table containing hallucinations are adapted from [17]. The correct fact in the red box is 18-5 and the blue box is a fact that is irrelevant to the contents of the table.

In addition to Seq2Seq, there is graph-to-sequence (Graph2Seq) which uses a graph encoder and a sequence decoder. The graph encoder accepts the graph as input and represents the structural information of the graph (nodes, neighbors and edges) into a vector space [18]. Then the vector is sent to the sequence decoder to produce text. The graph encoder has a better ability to capture important information in the data. It has been seen in several studies that Graph2Seq is able to outperform Seq2Seq, among others in the tasks of generating comments [19], rendering Graphs to text and syntactic-based neural machine translation [20], [21], [22]. Thus, compared to Seq2Seq, Graph2Seq has advantages in preserving data structure, capturing important information from tables, and reducing the problem of hallucinations. However, until now, there has been no research that applies Graph2Seq to T2XG because the input required by Graph2Seq is a graph, not a table. Meanwhile, tables are basically structured information that can be represented as graphs [2], [3], [11], [14], [15]. So Graph2Seq can be a potential solution for generating more accurate text, especially by capturing important information from tables, thus helping to reduce the occurrence of hallucinations. For this reason, there is a need for a table to graph conversion mechanism so that Graph2Seq can be applied to TX2G.

One approach that can be applied in representing tables in graphical form is the Relational Database to RDF Mapping Language (R2RML) approach. In representing data in tables, R2RML uses the perspective that in one subject, the column acts as a node, the column header as an edge, and the data as a node. Based on this mapping, the graph for the first row can be seen in Figure 4(b).



Method	Dev	Test
Transformer	29.42	35.15
this work	**30.40**	**36.04**

Table 4: Comparison with Transformer on FrenchEnglish translation task. The evaluation metric is caseinsensitive BLEU score.

Figure 4 - Illustration of table representation into a graph using R2RML perspective.

This study intends to apply Graph2Seq to solving T2XG problems. The idea is to first convert the table structure into a graph, which is then embedded into the vector space using a graph encoder such as GNN. The vector is then passed to a sequence-based decoder such as an RNN, to generate text. This study began with GNN variants, namely Graph Convolutional Network (GCN) [23] and Graph-SAGE [24] as graph encoders, respectively each is paired with an RNN as a sequence decoder. GCN is the most popular GNN variant used, while Graph-SAGE is computationally efficient because it uses a sampling method on its nodes.
In this paper, the following research questions have been proposed:

- How do Graph2Seq models compare to Seq2Seq models in terms of accuracy and coherence in table-to-text generation?
- What are the specific advantages of using the GCN-RNN and GraphSage-RNN models for T2XG?
- Can Graph2Seq models effectively handle numerical reasoning tasks in the context of T2XG?

The aim of this research is to overcome the limitations of the Seq2Seq model in table-to-text generation (T2XG) with numerical reasoning, namely loss of data structure in tables and hallucinations. Additionally, to explore the potential of Graph2Seq models (especially GCN-RNN and GraphSage-RNN) in generating accurate and coherent narratives from tabular data. As a research contribution, we developed two T2XG models based on the Graph2Seq architecture, namely the T2XG model with GCN-RNN and Graphsage-RNN.

The primary benefit of this research is the potential to produce more accurate and coherent narratives from tables. This improvement has wide-ranging implications for various fields, including:

- Data journalism: Automated generation of news articles from statistical data.
- Financial Analysis: Enhanced reporting from financial tables and spreadsheets.
- Scientific Communication: Improved generation of summaries and reports from experimental data tables.

The limitations of this study use a dataset which is a collection of numerical tables from scientific journals with descriptions in English. The dataset is taken from a study by [11] https://github.com/titech-nlp/numeric-nlg. The table types used are Single-cell-subject table and Composed-subject table and have header information. The input used is only tables and table captions, without involving context outside the table. The type of GNN used is a variant of GNN, namely Graph Convolutional Networks (GCN) [23] and Graph-SAGE [24]. The sequence decoder used is an RNN with an attention mechanism [25].

2. METHOD

2.1. Dataset and Baseline

This study uses research from [11], 2021 as a baseline, which in its research uses the Seq2Seq-based pretrained T5 model, which is equipped with a copy mechanism to generate text from tables. This model has the smallest hallucination score among the models offered, namely 50%. Meanwhile, this study uses the numericNLG dataset [11], which is a collection of numerical tables from English-language scientific journals with descriptions produced by experts with rich inferences. Dataset details can be seen in Table 1.

Table 1 – Details of numericNLG dataset

Dataset	Number of Tables	Train	Dev	Test	Vocabulary	Domain	Format
numericNLG	1.3 K	1.084	136	135	19.6 K	Scientific	Json

2.2. Proposed method

The method in this study begins by making a graph of the input table, which consists of four processes. (1) Preprocessing, which aims to clean and normalize the target text from symbols, HTML tags, and accented characters. (2) Detect and determine the subject in the table. (3) Enrich the table by adding information from the results of numerical reasoning. (4) Build graphs based on tables that have been enriched with numerical reasoning. The graph will then be used as input to the encoder-decoder model, GCN-RNN or Graphsage-RNN, to generate text. Overall, this research method is shown in Figure 5.

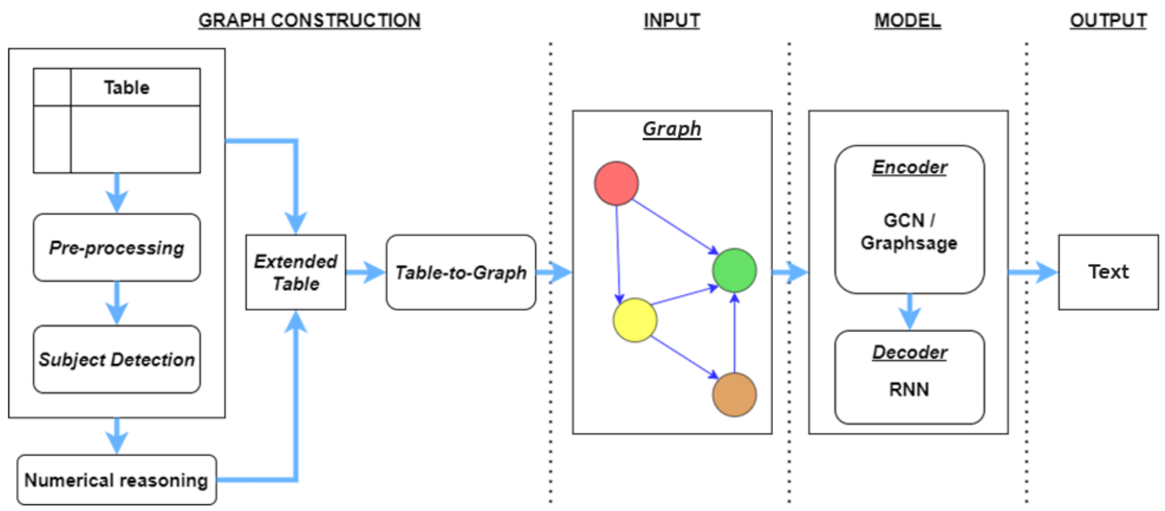


Figure 5 - Proposed method

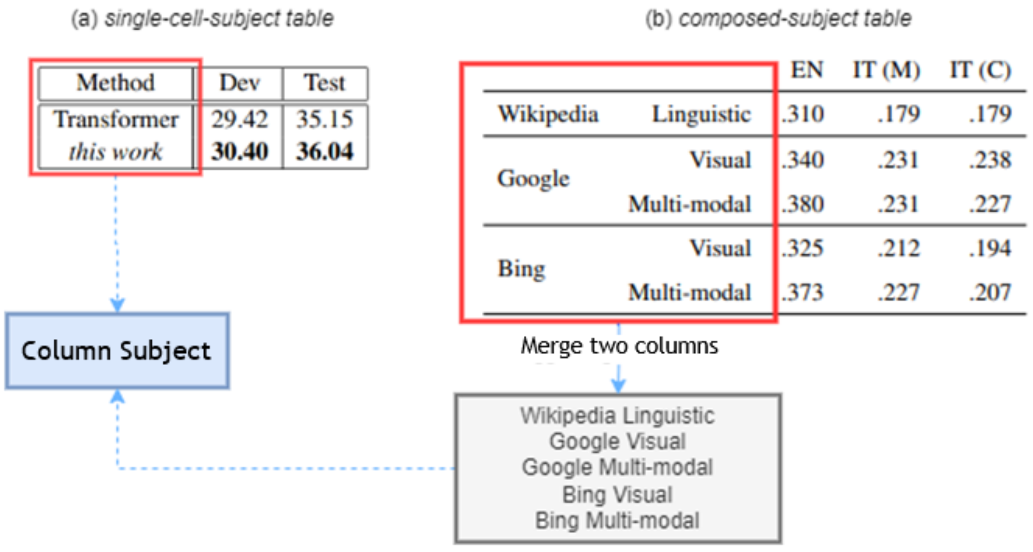
2.3. Graph construction

2.3.1. Preprocessing

The datasets will be cleaned and normalized to target text symbols, HTML tags, and accented characters.

2.3.2. Subject detection

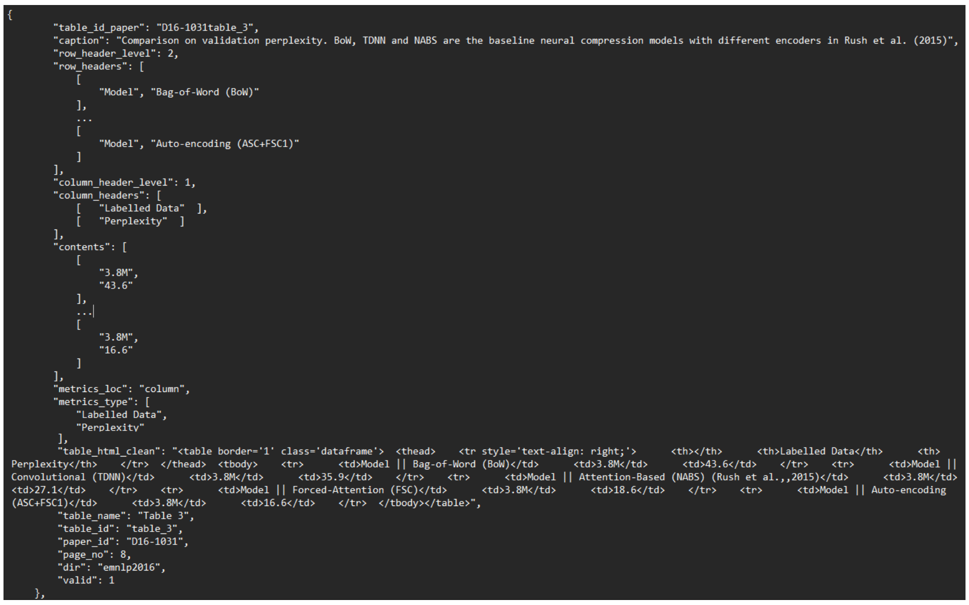
After preprocessing, the next step is to detect and determine the subject of the table. The types of tables used in this study are Single-cell-subject tables and Composed-subject tables. In a single-cell subject table, each row and corresponding column describe only one subject, and all subject labels are in a single column. While the subject-compiled table is a table that requires a combination of several cells to form the subject of each row. As an illustration, Figure 6 shows that (a) is a single-cell subject table so that the values in the first column, namely "transformer" and "this job" become the subject of the table. (b) Tables with categories consist of subject tables that have two subject columns so that the values in the two columns need to be combined into a subject table. For example, "Wikipedia" and "Linguistics" in the first line would need to be concatenated into "Wikipedia Linguistics" to become the subject of the table. The process of merging these two values is carried out until the last row. In the numericNLG dataset, there is information in the "row header" which consists of n values that form the subject of the table. If n > 1, then the subject of the table is the result of combining the values in the "header row".



Method	Dev	Test
Transformer	29.42	35.15
this work	**30.40**	**36.04**

		EN	IT (M)	IT (C)
Wikipedia	Linguistic	.310	.179	.179
Google	Visual	.340	.231	.238
	Multi-modal	.380	.231	.227
Bing	Visual	.325	.212	.194
	Multi-modal	.373	.227	.207

Figure 6 - Determine the subject table from (a) a single-cell subject table and (b) an array of subject tables.



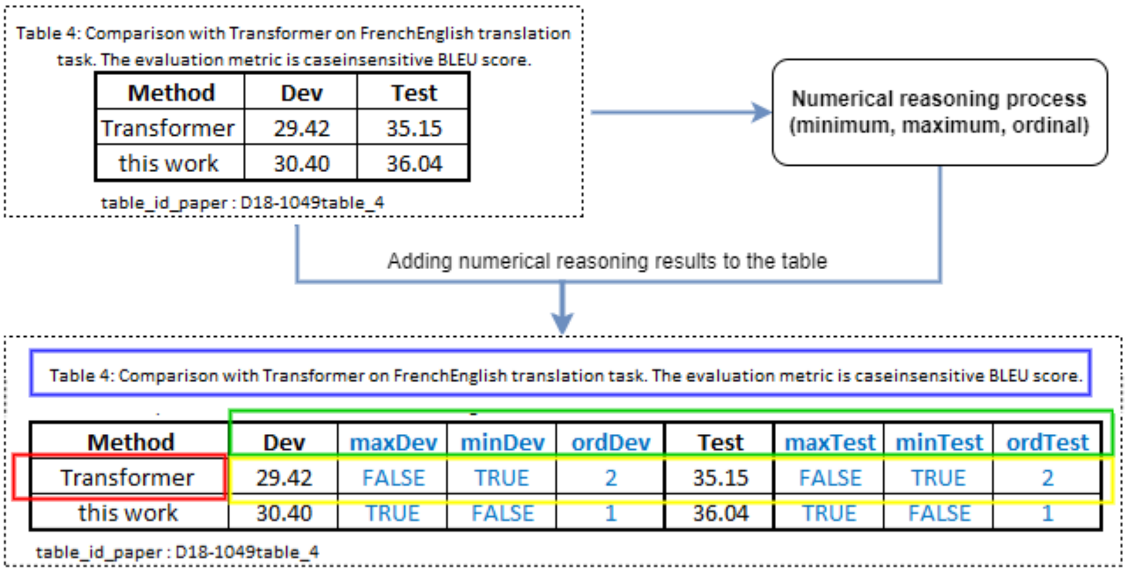
```
{
        "table_id_paper": "D16-1031table_3",
        "caption": "Comparison on validation perplexity. BoW, TDNN and NABS are the baseline neural compression models with different encoders in Rush et al. (2015)",
        "row_header_level": 2,
        "row_headers": [
            [
                "Model", "Bag-of-Word (BoW)"
            ],
            ...
            [
                "Model", "Auto-encoding (ASC+FSC1)"
            ]
        ],
        "column_header_level": 1,
        "column_headers": [
            [   "Labelled Data"  ],
            [   "Perplexity"  ]
        ],
        "contents": [
            [
                "3.8M",
                "43.6"
            ],
            ...
            [
                "3.8M",
                "16.6"
            ]
        ],
        "metrics_loc": "column",
        "metrics_type": [
            "Labelled Data",
            "Perplexity"
         ],
         "table_html_clean": "<table border='1' class='dataframe'>  <thead>    <tr style='text-align: right;'>      <th></th>      <th>Labelled Data</th>      <th>
Perplexity</th>    </tr>  </thead>  <tbody>    <tr>      <td>Model || Bag-of-Word (BoW)</td>      <td>3.8M</td>      <td>43.6</td>    </tr>    <tr>      <td>Model ||
Convolutional (TDNN)</td>      <td>3.8M</td>      <td>35.9</td>    </tr>    <tr>      <td>Model || Attention-Based (NABS) (Rush et al.,,2015)</td>      <td>3.8M</td>
<td>27.1</td>    </tr>    <tr>      <td>Model || Forced-Attention (FSC)</td>      <td>3.8M</td>      <td>18.6</td>    </tr>    <tr>      <td>Model || Auto-encoding
(ASC+FSC1)</td>      <td>3.8M</td>      <td>16.6</td>    </tr>  </tbody></table>",
         "table_name": "Table 3",
         "table_id": "table_3",
         "paper_id": "D16-1031",
         "page_no": 8,
         "dir": "emnlp2016",
         "valid": 1
     },
```

Figure 7 - Example of numericNLG dataset in json format.

2.3.3. Numerical reasoning

After detecting and determining the subject in the table, the next step is to perform numerical reasoning on the table, which aims to enrich the information from the table. Numerical reasoning is performed for each column and the result is added as a new column in the table. Tables that have been enriched with numerical reasoning are then referred to as extended tables. As seen in Figure 8, the blue text is the result of numerical reasoning added to the table as a new column. Referring to [26], [27], [7], [28], this study uses numerical reasoning that is commonly used, namely

- Minimum: Numerical reasoning to get the subject with the lowest score for column k. The subject with the lowest score in column k is given a TRUE score, while the other subjects are given a FALSE score. Reasoning results are given column names in the format "min + column name k.".
- Maximum: Numerical reasoning to find the subject with the highest score for column k. The subject with the highest score in column k was given a TRUE score while the other subjects were given a FALSE score. Reasoning results are given column names in the format "max + column name k".
- Ordinal: Numerical reasoning to get the order of values from the largest to the smallest for column k. The subject with the highest score in column k is given a value of 1 while the subject with the second highest score is given a value of 2 and so on until the subject with the lowest score. Reasoning results are given column names in the format "ord + column name k".



Numerical reasoning process (minimum, maximum, ordinal)
Adding numerical reasoning results to the table

Table 4: Comparison with Transformer on FrenchEnglish translation task. The evaluation metric is caseinsensitive BLEU score.

Method	Dev	Test
Transformer	29.42	35.15
this work	30.40	36.04

table_id_paper : D18-1049table_4

Table 4: Comparison with Transformer on FrenchEnglish translation task. The evaluation metric is caseinsensitive BLEU score.

Method	Dev	maxDev	minDev	ordDev	Test	maxTest	minTest	ordTest
Transformer	29.42	FALSE	TRUE	2	35.15	FALSE	TRUE	2
this work	30.40	TRUE	FALSE	1	36.04	TRUE	FALSE	1

table_id_paper : D18-1049table_4

Figure 8 - An extended table example of a table adopted from [28] which has gone through numerical reasoning. The blue column is an additional column for the results of numerical reasoning.

2.3.4. Table-to-graph

The next step is to build the graph using the R2RML perspective on the extended table. Each row in one table column will be mapped as node-edge-node. Each subject table that has been determined in the subject detection process will be mapped into a node in the graph. Then every column header, including the reasoned column headers, in the table will be edgemapped. Furthermore, each value from the column will be mapped into nodes, including the values resulting from numerical reasoning. But for numerical reasoning "maximum value" and "minimum value" only actual values are included in the graph. This is done so that the graphs produced are more efficient but still informative.

Based on Figure 8, the table with red boxes, namely "Transformer" and "this work" is the subject of the table to be mapped as a node. Tables marked with green boxes i.e. "maxdev", "mindev", "orddev" and so on, will be mapped as edges, whereas any values given in the yellow boxes will be mapped as nodes. In addition, the information in the table is also added to the graph as additional information. Each token in the description in the table, as given in the blue box, will be mapped as a connected node from the first token to the last token with an edge. The illustration is a table that is expanded in Figure 8 to become a graph that can be seen in Figure 9.

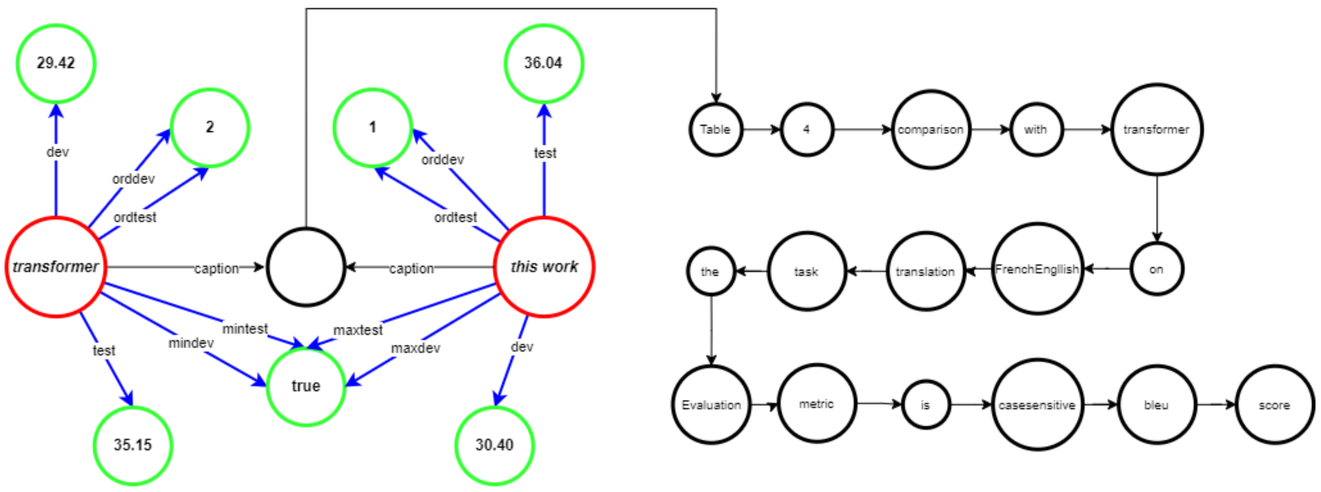


Figure 9 - Graphic illustration formed by the R2RML approach from the extended table (Figure 8).

2.4. Model

The architectural model used in this research is Graph2Seq which consists of a graph encoder and a sequence decoder. The graph encoder will receive the graph that was created in the previous process and convert it into a vector which is then sent to the sequence decoder to produce text. This study uses GCN and Graphsage as graph encoders, each of which is paired with an RNN as a sequence decoder. Each GCN-RNN and Graphsage-RNN will be evaluated to see their performance in generating text from tables.

2.5. Evaluation

In this study, the proposed method will be evaluated using automated and human evaluation. In order to see the performance of the proposed method, the evaluation results were then compared with research from [11]. The automatic evaluations used are BLEU, METEOR, ROUGE, and BertScore. The four measurements focus more on how well and correctly the data in the table appears in the resulting text.

In addition, this study also uses human evaluation to measure the accuracy of facts and numerical reasoning in the resulting text. Inspired by [11], the evaluation in this study was carried out by three evaluators who each got 20 tables and result texts that were randomly selected from the test data. The factual accuracy and numerical reasoning of a text are calcu- lated separately by measuring how many facts or numerical reasoning agree and how many contradict. The questionnaire is designed to be able to carry out this evaluation.

3. RESULT AND DISCUSSION

Table 2 - Evaluation results using automatic evaluation

Dataset	Input	Model	Bleu ↑	Rouge-L ↑	Meteor ↑	BertS ↑
numericNLG	Linearization	T5+Copy (Baseline)	**5.45**	28.15	19.16	**86.54**
	Graf	GCN-RNN	4.50	28.39	**22.68**	86.15
		Graphsage-RNN	4.73	**28.63**	21.45	86.22

Table 2 shows the average evaluation results from experiments on the two proposed models. In the numericNLG dataset, the Graphsage-RNN model has the highest Bleu value, namely 4.73, which means 0.72 smaller than the baseline. The two proposed models also have relatively similar numbers. This shows that the proposed model is not yet able to produce text that is close to the reference text. Nevertheless both proposed models show improvement on the evaluation of Rouge-L and Meteor. The highest Rouge-L value was obtained on GraphsageRNN, namely 28.63 with a difference of 0.48 from the baseline. Improvement was also shown by GCN-RNN although with a relatively small difference. However, it can be said that the two proposed models have better capabilities than the baseline in the Rouge-L evaluation. Meanwhile, in the Meteor evaluation, the two proposed models showed quite good improvements as seen in GCN-RNN which obtained the highest score, namely 22.68 or 3.52 better than the baseline. Meanwhile, in the Bertscore metric, Graphsage-RNN obtained a score of 86.22, which means it has a difference of 0.32 below the baseline. Similar values were also shown by the other two models, indicating that the three proposed models have the same capabilities as the baseline model based on Bertscore.

Table 3 - The evaluation results use automatic evaluation based on table type.

Dataset	Model	Type	Bleu ↑	Rouge-L ↑	Meteor ↑	BertS ↑
numericNLG	GCN-RNN	Single	4.14	28.01	**22.92**	**86.21**
		Composed	4.57	28.02	22.20	86.14
	Graphsage-RNN	Single	3.92	**28.49**	22.77	86.14
		Composed	**4.86**	28.40	21.78	86.15

Table 3 provides a detailed evaluation of the two models based on their table types: single-cell-subject tables and composed-subject tables. The test data for numericNLG includes 23 single-cell-subject tables and 112 composed-subject tables. Overall, both models perform better on single-cell-subject tables, although the differences in results are minimal. Notably, GCN-RNN demonstrates superior performance compared to Graphsage-RNN in terms of Meteor and BertS scores.

Table 4 - The results of the evaluation use human evaluation.

Model	Descriptive Facts			Inffered Facts		
	#Supp	#Cont	%Cont ↓	#Supp	#Cont	%Cont ↓
T5+Copy (Baseline)	0.04	0.04	50.00	0.78	0.57	42.62
GCN-RNN	0.06	0.06	50.00	0.74	0.76	50.67
Graphsage-RNN	0.06	0.05	**45.45**	0.82	0.59	**41.54**

Table 4 shows the results of human evaluation to show which model is better at reducing the appearance of intrinsic hallucinations in the generated text. The evaluation results show a comparison of the two proposed models, namely GCN-RNN and Graphsage-RNN on the numericNLG dataset. In numericNLG, the highest percentage of fact errors was obtained by GCN-RNN, namely 50.00%. This value shows that the model produces text with the appearance of incorrect facts amounting to 50.00% of the total facts that appear in the resulting text. At the same time, this value is the same as that obtained by the baseline model. However, what is different is the number of fact occurrences between GCN-RNN and the baseline. GCN-RNN brings up more facts from the table, namely 0.06. Next is GraphsageRNN which has the smallest fact error value compared to other models, namely 45.45%. This value shows that Graphsage-RNN can reduce the occurrence of factual errors.

Then in terms of reducing extrinsic hallucinations in numericNLG, GraphsageRNN also has better results than GCN-RNN and baseline. Graphsage-RNN obtained a percentage of 41.54%. This value shows that Graphsage-RNN is better at reducing hallucinations than GCN-RNN in terms of factual accuracy from the source table. Even though Graphsage-RNN has almost the same number of reasoning errors as the baseline, Graphsage-RNN is more capable of producing more precise conclusions than the baseline model.

Table 4 - Sample text generated from the proposed model on the numericNLG datasets.

Model	Generated Text
	numericNLG
GCN-RNN	table 3 shows the results of our model on the satisfaction comparison with previous work on the satisfaction comparison with previous work . table 2 shows the results of our model on the test set. the results are shown in table 2 and table 2. we can see that our proposed model outperforms all the baseline models in terms of bleu scores.
Graphsage-RNN	Table 3 shows the results of our model on the single-label test set. the first block shows the results of our model and the baselines on the test set. we can see that our proposed model achieves the best results in terms of accuracy and f1 score in terms of accuracy and f1 score on the test set.

4. CONCLUSION

The research proposes a Graph2Seq-based model in solving T2XG tasks with numerical reasoning. This research uses the numericNLG dataset, which is a dataset consisting of tables and descriptions from scientific papers. We conducted experiments using two pairs of graph encoders and sequence decoders, namely GCN-RNN and Graphsage-RNN. The experiment was evaluated using automatic evaluation, namely Bleu, Rouge-L, Meteor, and Bertscore to measure the closeness of the resulting text to the reference text from the dataset. Apart from that, human evaluation was also carried out to measure the possibility of hallucinations from the resulting text.

Based on the results of the automatic evaluation, experiments conducted on the numericNLG dataset show that the two proposed models, namely GCN-RNN and Graphsage-RNN, have capabilities that are close to those of the baseline model, namely T5. This can be seen from the Bleu and Bertscore of the proposed model which is not much different from the baseline and the Rouge-L and meteor values are better than the baseline. So it can be concluded that based on the automatic evaluation the application of Graph2Seq to the T2XG task has the same performance as the baseline model.

Meanwhile, in human evaluation, the proposed model, namely Graphsage-RNN, is more able to reduce the possibility of hallucinations appearing in text compared to the baseline model and GCN-RNN on the numericNLG dataset. Even though the evaluation results show a slight difference, this means that the implementation of Graph2Seq in T2XG, especially Graphsage-RNN, has good performance because it has almost the same results as PLMs in generating text and is better at reducing hallucinations in text. However, the problem of hallucinations is still an issue that can be explored more deeply. This is because the model that has been trained is still susceptible to confusion in maintaining the accuracy of the resulting text. Apart from that, the variations in the resulting text are still an interesting issue to work on in the future.

Apart from that, Graph2Seq has the potential to have better performance in the future by using a better decoder such as a transformer or combining it with a pretrained model such as T5 or by using a larger number of datatrains. Another thing that might be done in the future is to combine it with a pipeline-pattern which has the ability to control hallucinations better.

REFERENCES

[1] Y. Li, Z. Huang, J. Yan, Y. Zhou, F. Ye, and X. Liu, "GFTE: graph-based financial table extraction," in Pattern Recognition. ICPR International Workshops and Challenges - Virtual Event, January 10-15, 2021, Proceedings, Part II, ser. Lecture Notes in Computer Science, A. D. Bimbo, R. Cucchiara, S. Sclaroff, G. M. Farinella, T. Mei, M. Bertini, H. J. Escalante, and R. Vezzani, Eds., vol. 12662. Springer, 2020, pp. 644–658. [Online]. Available: https://doi.org/10.1007/978-3- 030-68790-8 50

[2] A. Liu, H. Dong, N. Okazaki, S. Han, and D. Zhang, "PLOG: table-to-logic pretraining for logical table-to-text generation," CoRR, vol. abs/2205.12697, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2205.12697

[3] T. Liu, K. Wang, L. Sha, B. Chang, and Z. Sui, "Table-to-text generation by structure-aware seq2seq learning," in Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, S. A. McIlraith and K. Q. Weinberger, Eds. AAAI Press, 2018, pp. 4881–4888. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16599

[4] H. Gong, X. Feng, B. Qin, and T. Liu, "Table-to-text generation with effective hierarchical encoder on three dimensions (row, column and time)," in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, K. Inui, J. Jiang, V. Ng, and

X. Wan, Eds. Association for Computational Linguistics, 2019, pp. 3141–3150. [Online]. Available: https://doi.org/10.18653/v1/D19-1310

[5] N. Jiang, J. Chen, R. Zhou, C. Wu, H. Chen, J. Zheng, and T. Wan, "PAN: pipeline assisted neural networks model for data-to-text generation in social internet of things," Inf. Sci., vol. 530, pp. 167–179, 2020. [Online]. Available: https://doi.org/10.1016/j.ins.2020.03.080

[6] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., 2014, pp. 3104–3112. [Online]. Available: https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2- Abstract.html

[7] W. Chen, J. Chen, Y. Su, Z. Chen, and W. Y. Wang, "Logical natural language generation from open-domain tables," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds. Association for Computational Linguistics, 2020, pp. 7929–7942. [Online]. Available: https://doi.org/10.18653/v1/2020.acl-main.708

[8] N. S. Moosavi, A. R ̈uckl ́e, D. Roth, and I. Gurevych, "Learning to reason for text generation from scientific tables," CoRR, vol. abs/2104.08296, 2021. [Online]. Available: https://arxiv.org/abs/2104.08296

[9] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds. Association for Computational Linguistics, 2020, pp. 7871–7880. [Online]. Available: https://doi.org/10.18653/v1/2020.acl-main.703

[10] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," CoRR, vol. abs/1910.10683, 2019. [Online]. Available: http://arxiv.org/abs/1910.10683

[11] L. H. Suadaa, H. Kamigaito, K. Funakoshi, M. Okumura, and H. Takamura, "Towards table-to-text generation with numerical reasoning," in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021, C. Zong, F. Xia, W. Li, and R. Navigli, Eds. Association for Computational Linguistics, 2021, pp. 1451–1465. [Online]. Available: https://doi.org/10.18653/v1/2021.acl-long.115

[12] M. Kale and A. Rastogi, "Template guided text generation for task-oriented dialogue," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, B. Webber, T. Cohn, Y. He, and Y. Liu, Eds. Association for Computational Linguistics, 2020, pp. 6505–6520. [Online]. Available: https://doi.org/10.18653/v1/2020.emnlp-main.527

[13] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners." 2019. [Online]. Available: https://cdn.openai.com/better-language- models/languagemodelsareunsupervisedmultitasklearners.pdf

[14] D. Petrak, N. S. Moosavi, and I. Gurevych, "Improving the numerical reasoning skills of pretrained language models," CoRR, vol. abs/2205.06733, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2205.06733

[15] J. Li, T. Tang, W. X. Zhao, and J. Wen, "Pretrained language models for text generation: A survey," CoRR, vol. abs/2105.10311, 2021. [Online]. Available: https://arxiv.org/abs/2105.10311

[16] C. Zhao, M. A. Walker, and S. Chaturvedi, "Bridging the structural gap between encoding and decoding for data-to-text generation," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds. Association for Computational Linguistics, 2020, pp. 2481–2491. [Online]. Available: https://doi.org/10.18653/v1/2020.acl- main.224

[17] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," CoRR, vol. abs/2202.03629, 2022. [Online]. Available: https://arxiv.org/abs/2202.03629

[18] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," IEEE Trans. Neural Networks, vol. 20, no. 1, pp. 61–80, 2009. [Online]. Available: https://doi.org/10.1109/TNN.2008.2005605

[19] W. Li, J. Xu, Y. He, S. Yan, Y. Wu, and X. Sun, "Coherent comments generation for chinese articles with a graph-to-sequence model," in Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, A. Korhonen, D. R. Traum, and L. M`arquez, Eds. Association for Computational Linguistics, 2019, pp. 4843–4852. [Online]. Available: https://doi.org/10.18653/v1/p19-1479

[20] D. Beck, G. Haffari, and T. Cohn, "Graph-to-sequence learning using gated graph neural networks," in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers, I. Gurevych and Y. Miyao, Eds. Association for Computational Linguistics, 2018, pp. 273–283. [Online]. Available: https://aclanthology.org/P18-1026/

[21] D. Marcheggiani and L. Perez-Beltrachini, "Deep graph convolutional encoders for structured data to text generation," in Proceedings of the 11th International Conference on Natural Language Generation, Tilburg University, The Netherlands, November 5-8, 2018, E. Krahmer, A. Gatt, and M. Goudbeek, Eds. Association for Computational Linguistics, 2018, pp. 1–9. [Online]. Available: https://doi.org/10.18653/v1/w18-6501

[22] Z. Guo, Y. Zhang, Z. Teng, and W. Lu, "Densely connected graph convolutional networks for graph-to-sequence learning," Trans. Assoc. Comput. Linguistics, vol. 7, pp. 297–312, 2019. [Online]. Available: https://doi.org/10.1162/tacl a 00269

[23] Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel, "Gated graph sequence neural networks," in 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://arxiv.org/abs/1511.05493

[24] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," CoRR, vol. abs/1706.02216, 2017. [Online]. Available: http://arxiv.org/abs/1706.02216

[25] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.0473

[26] W. Chen, H. Wang, J. Chen, Y. Zhang, H. Wang, S. Li, X. Zhou, and W. Y. Wang, "Tabfact: A large-scale dataset for table-based fact verification," 2019. [Online]. Available: https://arxiv.org/abs/1909.02164

[27] M. Geva, A. Gupta, and J. Berant, "Injecting numerical reasoning skills into language models," in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, Eds. Association for Computational Linguistics, 2020, pp. 946–958. [Online]. Available: https://doi.org/10.18653/v1/2020.acl-main.89

[28] Z. Chen, W. Chen, C. Smiley, S. Shah, I. Borova, D. Langdon, R. Moussa, M. Beane, T. Huang, B. R. Routledge, and W. Y. Wang, "Finqa: A dataset of numerical reasoning over financial data," in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, M. Moens, X. Huang, L. Specia, and S. W. Yih, Eds. Association for Computational Linguistics, 2021, pp. 3697–3711. [Online]. Available: https://doi.org/10.18653/v1/2021.emnlp-main.300

[29] J. Zhang, H. Luan, M. Sun, F. Zhai, J. Xu, M. Zhang, and Y. Liu, "Improving the transformer translation model with document-level context," in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4,

2018, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds. Association for Computational Linguistics, 2018, pp. 533–542. [Online]. Available: https://doi.org/10.18653/v1/d18-1049