Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 6147–6152
Marseille, 20-25 June 2022

FinMath: Injecting a Tree-structured Solver for Question Answering over Financial Reports

Chenying Li†**, Wenbo Ye**‡**, Yilun Zhao***
†Northeastern University
‡Zhejiang University
*Yale University
†li.chenyin@northeastern.edu
‡wenbo.17@intl.zju.edu.cn

Abstract

Answering questions over financial reports containing both tabular and textual data (hybrid data) is challenging as it requires models to select information from financial reports and perform complex quantitative analyses. Although current models have demonstrated a solid capability to solve simple questions, they struggle with complex questions that require a multiple-step numerical reasoning process. This paper proposes a new framework named FinMath, which improves the model's numerical reasoning capacity by injecting a tree-structured neural model to perform multi-step numerical reasoning processes. Specifically, in the first phase, FinMath extracts supporting evidence from the financial reports given the question. And in the second phase, a tree-structured neural model is applied to generate a tree expression in a top-down recursive way. Experiments on the TAT-QA dataset show that FinMath improves the previous best result by 8.5% absolute for Exact Match (EM) score (50.1% to 58.6%) and 6.1% absolute for numeracy-focused F_1 score (58.0% to 64.1%).

Keywords: Financial NLP, Question Answering, Numerical Reasoning, Math Word Problems Solving

1. Introduction

The sheer volume of financial statements and tables makes it difficult and time-consuming for humans to access and analyze financial reports. Robust numerical reasoning over hybrid data combining both tabular and textual content faces unique challenges in this domain. TAT-QA dataset (Zhu et al., 2021) focuses on questions that require numerical reasoning over financial report pages containing both paragraphs and tables. As the example shown in Figure 1, the question "What was the percentage change in gaming between 2018 and 2019?" requires the QA system to analyze the given paragraphs and tables, locate relevant cells in the tabular content and then perform subtraction and division operations to get the final answer.

However, current best model over TAT-QA dataset, named TAGOP (Zhu et al., 2021), can only perform symbolic reasoning with a single type of pre-defined aggregation operators (e.g. change ratio, division), and might fail to answer complex questions requiring multi-step reasoning. To address these shortcomings, we present a new framework called FinMath, which can perform arbitrary steps of numerical reasoning given the arithmetic questions. Motivated by the recent works in the task of Math Word Problems (MWP) solving (Xie and Sun, 2019a; Li et al., 2020; Shen and Jin, 2020), a tree-structured neural model is applied in the FinMath framework. Specifically, for those arithmetic questions in TAT-QA, after extracting the supporting evidence, the tree-structured neural model uses top-down goal decomposition and bottom-up subtree embedding construction to directly predict the expression tree from questions and extracted evidence. Then the expression tree is executed to get the final answer. The main contribution of this work can be summarized as follows:

- We propose a new framework named FinMath to answer financial questions in an expert-like way. Specifically, the model first understand the hybrid context of financial reports and extract supporting evidence given the questions. Then a tree-structured neural model is applied to perform multi-step numerical reasoning for those arithmetic questions in TAT-QA.
- The experimental results show FinMath significantly outperforms several state-of-the-art systems over TAT-QA dataset. Detailed ablation study shows that FinMath model improves the previous best result by 14.7% absolute for solving arithmetic questions, which illustrates the model's capability of multi-step numerical reasoning.

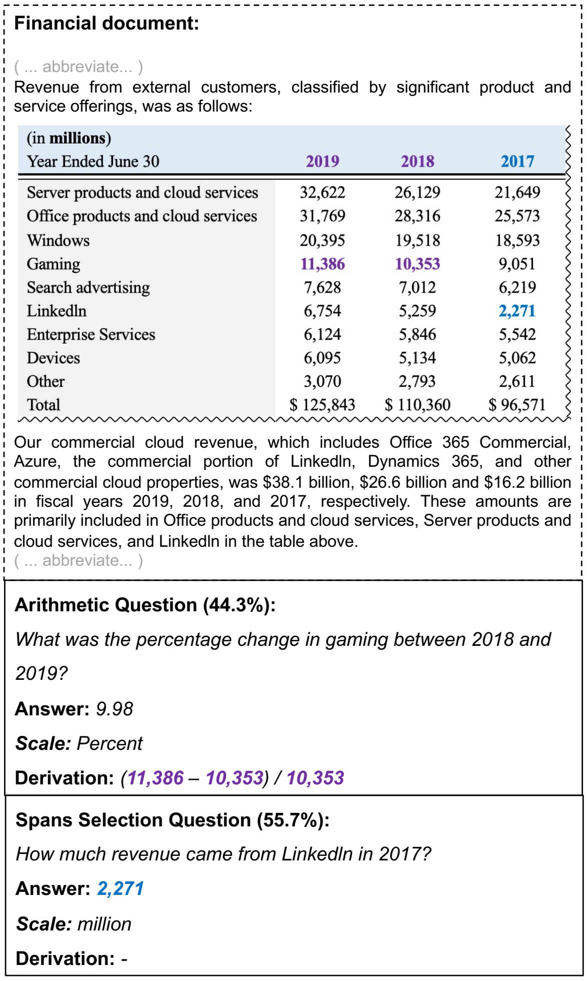
2. Task Formulation.

Presented with a financial report consisting of textual contents P and tabular contents T, given a question Q, the model first aims to classify whether the Q is a spans selection question Q_S, or an arithmetic (numerical reasoning) question Q_N.

For Q_S type questions, the task is to select all the predicted cells from T and spans from P as $X = \{x_0, x_1, ..., x_n\}$.

For Q_N type question, the task is to:

1. Generate the numerical expression $E = \{w_0, w_1, ..., w_n\}$, where w_i is constant quantity, mathematical operator, or numeric value from X.



Financial document:

(... abbreviate...)

Revenue from external customers, classified by significant product and service offerings, was as follows:

(in **millions**) Year Ended June 30	2019	2018	2017
Server products and cloud services	32,622	26,129	21,649
Office products and cloud services	31,769	28,316	25,573
Windows	20,395	19,518	18,593
Gaming	**11,386**	**10,353**	9,051
Search advertising	7,628	7,012	6,219
LinkedIn	6,754	5,259	**2,271**
Enterprise Services	6,124	5,846	5,542
Devices	6,095	5,134	5,062
Other	3,070	2,793	2,611
Total	$ 125,843	$ 110,360	$ 96,571

Our commercial cloud revenue, which includes Office 365 Commercial, Azure, the commercial portion of LinkedIn, Dynamics 365, and other commercial cloud properties, was $38.1 billion, $26.6 billion and $16.2 billion in fiscal years 2019, 2018, and 2017, respectively. These amounts are primarily included in Office products and cloud services, Server products and cloud services, and LinkedIn in the table above.

(... abbreviate...)

Arithmetic Question (44.3%):

What was the percentage change in gaming between 2018 and 2019?

Answer: *9.98*

Scale: *Percent*

Derivation: ***(11,386 – 10,353) / 10,353***

Spans Selection Question (55.7%):

How much revenue came from LinkedIn in 2017?

Answer: ***2,271***

Scale: *million*

Derivation: -

Figure 1: Examples of TAT-QA dataset. The financial document contain both tabular and textual content. Given the question, the QA systems are required to locate the relevant spans in the document and perform numerical reasoning if necessary.

2. Execute the expression tree to get the answer A for the question:

$$P(A|X,Q) = \sum P(E_i|X,Q) \tag{1}$$

where $\{E_i\}$ are all the correct numerical expressions to evaluate to get the answer.

For both type of questions, the model is also required to predict the scale of the answer, which might be {*None, Thousand, Million, Billion, Percent*}.

3. The FinMath Framework

To address the challenge of TAT-QA and improve the numerical reasoning capability of model, we propose a framework named FinMath. In the first phase, similar with TAGOP (Zhu et al., 2021), a sequence tagging module is applied to extract relevant cells from the table T and text spans from the paragraphs P as supporting evidence. And it also predicts the type of given question Q as spans selection question Q_S or numerical reasoning (arithmetic) question Q_N. In the second phase, inspired by GTS (Xie and Sun, 2019a; Li et al., 2020; Anonymous, 2022), a tree-structured neural model is applied to perform numerical reasoning over arithmetic questions. Details of two modules are discussed below.

3.1. Sequence Tagging

The given question, flattened table by row, and associated paragraphs are input sequentially to a RoBERTa (Liu et al., 2019) encoder to obtain corresponding input representations. Then the model assigns each sub-token either I or O label. The cell in the table or word in the paragraph would be regarded as positive if any of its sub-tokens is tagged with I. For spans selection questions Q_S, the continuous words predicted as positive are combined as a span. And during the testing stage, all positive cells and spans are taken as the outputs. For arithmetic questions Q_N, the tagged sequence $(x_1, x_2, ..., x_n)$ are used as input of tree-structured model in the second phase.

3.2. Tree-structured Neural Model

An auto-regressive sequence-to-tree model similar to GTS (Xie and Sun, 2019a; Anonymous, 2022) is applied in FinMath to generate a numerical expression tree. The generation process can be summarized as following three steps:

Encoding Given all tokens $(q_1, q_2, ..., q_m)$ in questions Q_N and all tokens $(x_1, x_2, ..., x_n)$ in tagged sequence evidence X, an embedding layer and a bidirectional GRU (Cho et al., 2014) are employed to encode all tokens as hidden states $(h_1, h_2, ..., h_{m+n})$, which are then concatenated as h^Q to represent problem Q_N.

Tree Initialization The root node embedding q_0 is initialized as h^Q. The embedding of target vocabulary V_{tar} is initialized as:

$$e(y|Q_N \cup X) = \begin{cases} E_o(y) & if y \in V_o \\ E_c(y) & if y \in V_c \\ h_{loc(y, Q_N \cup X)} & if y \in V_n \end{cases} \tag{2}$$

where V_o, V_c and V_n denote the vocabulary set of operators, constant values, and numeric values appearing in $Q_N \cup X$, respectively; E_o, E_c are two embedding matrices; $loc(y, Q_N \cup X)$ is the position of y in $Q_N \cup X$.

Tree Decoding The tree decoding process involves four modules:

1. Context Module: given the goal vector q and encoder outputs, it generates the context vector c.
2. Prediction Module: given the goal vector q and context vector c, it assigns the predicted token $\hat{y}$ to token with highest decoding score $s(y|Q_N \cup X)$.
3. Combination Module: given the left sub-tree, a recursive neural network is applied to encode it as embedding t_l.

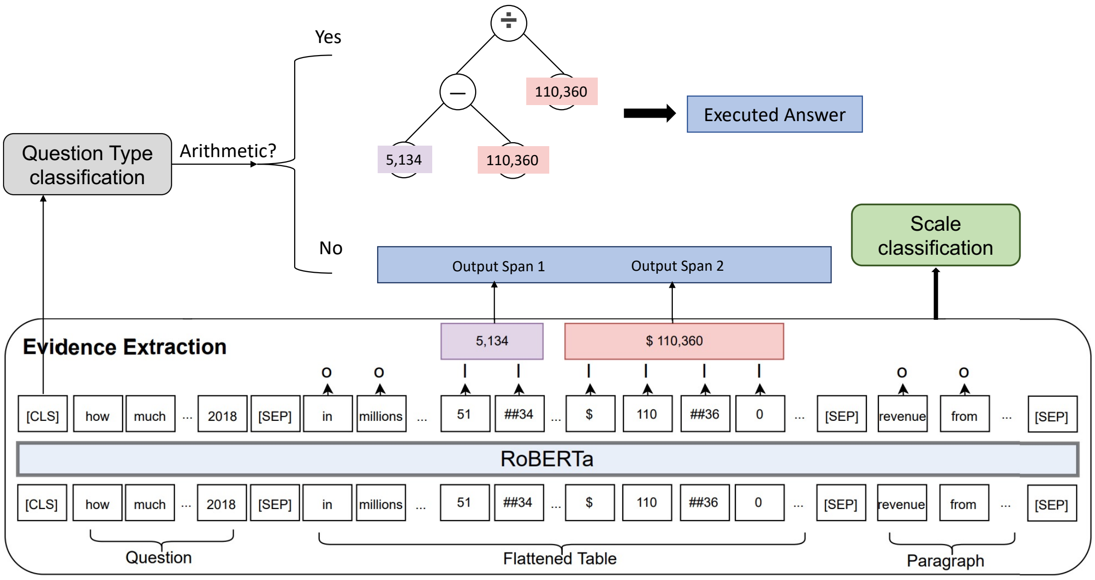

Yes
÷
−
110,360
5,134
110,360
Executed Answer
Question Type classification
Arithmetic?
No
Output Span 1
Output Span 2
Scale classification
Evidence Extraction
5,134
$ 110,360
O O I I I I I I O O
[CLS] how much ... 2018 [SEP] in millions ... 51 ##34 ... $ 110 ##36 0 ... [SEP] revenue from ... [SEP]
RoBERTa
[CLS] how much ... 2018 [SEP] in millions ... 51 ##34 ... $ 110 ##36 0 ... [SEP] revenue from ... [SEP]
Question
Flattened Table
Paragraph

Figure 2: Architecture of proposed FinMath framework.

4. *Left / Right Module*: given the goal vector q and predicted token $\hat{y}$, if $\hat{y}$ is an operator, left module is applied to generate the left sub-goal q_l as $LM(q, e(\hat{y}|Q_N \cup X)$. Otherwise, the right module is applied to generate the right sub-goal q_r as $RM(q, t_l, e(\hat{y}|Q_N \cup X)$. Here, the LM and RM are trainable networks, with implementation the same as GTS model.

The algorithm for tree decoding stage is described in Algorithm 1.

Algorithm 1 Tree Decoding

```
Input: $q_0$, $(h_1, h_2, ...h_{m+n})$
Output: expression tree
 1: Generate context vectors $c$
 2: Generate $q_l$, ŷ
 3: while ŷ $\in V_o$ do
 4:     ŷ = $PredictionModule(q_l, c)$
 5:     $q_l = LM(q, e(\hat{y}|x))$
 6:     $c = ContextModule(q, h_1, ..., h_{m+n})$
 7: end while
 8: Generate $q_r$, $\hat{y}_r$
 9: Combine the embedding of subtree
10: if $\hat{y}_r \in V_o$ then
11:     Jump to line 2
12: else
13:     Recursively find empty right node
14:     if $\hat{y}_r \in V_o$ then
15:         Jump to line 2
16:     else
17:         return expression tree
18:     end if
19: end if
```

4. Experimental Settings

4.1. Baseline Systems

Textual QA Models Two reading comprehension (RC) models over textual data are used as baselines: 1) BERT-RC (Devlin et al., 2019), which achieves promising performance on SQuAD(Rajpurkar et al., 2016; Rajpurkar et al., 2018), a machine reading comprehension dataset; and 2) NumNet+V2 (Ran et al., 2019), which achieves competent performance on DROP dataset (Dua et al., 2019) that requires the model to perform numerical reasoning over textual data.

Tabular QA Model TaPas (Herzig et al., 2020) for WikiTableQuestion dataset (Pasupat and Liang, 2015) is adopted for TAT-QA dataset.

Hybrid QA Model Two hybrid models over textual and tabular data are used as baselines: 1) HyBrider (Chen et al., 2020b), which is the baseline model for HybridQA (Chen et al., 2020b) and tackles hybrid data from Wikipedia; 2) TAGOP (Zhu et al., 2021), which is the state-of-the-art model of TAT-QA dataset. TAGOP first applies sequence tagging, which is also applied in FinMath, to extract relevant cells or text spans from the tables and paragraphs. Then it performs symbolic reasoning over the extracted evidence with a single type of pre-defined aggregation operators (e.g. change ratio, division). Compared with FinMath which can generate numerical expressions with arbitrary steps, TAGOP only supports a single type of aggregation operators, and might fail to answer complex questions requiring multi-step reasoning.

4.2. Evaluation Metrics

Following previous work, we use Exact Match (EM) and numeracy-focused F_1 score as evaluation metrics.

Noted that the calculations of EM and F_1 score are the same for arithmetic question Q_N.

4.3. Implementation Details

To ensure fairness, we use the same encoder (RoBERTa-large), batch size (32), and other training parameter settings (e.g., Adam optimizer, learning rate, etc.) as TAGOP to train FinMath. The training of FinMath model is conducted on two RTX 1080Ti within 12 hours. Since TAT-QA does not release the test set publicly, we use our own split of train, dev, and test set for evaluation (with proportion 8:1:1).

5. Results and Analysis

5.1. Overall Results

	Dev		**Test**	
	EM	F1	EM	F1
HyBrider	6.6	8.3	6.3	7.5
BERT-RC	9.5	17.9	9.1	18.7
TaPas for WTQ	18.9	26.5	16.6	22.8
NumNet+V2	38.1	48.3	37.0	46.9
TAGOP	55.2	62.7	50.1	58.0
FinMath	**60.5**	**66.3**	**58.6**	**64.1**

Table 1: Performance of FinMath compared with different baseline models on dev and test sets of TAT-QA dataset. The results of baseline models are copied from the original TAT-QA paper.

The evaluation results of baseline models and FinMath are summarized in Table 1. It is shown that our model performs better than any other baselines for both EM and F_1 metrics. Specifically, FinMath improves the previous best result (TAGOP) by 8.5% for EM score and 6.1% for F_1 score. The results demonstrates the effectiveness of FinMath in numerical reasoning over tabular and textual data.

5.2. Ablation Study

We also compare detailed performance of FinMath and TAGOP in different answer types of TAT-QA dataset, with the results shown in Table 2. It is shown that FinMath performs much better in arithmetic questions, reaching more than 20% improvement on both three kinds of answer sources. This is because the tree-structured numerical reasoning module in FinMath supports the model to perform a more complex reasoning process than TAGOP. Additionally, FinMath applies the same tagging sequence module as TAGOP, therefore, its performance on spans selection questions are similar with TAGOP.

6. Related Work

Financial NLP Financial NLP has attracted much attention recently. There have been some previous works in the financial domain like fraud detection (Han et al., 2018; Nourbakhsh and Bang, 2019; Wang et al., 2019), market prediction (Day and Lee, 2016; Akhtar et al., 2017) and financial opinion mining and question answering (Maia et al., 2018). More recently, pre-trained language models are presented for finance text mining. (Araci, 2019; Yang et al., 2020). And some recent works (Zhu et al., 2021; Chen et al., 2021; Zhao et al., 2022) focus on numerical reasoning over financial reports with tables.

Answer Source	Spans (EM/F_1)		Arithmetic (EM/F_1)	
	TAGOP	FinMath	TAGOP	FinMath
Table	59.5/63.6	**60.7/64.4**	41.6/41.6	**59.6/59.6**
Text	**43.4/69.8**	42.2/68.9	27.3/27.3	**43.4/43.4**
Table-text	66.4/73.6	**68.7/74.9**	48.3/48.3	**65.7/65.7**
Total	55.4/68.21	**58.9/68.7**	43.5/43.5	**58.2/58.2**

Table 2: Detailed experimental results of TAGOP and FinMath w.r.t. answer types and sources on test set of TAT-QA dataset.

Numerical Reasoning Numerical reasoning plays an important role in areas like question answering (Dua et al., 2019; Andor et al., 2019; Ran et al., 2019; Herzig et al., 2020; Chen et al., 2020a; Yin et al., 2020; Chen et al., 2021) and math word problems (MWPs) solving (Xie and Sun, 2019b; Amini et al., 2019; Koncel-Kedziorski et al., 2016; Hendrycks et al., 2021; Hong et al., 2021). Current approaches usually regard solving MWPs as a sequence to sequence task. And Seq2Seq model (Wang et al., 2017; Robaidek et al., 2018; Anonymous, 2022), with an encoder-decoder framework to generate the solution, has attracted much attention before 2018. Later some work (Xie and Sun, 2019a; Li et al., 2020) proposed tree-structured model to better fit the goal-driven mechanism in human problem solving.

7. Conclusion

In this paper, we have proposed FinMath, a novel framework that aims to conduct complex numerical reasoning over financial reports containing both tabular and textual data. We evaluate the effectiveness of FinMath on TAT-QA dataset. The results of comprehensive experiments showed that the proposed FinMath, with the tree-structured neural model to perform multi-step numerical reasoning, improves the previous best result by 8.5% absolute for Exact Match (EM) score and 6.1% absolute for numeracy-focused F_1 score.

8. Bibliographical References

Akhtar, M. S., Kumar, A., Ghosal, D., Ekbal, A., and Bhattacharyya, P. (2017). A multilayer perceptron based ensemble technique for fine-grained financial sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language*

Processing, pages 540–546, Copenhagen, Denmark, September. Association for Computational Linguistics.

Amini, A., Gabriel, S., Lin, S., Koncel-Kedziorski, R., Choi, Y., and Hajishirzi, H. (2019). Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367.

Andor, D., He, L., Lee, K., and Pitler, E. (2019). Giving BERT a calculator: Finding operations and arguments with reading comprehension. 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)*, pages 5947–5952, Hong Kong, China, November. Association for Computational Linguistics.

Anonymous. (2022). Mathion: Solving math word problems with logically consistent problems. anonymous preprint under review.

Araci, D. (2019). Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Chen, K., Xu, W., Cheng, X., Xiaochuan, Z., Zhang, Y., Song, L., Wang, T., Qi, Y., and Chu, W. (2020a). Question directed graph attention network for numerical reasoning over text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6759–6768, Online, November. Association for Computational Linguistics.

Chen, W., Zha, H., Chen, Z., Xiong, W., Wang, H., and Wang, W. (2020b). Hybridqa: A dataset of multi-hop question answering over tabular and textual data. *Findings of EMNLP 2020*.

Chen, Z., Chen, W., Smiley, C., Shah, S., Borova, I., Langdon, D., Moussa, R., Beane, M., Huang, T.-H., Routledge, B., and Wang, W. Y. (2021). Finqa: A dataset of numerical reasoning over financial data. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Day, M.-Y. and Lee, C.-C. (2016). Deep learning for financial sentiment analysis on finance news providers. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1127–1134. IEEE.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., and Gardner, M. (2019). DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proc. of NAACL*.

Han, J., Barman, U., Hayes, J., Du, J., Burgin, E., and Wan, D. (2018). NextGen AML: Distributed deep learning based language technologies to augment anti money laundering investigation. In *Proceedings of ACL 2018, System Demonstrations*, pages 37–42, Melbourne, Australia, July. Association for Computational Linguistics.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. (2021). Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Herzig, J., Nowak, P. K., Mueller, T., Piccinno, F., and Eisenschlos, J. (2020). Tapas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333.

Hong, Y., Li, Q., Ciao, D., Huang, S., and Zhu, S.-C. (2021). Learning by fixing: Solving math word problems with weak supervision. In *AAAI Conference on Artificial Intelligence*.

Koncel-Kedziorski, R., Roy, S., Amini, A., Kushman, N., and Hajishirzi, H. (2016). Mawps: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1152–1157.

Li, S., Wu, L., Feng, S., Xu, F., Xu, F., and Zhong, S. (2020). Graph-to-tree neural networks for learning structured input-output translation with applications to semantic parsing and math word problem. *arXiv preprint arXiv:2004.13781*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach.

Maia, M., Handschuh, S., Freitas, A., Davis, B., McDermott, R., Zarrouk, M., and Balahur, A. (2018). Www'18 open challenge: financial opinion mining and question answering. In *Companion Proceedings of the The Web Conference 2018*, pages 1941–1942.

Nourbakhsh, A. and Bang, G. (2019). A framework for anomaly detection using language modeling, and its applications to finance. *CoRR*, abs/1908.09156.

Pasupat, P. and Liang, P. (2015). Compositional semantic parsing on semi-structured tables.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.

Ran, Q., Lin, Y., Li, P., Zhou, J., and Liu, Z. (2019). Numnet: Machine reading comprehension with numerical reasoning. 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)*, pages 2474–2484.

Robaidek, B., Koncel-Kedziorski, R., and Hajishirzi, H. (2018). Data-driven methods for solving algebra word problems.

Shen, Y. and Jin, C. (2020). Solving math word problems with multi-encoders and multi-decoders. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2924–2934.

Wang, Y., Liu, X., and Shi, S. (2017). Deep neural solver for math word problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 845–854, Copenhagen, Denmark, September. Association for Computational Linguistics.

Wang, W., Zhang, J., Li, Q., Zong, C., and Li, Z. (2019). Are you for real? detecting identity fraud via dialogue interactions. 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)*, pages 1762–1771, Hong Kong, China, November. Association for Computational Linguistics.

Xie, Z. and Sun, S. (2019a). A goal-driven tree-structured neural model for math word problems. In *IJCAI*, pages 5299–5305.

Xie, Z. and Sun, S. (2019b). A goal-driven tree-structured neural model for math word problems. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5299–5305. International Joint Conferences on Artificial Intelligence Organization, 7.

Yang, Y., Uy, M. C. S., and Huang, A. (2020). Finbert: A pretrained language model for financial communications. *CoRR*, abs/2006.08097.

Yin, P., Neubig, G., Yih, W.-t., and Riedel, S. (2020). TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online, July. Association for Computational Linguistics.

Zhao, Y., Li, Y., Li, C., and Zhang, R. (2022). Multihiertt: Numerical reasoning over multi hierarchical tabular and textual data. In *ACL 2022*.

Zhu, F., Lei, W., Huang, Y., Wang, C., Zhang, S., Lv, J., Feng, F., and Chua, T.-S. (2021). TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, August.