Arthur Poon
Professor Dave Sahota
MPCS 53120: Applied Data Analysis
May 24, 2023

Improving XBRL Tagging by Randomizing Numeric and Shape Tokens

Executive Summary:

Despite vast data within the SEC's EDGAR system, extracting information for analysis is manual and labor-intensive due to its analog nature. Although the SEC has adopted machine-readable disclosure formats like XBRL and iXBRL, the challenge of creating custom tags persists. These tags, which account for about 42% of those in financial statement notes, are time-consuming and costly to develop, and financial statement notes are a goldmine of information about companies. This paper aims to contribute to streamlining the creation of custom tags, enhancing the efficiency of data analysis, and fostering informed decision-making.

I. Introduction:

On May 3, 2018, Deputy Chief Economist and Deputy Director, Division of Economic and Risk Analysis Scott W. Bauguess delivered a speech on *The Role of Machine Readability in an AI World*. While he touches on a variety of topics, I want to highlight the aspects of his speech that elucidates a 'hair-on-fire' need that is an underpinning motivation behind this paper—the need for financial disclosures to be machine readable.

a) Plethora of data, but low accessibility:

The volume of decision-relevant data available in SEC (also hereon referred to as "the Commission") registrant disclosures is immense. The EDGAR filing system contains financial information that covers over $82 trillion in assets managed by registered investment advisors. It also hosts financial statements from publicly traded companies with a combined market capitalization of approximately $30 trillion. Since its inception, more than 11 million filings have been made by over 600,000 reporting entities using 478 unique form types. In 2016 alone, there were over 1.5 billion unique requests for this information through the SEC.gov website.[1]

However, not all the data in SEC filings is easily accessible for data analysis. Many of the required forms and filings are narrative-based and intended for human reading,

[1] Bauguess, Scott W. The Role of Machine Readability in an AI World, 3 May 2018, www.sec.gov/news/speech/speech-bauguess-050318.

making the extraction of numerical information and text-based disclosures a manual process. Analysts and researchers had to manually identify, copy, and paste each relevant piece of information from the filings to spreadsheets to conduct their analyses to inform their opinions.

The analog characteristics of the reporting system were designed before the advent of machine learning methods, which have complicated the use of the EDGAR filing system today. Now, on any given day, as much as 85 percent of the accessed documents are visited by internet bots. Fortunately, the Commission has been preparing for this by implementing rules for machine-readable disclosures since 2003 and has been proposing or adopting additional rules requiring structured disclosure.

b) XBRL and iXBRL formats

In 2009, the Commission ruled for the adoption of eXtensive Business Reporting Language (XBRL), an open standard format that is widely available to the public royalty-free at no cost.[2] XBRL is a standardized markup language specifically developed for financial reporting and analysis. It enables the tagging of financial data with metadata, providing context and meaning to individual data elements. It allows for the structured representation of financial information, making it easier for machines to interpret financial data. However, XBRL files were typically separate from the human-readable versions, and special software or tools were required to interpret and work with the data.

In 2017, the Commission then proposed the requirement for Inline XBRL format (iXBRL). iXBRL is an extension of XBRL that combines the machine-readable XBRL data with the human-readable HTML format into a single document. In iXBRL, the XBRL tags are embedded directly within the HTML document using inline tagging. This means that users can view the document in a web browser and read it as they would with a regular HTML file, while still having access to the underlying structured data encoded in XBRL. iXBRL files retain the advantages of XBRL in terms of data analysis and exchange, but they also provide a user-friendly presentation that requires no specialized software to read.

In 2018, the SEC voted to require the use of iXBRL for financial statement information and risk/return summaries.[3]

From a machine-readability perspective, financial statement data, footnotes, and other important information contained in an iXBRL filing can be easily and automatically extracted, processed, and combined with similar data from other 10-K filings. This aggregation is possible because each data element or section of text is tagged using definitions from a common reporting taxonomy. From a machine learning perspective, this standardized data can be analyzed along with other financial information and market actions to identify patterns and potentially predict future registrant behavior.

---

[2] “Final Rule: Interactive Data to Improve Financial Reporting - Sec.Gov.” Interactive Data to Improve Financial Reporting , 13 Apr. 2009, www.sec.gov/rules/final/2009/33-9002.pdf.
[3] SEC Adopts Inline XBRL for Tagged Data, 28 June 2018, www.sec.gov/news/press-release/2018-117.

c) Custom Tags

Over time, the XBRL standard has developed over 300 standard axis tags exist in the U.S. GAAP taxonomy, and the average annual XBRL exhibit uses about 20 axis tags.[4] However, there is often a need to create a custom tag – one that is not in the standard taxonomy. This is a particularly acute need in the financial statement notes, which include the company-specific nuances underlying a company's financials. In 2021, the Commission's Division of Economic and Risk Analysis (DERA) analyzed custom tags used in XBRL filings submitted during fiscal years 2019 through 2021. They found that in FY2019 to FY2021, the percentage of custom tags across several crucial IFRS financial forms reached ~42% of total tags in financial statement notes, as compared to the ~20% of tags in financial statements.[5]

d) The problem: cost of XBRL compliance in financial statement notes

Manually tagging reports with XBRL tags is a laborious and highly specialized skill, which is why the human resource-intensive task is often outsourced companies such as PwC, DFIN, and M2 Compliance to complete.[6] In general, being XBRL compliant has become easier over time if a company need only draw from the existing taxonomy of tags. However, for businesses with high degree of fluctuation in business operations and complex financial notes, e.g. companies with complicated organization structures, diversified industry verticals, and underwriting structures like Berkshire Hathaway and General Electric, end up creating custom tags in highly variant, unstructured, financial statement notes—this is the main problem that I hope to tackle in this paper.

II. Literature Review:

In connection with this paper, I will be reviewing two pieces of relevant literature that have inspired/informed my proposed solution. The first paper is *FiNER: Financial Numeric Entity Recognition for XBRL Tagging* by Loukas et al., a paper that first defined XBRL Tagging as the entity extraction task for the financial domain. The second paper is *FinBERT-MRC: financial named entity recognition using BERT under the machine reading comprehension paradigm* by Zhang and Zhang, a paper that focuses on

---

[4] Staff Observations of Custom Axis Tags, 29 Mar. 2016, www.sec.gov/structureddata/reportspubs/osd_assessment_custom-axis-tags.
[5] IFRS – XBRL Custom Tags Trend, 29 Sept. 2022, www.sec.gov/structureddata/ifrs_trends_2021.
[6] M2 Compliance, Unlimited EDGAR | iXBRL, www.m2compliance.com/. Accessed 24 May 2023.; PricewaterhouseCoopers. "Achieving Trust and Efficient Financial Reporting through XBRL." PwC, www.pwc.com/us/en/services/trust-solutions/xbrl.html. Accessed 24 May 2023.; "Secure Financial Reporting and SEC Filing." DFIN, 1 May 2023, www.dfinsolutions.com/products/activedisclosure?utm_source=google&utm_medium=Search-Paid&utm_offer=Product-Push&utm_touch=dfin-sfdc-NA_Google_Search-Paid&utm_campaign=nb_activedisclosure_specific&sfcampid=7013b000001f5lMAAQ&utm_content=ad_campaign_launch&utm_custom1=expanded_text_ad&utm_custom2=ixbrl&gclid=Cj0KCQjwmZejBhC_ARIsAGhCqndc-vM2KuMdGaEpQFiT-5fjbJpV09J72M5CW6lmuVatUXohWAJFcvsaAuBOEALw_wcB.

financial named entity recognition (FinNER), a task that aims to automatically retrieve financial entities in unstructured texts.

a) Loukas et al.: FiNER: Financial Numeric Entity Recognition for XBRL Tagging[7]

i) Overview

This study focuses on developing a model that could automate the process of annotating financial reports with eXtensible Business Reporting Language (XBRL) tags. XBRL is a standardized language for digital business reporting that allows data to be extracted and analyzed quickly and cost-effectively. While structured data tables in these reports can be easily tagged, the unstructured narrative or text notes pose a challenge. These notes often contain critical financial details but are difficult to annotate due to their free-form nature. The study aims to address this gap by using machine learning models that can automate the process, making both new and old financial reports more accessible and analyzable.

ii) Data used

The research used a dataset, named FiNER-139, which was built from approximately 10,000 annual and quarterly reports filed with the SEC by publicly traded companies over a five-year period from 2016 to 2020. This dataset is unique because it comprises 1.1 million sentences with XBRL tags annotated by professional auditors, which covers a wide range of financial statements detailing company performance and future projections. The text notes within these reports were extracted via regular expression, a tool used to identify specific patterns within text data.

iii) Data preprocessing

The sentences were stripped of HTML formatting, normalized, and converted to lowercase. To distinguish between trivial and more complex cases, they employed heuristic rules to discard sentences that almost certainly did not require tagging.

The heuristic rules were formulated by inspecting the training subset and include regular expressions to identify amounts and other expressions typically annotated. These rules resulted in the removal of about 40% of the 1.8 million sentences, eliminating only 1% of tagged sentences. The remaining sentences were chronologically split into training, development, and test sets in an 80/10/10 ratio.

For the handling of numerical values, Loukas et al. introduced two approaches: BERT + NUM and BERT + [shape]. In BERT + NUM, numbers detected using regular expressions were replaced with a single [num] pseudo-token, a token incapable of

[7] Loukas, Lefteris, et al. “Finer: Financial Numeric Entity Recognition for XBRL Tagging.” Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, 19 Apr. 2022, https://doi.org/10.18653/v1/2022.acl-long.303.

being split. This pseudo-token was added to the BERT vocabulary, and its representation was learned during fine-tuning. This procedure allows all numeric expressions to be handled uniformly, preventing their fragmentation.

In BERT + [shape], numbers were replaced with pseudo-tokens that represent the shape of the number and cannot be split (i.e., only retaining digits). For instance, '53.2' was transformed into '[XX.X]', and '40,200.5' became '[XX,XXX.X]'. They utilized 214 special tokens that covered all number shapes in the training set. The representations of these pseudo-tokens were also fine-tuned, and as a result, numeric expressions (of known shapes) were no longer fragmented. The shape pseudo-tokens also capture information about the magnitude of each number, hinting at the idea that numeric tokens of similar magnitudes may necessitate similar XBRL tags.

iv) Models considered and used

spaCy: This open-source Natural Language Processing (NLP) library from Honnibal et al., 2020 includes an industrial Named Entity Recognition (NER) tool that uses word-level Bloom embeddings (Serrà and Karatzoglou, 2017) and residual Convolutional Neural Networks (CNNs) (He et al., 2016). For this work, Loukas et al. trained spaCy's NER from scratch on the FiNER-139 dataset.

BiLSTM: As a baseline, they also utilized a stacked bidirectional Long Short-Term Memory (LSTM) network (Graves et al., 2013; Lample et al., 2016) with residual connections. In this model, each token of a sentence is mapped to an embedding and passed through the BiLSTM stack to extract the corresponding contextualized embedding. A shared multinomial logistic regression (LR) layer operates on top of each contextualized embedding to predict the correct label.

BERT: Similar to BiLSTM, Loukas et al. fine-tuned BERT-base (Devlin et al., 2019) to extract contextualized embeddings of subwords. A multinomial LR layer was also used to operate on the contextualized embeddings to predict the appropriate label for the corresponding subword.

CRFs: Loukas et al. replaced the LR layer in the previous two models with a Conditional Random Field (CRF) layer (Lafferty et al., 2001). This approach has demonstrated its utility in several token labeling tasks (Huang et al., 2015; Lample et al., 2016; Chalkidis et al., 2020b).

In an attempt to enhance BERT's performance, Loukas et al. further fine-tuned and utilized two BERT models:

Fin-BERT: Fin-BERT (Yang et al., 2020) is a BERT model pre-trained on a financial corpus comprising SEC documents, earnings call transcripts, and analyst reports. Its 30k subwords vocabulary is built from scratch from its pre-training corpus. Loukas et al. used Fin-BERT with and without their numeric pseudo-tokens, whose representations are learned during fine-tuning.

SEC-BERT: Loukas et al. created their own family of BERT models, initially pre-training BERT from scratch on the EDGAR-corpus, a collection of financial documents they created in a previous paper. The resulting model, termed SEC-BERT, had a new vocabulary of 30k subwords. To further assess the impact of the proposed [num] and [shape] special tokens, they pre-trained two additional BERT variants, SEC-BERT-NUM and SEC-BERT-SHAPE, on the same corpus. In these models, all numbers are replaced by [num] or [shape] pseudo-tokens, respectively, and the representations of these pseudo-tokens are learned during pre-training and updated during fine-tuning.

v) Metrics for performance evaluation

Loukas et al. used the micro-F1 (μ-F1) and macro-F1 (m-F1) scores as metrics to evaluate the models' performance at the entity level. The F1 score is the harmonic mean of precision and recall, and it's a common measure of a test's accuracy. Micro-F1 considers the total true positives, false negatives, and false positives, while macro-F1 calculates the F1 separately for each class and then takes the average.

vi) Evaluation

The effectiveness of Conditional Random Fields (CRFs) was examined, and the results indicated that their inconsistent performance could be due to tokenization differences. A significant drawback of CRFs is their tendency to slow down the models during both training and inference, particularly with large label sets, as in their case.

In FiNER-139, the majority (91.2%) of the tagged spans are numeric expressions. Because these cannot all be included in BERT’s finite vocabulary, they often result in excessive fragmentation when subword tokenization is used. This fragmentation leads to an increased probability of producing nonsensical sequences of labels.

Further, an exploratory data and error analysis was carried out to understand the peculiarities of FiNER-139 and the limitations of their best model. Loukas et al. identified three main sources of errors: specialized terminology, financial dates, and annotation inconsistencies. Despite the gold XBRL tags of FiNER-139 being provided by professional auditors, some discrepancies were still observed, likely due to the inherent complexity of real-world financial reporting.

vii) Results

The results demonstrated that spaCy performed poorly. BiLSTM (with word embeddings) performed slightly better than BERT, but with the addition of a CRF layer, BERT demonstrated superior results while the BiLSTM's performance significantly worsened.

BERT's performance improved with the addition of the [num] pseudo-token, which prevents fragmentation of numeric expressions. The implementation of shape pseudo-

tokens ([shape]) led to further performance improvement, achieving a 79.4 micro-averaged F1 score, indicating the importance of numeric magnitude in XBRL tagging.

FinBERT performed worse than BERT despite being pre-trained on financial data. This was attributed to the fragmentation of numbers, similar to the issue faced by BERT. However, the introduction of the proposed pseudo-tokens ([num], [shape]) significantly improved FinBERT's performance, enabling it to leverage its in-domain pre-training to surpass BERT, achieving an 80.1 micro-averaged F1 test score.

The SEC-BERT model, pre-trained on SEC reports, outperformed the existing BERT and FinBERT models when numeric pseudo-tokens were not used. Nevertheless, it performed worse than BERT with numeric pseudo-tokens and also worse than the BiLSTM with word embeddings. The incorporation of the proposed pseudo-tokens led to a significant improvement in SEC-BERT's performance, with SEC-BERT-num and SEC-BERT-shape achieving the best overall results, attaining a micro-averaged F1 test score of 80.4 and 82.1, respectively.

viii) Implications for my project

One implication of this publication for my project is the creation of the FiNER-139 dataset on which I can continue to explore new models. Furthermore, they showed numeric tokens and pseudo-tokens reflecting token shapes and magnitudes significantly boosts the performance of BERT-based models in this task.

b) Zhang and Zhang: FinBERT-MRC: financial named entity recognition using BERT under the machine reading comprehension paradigm[8]

i) Overview

The challenge in the field of financial text information extraction is the recognition of financial entities from literature. A common approach is the financial named entity recognition (FinNER) task, using a sequence tagging framework. However, it doesn't fully exploit semantic information. In this study, Zhang and Zhang redefine the FinNER task as a machine reading comprehension (MRC) problem and propose a novel model, FinBERT-MRC. This model introduces significant prior information using well-designed queries and extracts the start and end index of target entities without the need for decoding modules like conditional random fields (CRF). The MRC framework, in contrast to the sequence labeling framework, allows the introduction of prior knowledge, improving model performance.

ii) Data used

[8] Zhang, Yuzhe, and Hong Zhang. “FinBERT–MRC: Financial Named Entity Recognition Using Bert under the Machine Reading Comprehension Paradigm.” Neural Processing Letters, 31 May 2022, https://doi.org/10.1007/s11063-023-11266-5.

The first dataset used is the ten-year ChFinAnn3 documents from 2008-2018, containing 32,040 Chinese financial announcement documents, categorized into 24 different entity types. This data has been pre-split into training, validation, and test sets.

The second dataset, AdminPunish, is a real-world business dataset collecting punishment announcements released by local authorities to punish violations of administrative orders. It contains 2,596 annotated sentences with seven types of entities.

iii) Data preprocessing

Due to computational constraints, a portion of the ChFinAnn dataset was used through a resampling scheme. This process involved randomly extracting documents five times from the original training, validation, and test sets, omitting sentences that did not contain entities from the 3 sets. The AdminPunish dataset was split into training, validation, and test sets with a ratio of 6:2:2, also repeating five times.

Entities in the ChFinAnn dataset were merged according to their actual meanings, simplifying the task, resulting in 10 types of entities. For example, both of the entities “HighestTradingPrice” and “LowestTradingPrice” represent price, so they were integrated into a common entity “Price”. For the AdminPunish dataset, all seven entities remain unchanged.

iv) Models considered and used

The FinBERT model was used, considering its domain-specific attributes for financial tasks. This model was compared with other well-known models like BiLSTM-CRF, BERT-Tagger, BERT-CRF, and BERT-MRC. For fair comparison, identical hyperparameters were set for BERT-MRC and FinBERT-MRC. For BERT-Tagger, BERT-CRF and BERT-MRC, they utilize the pretrained BERTbase-chinese as the model backbone, and for FinBERT-MRC, they choose FinBERT as the model backbone. For fair comparison, identical hyperparameters were set for BERT-MRC and FinBERT-MRC.

v) Metrics for performance evaluation

The performance of each model is evaluated with the F1-score (F1) defined as F1 = 2P R/(P + R), where P and R are precision and recall, respectively. Zhang and Zhang then report the mean value of each measure (P, R, F1) as well as the standard deviation on the test sets in mean ± std format.

vi) Evaluation

The effectiveness of machine reading comprehension models can significantly depend on the structure and information richness of the query presented. In this context, Zhang and Zhang explore three methods of constructing queries:

- Keyword: The keyword representing the entity tag serves as the query.
- Rule-based query template: Templates are used to generate the queries.
- Wikipedia: The Wikipedia definition of the entity is used to build the query.

They used the above three query construction methods to train the FinBERT-MRC model on both the ChFinAnn and AdminPunish datasets. The evaluation showed that all three methods outperformed the baseline BERT-Tagger model, indicating that introducing prior knowledge via the queries can enhance model performance. Among the three, the Wikipedia definition-based queries (MRC-Wiki) performed the best across all test cases.

In addition, they found the F1 score for both BERT-Tagger and FinBERT-MRC decreased as the size of the training samples increased. However, FinBERT-MRC demonstrated more robust performance, maintaining a relatively high F1 score even with larger sample sizes. This suggests that the use of prior knowledge encoded within the entity further enhances the performance of FinBERT-MRC.

Furthermore, they found that using a domain-specific pretrained language model, such as FinBERT-MRC, can lead to improved generalization capability, including the ability to mitigate false negatives and improve boundary recognition, thereby surpassing BERT-Tagger in both syntactic and semantic learning.

vii) Results

The results clearly demonstrate that FinBERT-MRC achieves the highest F1 scores on both ChFinAnn and AdminPunish datasets, with average F1 scores of 92.78% and 96.80% respectively. Compared to BERT-MRC, FinBERT-MRC improves the F1 scores by 0.88% and 0.23% on the ChFinAnn and AdminPunish datasets respectively, which confirms that using domain-specific pretrained language models effectively improves performance in downstream tasks.

viii) Implications for your project

First, the use of a pretrained language model trained on domain-specific corpora offers better semantic representation for input texts, beneficial for domain-specific downstream tasks. Secondly, appropriately designed queries encode important prior knowledge about the entity to extract, learning jointly with the original texts. This instructs the model to accurately locate the entity span.

III. Proposed solution: randomizing numeric and shape tokens

My proposed solution is to combine the best performing BERT-NUM and BERT-SHAPE models from Loukas et al. in a mixed model. I randomly replaced recognized numeric entities with the shape token or the NUM token with 50% probability for each.

The inspiration behind is that hopefully we could leverage the two high performing models and benefit from the benefits of both them to see if this would improve recognition.

I decided not to utilize the methods from the Zhang and Zhang model because the MRC method requires many forward passes to build the MRC and this would take too long to train.

a) Data used:

The data that I used was the FiNER-139 dataset that was used in the Loukas et al. paper. This was taken from Hugging Face.[9] The details of this dataset are also discussed in section II. a) ii) of this paper.

b) Software used:

My work can be found in the following Google Colab notebook.

Mainly, I leveraged the framework for BERT model training that Loukas et al. had created in their finer github directory.[10]

The libraries that I used included:

- datasets: The datasets library by Hugging Face which I use to load the finer-139 dataset.
- transformers: A library from Hugging Face that provides pretrained models. From this library, Loukas et al. utilized the following:
    - Autotokenizer: they used this to load the pre-trained tokenizer that Loukas et al. utilized in their SEC-BERT-base model
    - BertConfig: they used the BertConfig class to specify the configuration for the BERT model.
    - BertForMaskedLM: the BERT model used on the data, it was configured with the above BertConfig configurations.

c) Results and comparison against benchmark

The results of my experiment were as follows:

Loss & Macro F1 Report					
Epoch	Loss	Val Loss	Val Micro F1	Val Macro F1	Learning Rate
1	0.014603	0.006380	0.423119	0.164312	1.000e-05

[9] https://huggingface.co/datasets/nlpaueb/finer-139
[10] https://github.com/nlpaueb/finer

The model that I proposed produces a micro-averaged F1 test score of 42.3, which is significantly worse than that of the SEC-BERT-SHAPE and SEC-BERT-NUM models which have micro F1 scores of 80.4 and 82.1, respectively. Thus, based on these results and comparison, my solution does not improve upon the models individually, by randomizing the application of both.

However, my results were also produced under 50,000 training and validation rows, and only under one epoch due to time constraints given my limited computing resources. It is unknown if my model would perform better, equally, or worse than benchmark if it was able to train on all 900,000+ rows of training data and run along 20 epochs as Loukas et al. trained their models on.

IV. Project Reflection

d) Project effort distribution

The effort distribution on this project was as follows:

Activity	Effort/Time distribution
Research into past publications	10%
Ideation of solution	2%
Software documentation comprehension	70%
Creation of code	18%

Undoubtedly, I spent most of my effort on understanding software documentation. The learning curve to be able to replicate the work that Loukas et al. did to create their models was extremely high. Having to parse through their entire Github directory and understand how their models were made to then subsequently manipulate them for my own solution was extremely difficult. The writing of the code itself was not as difficult because I could lean on the structure and design that was within Loukas et al.'s repository, but I needed to learn how to modify and manipulate the models for my solution, and to decrease the time requirements for training.

e) Roadblocks, tradeoffs, and workarounds

Initially, I tried to combine the MRC and querying methods from Zhang and Zhang into the SEC-BERT models that were created by Loukas et al. However, this was far too complex and the documentation from Zhang and Zhang was extremely difficult to navigate and lacked a lot of support for me to properly understand and replicate the methodologies of their work. Furthermore, the MRC method would require multiple forward passes over the data for each variable which would take immense amount of time. Thus, in the essence of time, I decided to focus only on Loukas et al.'s models and not bring in the MRC methods.

The final solution of combing the SEC-BERT-Shape and SEC-BERT-NUM models was in hopes to increase the accuracy of recognizing XBRL tags even more by randomizing the use of both.

In evaluating and training this solution, the main tradeoff I had to think about was training sample size and number of epochs versus training time. The original FiNER dataset had 900,000 plus rows. The first time I ran the model training it took more than 7 hours to complete a single epoch. Thus, when I made future modifications, I decided I needed to shrink the model's training and validation sets drastically to 50,000 rows in order to have a reasonable training time and work on the model iteratively.[11] Furthermore, Loukas et al. originally had their models train on 20 epochs. As previously mentioned, training on a single epoch already took 7 hours, so I could only work with 1 to allow myself to work on the model iteratively.[12]

f) Interesting/challenging aspects of the project

I think what I found interesting about the project was most definitely the potential there is to apply structure to such a dense piece of data such as financial filings. As someone who had previously worked in finance and financial statements and filings were very much my entire life, I spent endless hours manually analyzing them to understand companies. The fact that we can now potentially automate machine readability and to reason on this data seems like an enormous potential to draw insights across companies at scale.

What I found most challenging about the project was undoubtedly the learning curve. The documentation was extremely dense. I had not had previous experience using BERT models and it was an extreme challenge over 6 entire days and roughly 38 hours over the course of the quarter to simply get the BERT models to work in a similar way they did in Loukas et al.

g) Ideal data that was not available

In an ideal world, I would have wanted to get all the most recent filing data to get the most recent up to date taxonomy of XBRL tags. However, the FiNER dataset was only limited to the timeframe of 2016 to 2020.

h) Next steps, 1-3 month goals

If I had more time, I would greatly increase my training dataset and the number of epochs to really test out whether my solution works if it is given the same training opportunity and amount of data that Loukas et al. provided their models.

---

[11] All edits to the code were completed locally on a copied version of the FiNER directory and then uploaded to my own final report repository. This edit of sample sizes was made in finer.py.
[12] This edit was made in transformers.json, where "epochs" was replaced with a value of 1.

Furthermore, I would want to expand upon the idea of discerning time-associated tags. Loukas et al made crucial contributions in identifying the best way to pick up on XBRL tags. However, as they note in their paper, it is very difficult to discern the relation between numbers. I think this is where the contributions of the MRC method from Zhang and Zhang become very interesting. If I had more time I would love to explore the idea of introducing previous knowledge into XBRL tagging using the MRC method.

Works Cited

Staff Observations of Custom Axis Tags, 29 Mar. 2016, www.sec.gov/structureddata/reportspubs/osd_assessment_custom-axis-tags.

Bauguess, Scott W. *The Role of Machine Readability in an AI World*, 3 May 2018, www.sec.gov/news/speech/speech-bauguess-050318.

"Final Rule: Interactive Data to Improve Financial Reporting - Sec.Gov." *Interactive Data to Improve Financial Reporting* , 13 Apr. 2009, www.sec.gov/rules/final/2009/33-9002.pdf.

Financial Numeric Entity Recognition for XBRL Tagging, 22 May 2022, github.com/nlpaueb/finer.

IFRS – XBRL Custom Tags Trend, 29 Sept. 2022, www.sec.gov/structureddata/ifrs_trends_2021.

Loukas, Lefteris, et al. "Finer: Financial Numeric Entity Recognition for XBRL Tagging." *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 19 Apr. 2022, https://doi.org/10.18653/v1/2022.acl-long.303.

Loukas, Lefteris, et al. "NLPAUEB/Finer-139." *Nlpaueb/Finer-139 · Datasets at Hugging Face*, huggingface.co/datasets/nlpaueb/finer-139. Accessed 24 May 2023.

M2 Compliance, Unlimited EDGAR | iXBRL, www.m2compliance.com/. Accessed 24 May 2023.

PricewaterhouseCoopers. "Achieving Trust and Efficient Financial Reporting through XBRL." *PwC*, www.pwc.com/us/en/services/trust-solutions/xbrl.html. Accessed 24 May 2023.

SEC Adopts Inline XBRL for Tagged Data, 28 June 2018, www.sec.gov/news/press-release/2018-117.

"Secure Financial Reporting and SEC Filing." *DFIN*, 1 May 2023, www.dfinsolutions.com/products/activedisclosure?utm_source=google&utm_medium=Search-Paid&utm_offer=Product-Push&utm_touch=dfin-sfdc-NA_Google_Search-Paid&utm_campaign=nb_activedisclosure_specific&sfcampid=7013b000001f5lMAAQ&utm_content=ad_campaign_launch&utm_custom1=expanded_text_ad&utm_custom2=ixbrl&gclid=Cj0KCQjwmZejBhC_ARIsAGhCqndc-vM2KuMdGaEpQFiT-5fjbJpV09J72M5CW6lmuVatUXohWAJFcvsaAuBOEALw_wcB.

Zhang, Yuzhe, and Hong Zhang. “FinBERT–MRC: Financial Named Entity Recognition Using Bert under the Machine Reading Comprehension Paradigm.” *Neural Processing Letters*, 31 May 2022, https://doi.org/10.1007/s11063-023-11266-5.