FinBERT: pre-trained model on SEC filings for financial natural language tasks

Vinicio DeSola[1], Kevin Hanna[1], Pri Nonis[1]
{penpen1986, kevinhanna, priyankara.nonis}@berkeley.edu

Abstract

Motivation: Machine learning has become an industry changing technology used in securities trading. Models are used to inform human decisions, and increasingly, they are used in making buy and sell decisions without human intervention at speeds no human can perform. The models are critical to the future of this highly competitive and profitable industry, slight improvements can yield large profits and competitive advantages. Annual 10-K reports create a rich source of information as they often disclose new important statements which are often kept deliberately vague making it difficult to identify important changes by experts, let alone natural language processing. Therefore, understanding the minutiae of these reports requires a specialized understanding of the technical language used in these reports and there are no published papers using financial domain specific word embeddings to extract information from 10-K reports more accurately.

Results: We introduce our FinBert models, which outperform BERT on several NLP performance metrics, including a 30% improvement on Next Sentence Predictions and a roughly 25% improvement on Masked LM Prediction on our 10-Q Corpus. We also describe several of the financial language changes between 1999 to 2019.

Implementation: We made our cleaned text training corpora public, and can be found at `http://people.ischool.berkeley.edu/~khanna/fin10-K`. Our implementation can be found at `https://github.com/psnonis/FinBERT`.

Keywords
FinBERT, BERT, Transfer Learning, 10-K

[1] *School of Information - University of California, Berkeley. W266*

Contents

1. Introduction

Federal law requires all public companies to disclose important information about their financial health (SEC, 2009) [15]: How their assets are being used, how much outstanding debt do they have, revenue, and many other facts. This information is very important for both current shareholders of the company, as well as future investors. This information is made public by filing a 10-K report to the SEC (U.S. Securities and Exchange Commission). On any given year, around 5,000 companies will file 10-K reports, collectively these will encapsulate important information on the health of the US economy, and we can cluster it by sectors. Therefore, we mined this large corpus of text to achieve three different goals: 1) Evaluate the evolution of the financial language, 2) Evaluate a language model based on Financial Documents, and 3) Use transfer learning from a generalized domain into financial documents.

In the last 5 years, machine learning techniques have been used extensively in finance for Quantitative Research, fraud detection, compliance and time series analysis, to name a few examples. Potential future advancements are likely in Natural Language Processing (NLP) to better understand the large body of published text. Of the many improvements in the space, the most revolutionary is the use of deep learning and neural networks which are better able to learn context in language corpora.

Deep learning models are efficient and effective for NLP tasks because they require relatively little feature engineering when compared to other options, and they often learn many of the intricacies of languages. However, to become successful performing these tasks, a large amount of training data is required (Sun et al., 2017) [18]. NLP models have made resurgence in the last few years due to the large amount of machine readable data being produced of late, and the increased power of GPU's which allow the parallelization of tasks (Peters, 2017) [12]. This resurgence first started in the field of Image Recognition (Balaban, 2019) [2], and its success has translated to NLP in recent years.

A large recent development comes from the application of Transformer models (Vaswani et al., 2017) [20]: based on sequence-to-sequence architecture, a neural network that transforms any sequence (in this case words) into another sequence. This architecture is then feed in into a Long Short Term Memory (LSTM) network, using heads as a means of capturing attention. Using this architecture, the field had several breakthroughs in the last two years: ELMo (Peters et al., 2018) [11] and BERT (Devlin et al., 2018) [4]. We based our model in BERT for transfer learning, and while BERT is trained on a large corpus of text, mostly from Wikipedia and BooksCorpus; we use a financial contextualized language representation, based on domain-specific language models like BioBERT (Lee et al., 2019) [8]

In this paper we will introduce three models based on transfer learning: FinBERT Prime, pre-trained from scratch on 10-K filings from 2017 to 2019 (436 Million Words) extracted from the SEC; FinBERT-Pre2K, pre-trained from scratch on 10-K filings from 1998 to 1999 (61 Million Words), and FinBERT-Combo, trained using all the fillings, while using BERT-base as an initial checkpoint, as seen in Figure 1.

1.1 Justification

With the release of BERT and ELMO, many new possibilities are opening up in the field of NLP. However, while they are powerful tools to use in many general areas, they will underperform when faced with new technical language that haven't been used in their training (Ettinger, 2019)[5].

> *Our total debt at December 31, 2018 was $5,960.1 million, compared to $5,957.1 million at December 31, 2017, net of the unamortized discount and issuance costs of notes issued under par of $91.1 million and $94.1 million at December 31, 2018 and 2017, respectively. This debt is all denominated in dollars at fixed interest rates, weighed at 5.89%.*
>
> *Fragment from Fin10-K 2019*

As seen above, these financial documents can be extremely technical. More extracts of this type of documents can be found on Appendix 7. 10Ks

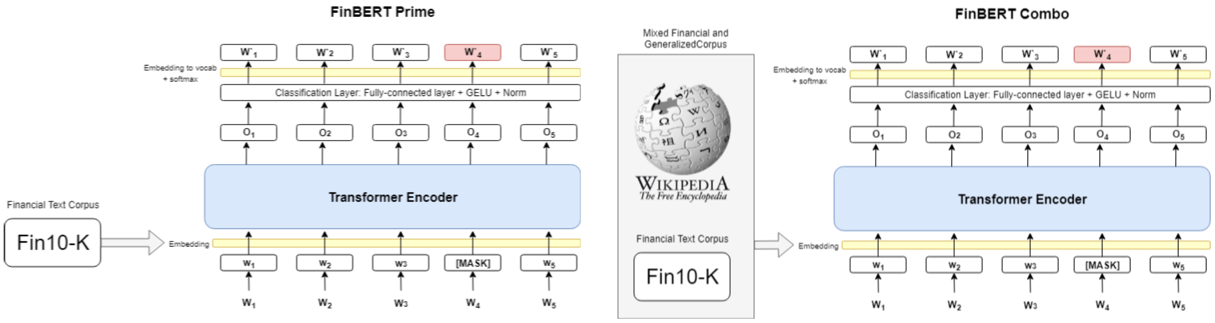


Figure 1. Pre-Trained transformers of FinBERT Prime and Combo

are interesting documents, because apart from all the technical knowledge that they encapsulate, they also have opinionated text in the Management Discussion and Analysis (MD&A) sections. Thus, they are ideal for learning financial contextual information. Another problem, is the scarcity of financial language in datasets commonly used for NLP training tasks. Therefore, generalized language models will fail to capture the essence of financial-domain specific tasks. So our first goal was to extract the data needed from the SEC to train our models, as described on the next subsections. Therefore, we propose the creation of FinBERT as a pre-trained language representation model for the finance domain. Also, given the flexibility of BERT, we pre-trained three models to help us determine: 1) How the financial language has changed over the last two decades and 2) How FinBERT better predicts next sentences and masked word in financial documents relative to generalized BERT. The first model (FinBERT Prime), is trained from scratch using 10K filings from the last three years (2017, 2018, 2019). The second model (FinBERT Pre2K), is trained from scratch using 10K filings from 1998 and 1999. Finally, our last model (FinBERT Combo), is trained on top of BERT-Base Uncased model using both the 10K from the last three years and the 10K from 1998 and 1999. A summary of this can be seen in Table 1.

Table 1. List of Models and their Corpora

Model Name	Corpus	Domain
BERT	Wikipedia + Book Corpus	General
FinBERT Prime	SEC Filings (2017, 2018, 2019)	Business & Finance
FinBERT Pre2K	SEC Filings (1998, 1999)	Business & Finance
FinBERT Combo	Wikipedia + Book Corpus + SEC Fillings	General + Business & Finance

2. Data

In this section we will describe the process to obtain a financial corpus built from 10-K filings. First, we worked in acquiring the data, and then parsed it using HTML, XML, and other techniques. Finally, we will describe the pre-processing needed to make the data compatible to BERT's architecture.

2.1 Data Acquisition

The SEC makes all electronic filings available through Electronic Data Gathering, Analysis, and Retrieval system (EDGAR), their online database and toolset. The user interface and application programming interface are designed to provide access to a single company at a time, not in aggregate as we needed to build our corpora. Therefore we first acquired a list of all publicly traded companies including corporations that have been delisted along with their Central Index Key (CIK) (SEC, 2009) [16]. With that information, we knew what 10-K's should be available. Using the API we can see all the years for which there was an electronic filing for the company, and a list of files in each year, but there is no naming convention, and we wanted only the 10-K, which is a small subset of the files available. There is an index file however which identifies and links to the documents we wanted and using a library (Foundation, 2019) [7] created to parse this information we were able to download every 10-K available. Our available corpus consists of 131,153 10-K's filed by 11,494 separate corporations. To avoid taxing EDGAR, we only ran a single thread to pull data, 900GB in total, and the process took approximately 78 hours to complete.

2.2 Data Cleansing

The filings are, for the most part, stored in extensible business reporting language (XRBL). Inside the XRBL 10-K filings, there are multiple documents, documents in this context can mean several things, plain-text, HTML, XML, Microsoft Excel or images. Our first data cleansing task then was to identify the various documents and separate them in to individual files and remove any binary files as these would not be helpful downstream in our training and analysis. We managed this task using *beatifulsoup* (Richardson, 2019) [14], XRBL allows for markup that is not compatible with either XML or HTML standards, and built-in Python markup parsers struggled with these files, however, lxml (LXML, 2019) [9] is a much more flexible parser and was able construct a DOM from each of the 10-K filings allowing us to separate each document which we stored temporarily in a dictionary for further processing.

These documents contained several tables consisting of primarily numeric data, fortunately, each table, even in the otherwise plain-text documents had *table* tags wrapped around them, and this made it easy to remove all the tables from the DOM's. At this point, we wrote the DOM's out to files excluding any markup in the output. These steps were run using 4 threads on a Google Compute Engine n1-standard-4 instance allowing these parsing and clean-up tasks to complete in just under 4 days.

The remaining documents included, at a minimum, the 10-K for each company each year, and often several supporting documents. An investigation of these other text documents revealed that they were primarily numeric data or links to other documents inside the same XRBL and we concluded that they were not representative of the language we were training and they were not included in our corpora.

The remaining issues with our data at this point were miscellaneous lines containing only page numbers, sections headings, and filenames. These were all resolved by removing any lines with fewer than 1,000 characters as our investigation of this method revealed that all of the unwanted text would be removed, along with a lot of repeated text which was included in the 10-K templates.

With each 10-K filing separated in to cleansed files with the document types identified in their filenames, we then combined each 10-K in to a contiguous file for each year and combined some of those years and created the following training datasets, described in Table 2: The original data from which BERT (Devlin et al., 2018) [4] was trained on, plus three additional corpora: the **SEC2019**, comprised of 10K fillings from the years 2017, 2018, and 2019, covering **4392** companies; and **SEC1999**, comprised of 10K filings from 1998 and 1999, covering **4786** companies and a combination of the three corpora which includes filings from **7272** companies.

Furthermore, to facilitate distributed GPU training and to speed up the [MASK] and next sentence generation process the text data was split into smaller shards. This allowed us to parallelize the pre-training data generation process by utilizing multiple CPU cores which brought down this step from 4-5hours to 30 minutes. The final training examples in TFRecord format consumed approximately 30 GB of space.

2.3 Our Published Data

The time, effort and costs associated with compute and storage create value which we hope others can also benefit. As such we have made our cleaned corpus available to the public through the following URL: `http://people.ischool.berkeley.edu/~khanna/fin10-K`.

Name	Words (M)	Domain
Wikipedia	2,500	General
BookCorpus	800	General
SEC2019	436	Finance
SEC1999	61	Finance

Table 2. List of Corpora for FinBERT & BERT

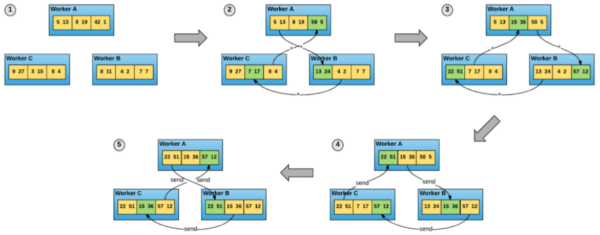


Figure 2. Hovorod distributed learning
Source: Sergeev and Balso, 2018

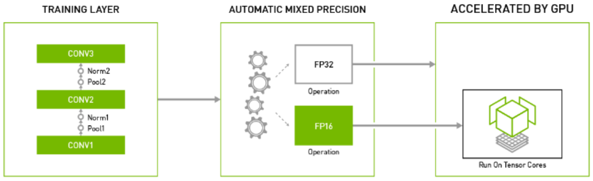


Figure 3. Automatic Mixed Precision Training
Source: NVIDIA, 2019

3. Pre-Training Methods

In terms of structure, and following the path laid down by BioBERT (Lee et al., 2019) [8], we can use the preexisting structure of BERT for training. We discuss our GPU architecture for training, how BERT works under the hood, the process we used for pre-training FinBERT, and, finally, the different performance measures that we used for comparing our models and evaluate the change in the financial language.

3.1 Architecture

We trained FinBERT using two servers each with two 32GB V100 GPUs on the IBM Cloud: one server used for training FinBERT Pre2K and FinBERT Combo, the other used for training FinBERT Prime. The training process was accelerated using Hovorod (Sergeev and Balso, 2018) [17], a distributed deep learning framework on TensorFlow, which allows inter-GPU communication, either inside the same environment or with GPUs on other servers, as seen in Figure 2. Another significant technique used for faster training is the NVIDIA implementation of AMP (Automatic Mixed Precision) (NVIDIA, 2019) [10], which halves the precision of the tensors to FP16 where possible, but maintains the network accuracy, as seen on Figure 3. Using these techniques, we sped up the pre-training of the models to around 2 days each.

3.2 BERT

BERT stands for Bidirectional Encoder Representations from Transformers, and it represents a state-of-the-art solution in language models, based on a self-attention mechanism (Devlin et al., 2018) [4]. The way they achieved the contextualized word representation in a bidirectional form is by masking some percentage of the input tokens and then used the deep network to predict them. In their original work, they masked 15% in each sentence at random, but this is a hyperparameter that can be tuned for other implementations. Later, they combine this task with predicting the next sentence, an innovation over the typical next word prediction, similar to Even-Zohar and Roth, 2000 [6]. This translates into both word and sentence embeddings, which can be used for classification purposes. Thanks to it, BERT can perform many NLP tasks with minimal specific architecture modifications (Lee et al., 2019) [8]

3.3 Pre-training FinBERT

As discussed earlier, our main goal is to create a language model which has most of the domain-specific words commonly used by financial experts, by training on corpora that include contextual information not present in general language models like BERT. The authors of BERT (Devlin et al., 2018) [4] created a GitHub repository [1] that provides the script needed to pre-train a model, either from a

[1] https://github.com/google-research/bert

given checkpoint or from scratch. We need to define four key hyperparameters: 1) Number of steps, 2) Warm-up steps, 3) Batch Size, and 4) Maximum Sequence Length (MSL). The first defines the number of times the optimizer is called to run gradient descent. The second sets the Learning rate of the model being calibrated during the warm-up period, for it to later linearly decrease across the number of steps. The third defines the amount of data per batch, where we want to fit the most data to speed up the training time, constrained by the GPU memory. Finally, the last parameter is the number of tokens that each sentence is truncated to (Adhikari et al., 2019) [1]. In the following sections, we will cover how we pre-trained our three models and the final results that came out of it.

3.3.1 FinBERT Prime

We decided to train FinBERT Prime from scratch: a model that captures specific context on financial documents, but it won't have any of the references from BERT, like pop-culture or history. Following the BioBERT approach (Lee et al., 2019) [8], and having a corpus of 630 million words, we decided to run it first using 250,000 steps (around 6 epochs of data), consistent with previous work in DocBERT (Adhikari et al., 2019) [1]; with a warm-up period of 10,000 steps, to give time to the Learning Rate to adapt to our data. Using the original architecture of BERT (Devlin et al., 2018) [4], we have a Learning Rate Schedule where it starts at 1e-4, to then ramp up over the warm-up period, and then a linear decay for the remainder of the training. For the MSL, we calculated the average sentence size of SEC2019 (20 words) and only 6% of the sentences where greater than 64 words. Thus, to speed up the training and based on our corpus, we decided to use a MSL of 128, which translates to a batch size of 96 (because we are using Hovorod. the effective batch size is 192). Another consideration to pre-train using a MSL of 128, is that according to Devlin et al., 2018, even though though the fully-connected/convolutional cost is the same for any MSL, the attention head costs increase quadractically. Our initial results were promising: ∼78% of Masked LM accuracy (MLM) and ∼98% of Next Sentence Accuracy (NS), after 1 day of training.

However, we decided that given we were starting the pre-training from scratch, our model would benefit from increasing the number of steps to half of what Devlin et al., 2018 did on the original BERT (1,000,000 steps). Thanks to this, we increase our accuracies to ∼81% and ∼99% respectively. Following Devlin et al., 2018 advice, we decided to add 200,000 steps with a 512 MSL: These long sentences are needed to learn positional embeddings, which can be learned quickly. This brought down our batch size to 16 (effective 32). Our final set of hyperparameters and results are summarized on Table 3 and Table 4 respectively.

3.3.2 FinBERT Pre2K

To compare how financial language has change over the last two decades, we also trained a FinBERT Pre2K with 10K filings from 1998 and 1999 (SEC1999 dataset). This model was created for comparison of the two periods, so we decided to train once: 250,000 steps with a 10,000 warm-up period, using a MSL of 128. We also used a batch size of 96. As before, all the results are summarized in Table 4

3.3.3 FinBERT Combo

Finally, to take advantage of transfer learning from the original BERT, we chose to train a Combo Model on top of the last checkpoint of BERT-Base Uncased (Devlin et al., 2018). This training was done in parallel with FinBERT Prime, using SEC2019 for the first 250,000 and using SEC1999 for the last 250,000. After the results from FinBERT Prime, we decided to only add 100,000 steps with a MSL of 512. Because of the average size of the sentences, when we ran an MSL of 128, we combined 2 to 3 sentences, enough to get positional embeddings and context. However, when we extended to a MSL of 512, the number of sentences combined grew to 8 to 12 sentences, enough for the context to be lost: perhaps changing to a different paragraph or a new section of the document. The reasoning behind creating the Combo model is to have a model which can also be applied to more general news or corpus that, while still heavy on domain-specific words and context, is written more for mass consumption. As usual, all the results are summarized on Table 4.

Model	MSL	Batch Size	# Steps	Warmup
FinBERT-Prime	128+512	96+16	500K + 200K	50K + 10K
FinBERT Pre2K	128	96	250K	50K
FinBERT Combo	128+512	96+16	500K + 100K	50K + 10K

Table 3. Pre-Train Hyperparameters

3.4 Measuring Performance

To measure the performance of our models, and according to the goals defined at the beginning, we accomplish the following tasks:

1. Using a new dataset to perform Masked LM and Next Sentence predictions with our three models and BERT, to establish comparisons between them. We used the 10-Q filings, which are quarterly reports that companies are also required to file under the SEC. These documents are similar to the 10-K, however were not included in our training sets, thus makes the for an ideal test set of unseen data. The other dataset used for evaluation consists of Earnings Call transcripts: where public companies discuss the results of financial results of a given period (Chen, 2019)[3]

2. To compare how the financial language has change between 1999 and 2019, we ran analogies from the embeddings of FinBERT Prime and FinBERT Pre2K, and determine their cosine similarity. We selected three particular words to do it.

Model	Metric	128MSL		512MSL	
		250K	500K	100K	200K
FinBERT-Prime	MLM	78.07%	81.30%	79.37%	76.14%
	NSP	97.88%	99.13%	98.38%	97.50%
	Loss	1.021	0.820	0.913	1.112
FinBERT-Pre2K	MLM	84.67%			
	NSP	100.00%			
	Loss	0.594			
FinBERT-Combo	MLM	83.16%	87.16%	80.42%	
	NSP	98.88%	100.00%	98.13%	
	Loss	0.729	0.497	0.870	
Global Step		250K	500K	600K	700K

Table 4. Pre-Trained FinBERT models results

4. Results and Discussion

4.1 Pre-Trained Results

In Table 5, we show our evaluation results using only 10K filings of 2019 that we didn't use for training. We also ran BERT on the same dataset to compare the performance of all the models.

Our models beat BERT in all metrics, again supporting the concept that some domain-specific language can't be spotted in general language models. It is expected that FinBERT Prime had the best accuracies, as it is the closest to the evaluation set.

Model	MLM	NSP	Loss
FinBERT-Prime	80.17%	98.50%	0.87
FinBERT-Pre2K	77.20%	91.88%	2.06
FinBERT-Combo	77.20%	90.63%	1.35
BERT	51.16%	62.38%	5.379

Table 5. Pre-Trained Evaluation on 2019

Another important result is how FinBERT Combo has similar results as FinBERT Pre2K, which means that an important amount of context from the 1999 period was transferred to Combo. Finally, although

Pre2K beats BERT succinctly, we can argue that changes in the financial language leads to a loss in accuracy compared to Prime.

4.2 Tests on New Data

As previously explained, we ran our models against two new datasets: 10-Qs from the last year and Earning Calls. All results are summarized on Table 6

Dataset	Model	MLM	NSP
10-Q	FinBERT-Prime	77.52%	94.50%
	FinBERT-Pre2K	70.33%	93.00%
	FinBERT-Combo	75.33%	94.38%
	BERT	51.18%	60.88%
Earnings Call	FinBERT-Prime	42.81%	53.13%
	FinBERT-Pre2K	38.44%	51.63%
	FinBERT-Combo	45.81%	56.38%
	BERT	46.87%	29.88%

Table 6. Test Results on 10-Q's and Earning Calls

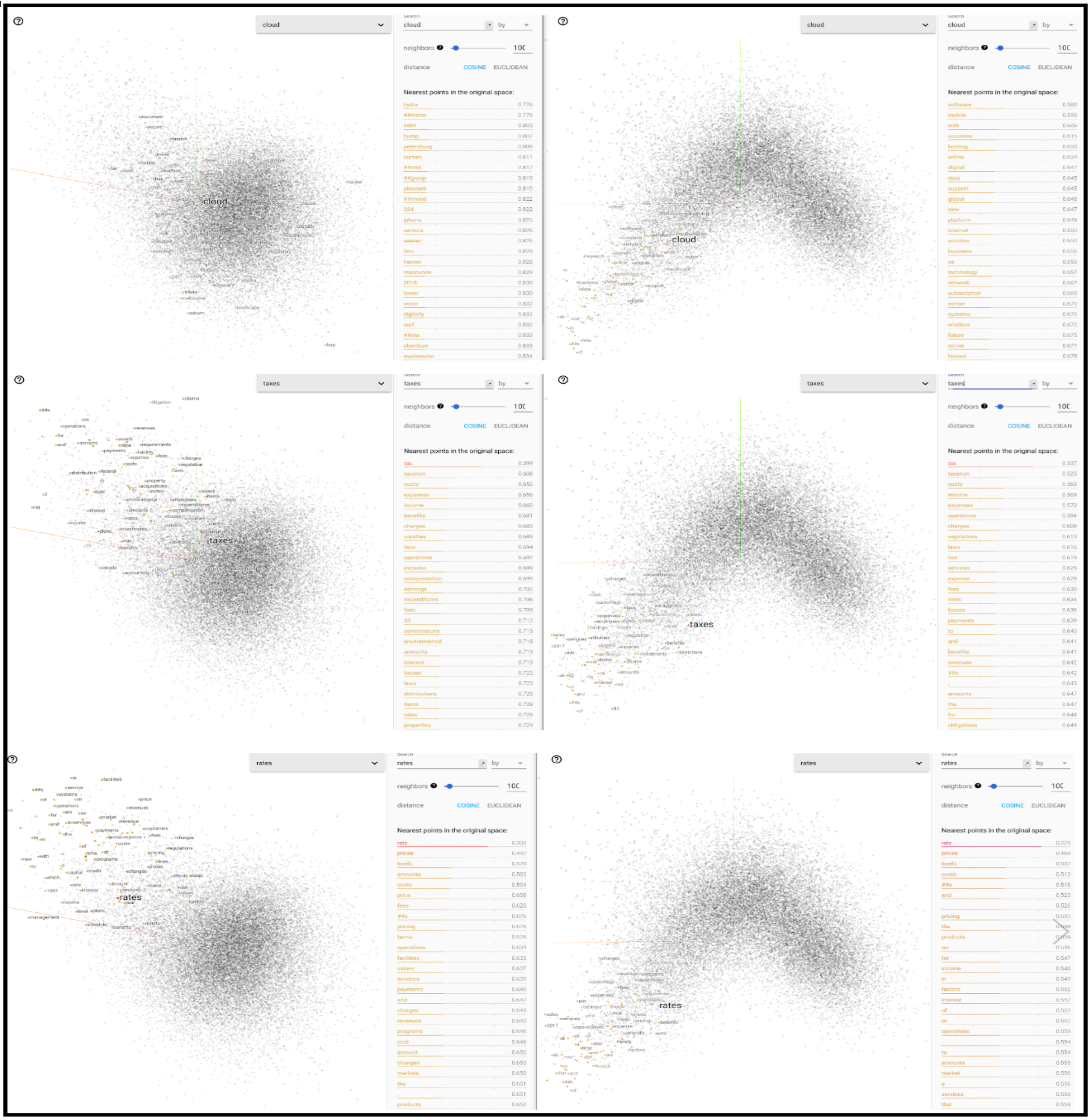
On the 10-Q's dataset, our models vastly outperform BERT in any prediction task: we have an increase of 30% on Next Sentence Predictions and around a 25% increase on Masked LM Prediction. On the other hand, on the Earning Calls, which is a mixture of common language between parties and financial jargon, BERT beats our models on Masked LM prediction, which means it has better contextualization - this is related to the amount of general context that exist in the informal part of the conversation. However, our models outperform BERT on next sentence predictions, which are more related to the financial context of the call: they have to review each of the subsections on the 10-K. Finally, FinBERT Combo seems to be the best model to use in this circumstance: it has the transferred general context from BERT, plus the learned financial context of our datasets.

4.3 Time changes in the financial language: FinBERT Prime vs. Pre2K

Thanks to TensorBoard (Tensorflow, 2019) [19], we can visualize the word embeddings on any of our FinBERT models. In Figure 4 we can see the nearest words for three selected words: *cloud*, *taxes*, and *rates*. For *cloud* we can observe the big change in the last two decades: from *hydra* to *software*. Besides the clear differences of domain-space, words like business, subscriptions, etc. appears in 2019, and are clearly missing from 1999. For *taxes*, we can also spot the change between the two environments: while many of the nearest words are similar (expenses, costs, etc.), in 2019 the word income and benefits come closer in context than in 1999, a behavior possibly explained by the new tax cuts from the Trump's administration (Tax Cuts and Jobs Act, TCJA), which propelled many new financial benefits for businesses. Finally, for *rates*, we also can observe the similarities in vocabulary and terms of financial words (products, markets, pricing, etc.), expected from financial domain-specific training.

5. Future Work

Although financial datasets with an NLP approach are sparse, we found one called FiQA, and Open Challenge that ran from April 23rd to the 27th in Lyon France (Project, 2018)[13]. This dataset consists in Questions and Answers crawled from Stackexchange, Reddit, and StockTwits, as well as Sentiment analysis on Headlines and Posts. As future work, we will continue to train on top of FinBERT Combo with news and posts for financial sites: This way we capture both the dryness and technical jargon of the 10Ks with the analysts' know-how and opinionated context. This will allows to build a one-size-fits-all model for financial sentiment analysis of financial Analyses, which can be used on Earnings Calls and many other tasks.



Search: cloud | by
neighbors 100
distance COSINE EUCLIDEAN

Nearest points in the original space:

hydra	0.776
##mmer	0.779
eden	0.803
bump	0.807
petersburg	0.808
curtain	0.811
##rent	0.812
##group	0.819
pleasant	0.819
##mond	0.822
324	0.822
iphone	0.825
ventura	0.826
seeker	0.826
fern	0.828
hacker	0.828
resonance	0.829
2018	0.830
tower	0.830
victor	0.832
digitally	0.832
surf	0.832
##eta	0.833
aberdeen	0.833
wastewater	0.834

Search: cloud | by
neighbors 100
distance COSINE EUCLIDEAN

Nearest points in the original space:

software	0.580
mobile	0.590
web	0.606
solutions	0.615
hosting	0.633
online	0.634
digital	0.642
data	0.645
support	0.645
global	0.646
new	0.647
platform	0.649
internet	0.650
solution	0.650
business	0.656
sa	0.656
technology	0.657
network	0.667
subscription	0.669
center	0.670
systems	0.670
wireless	0.672
future	0.675
server	0.677
hosted	0.678

Search: taxes | by
neighbors 100
distance COSINE EUCLIDEAN

Nearest points in the original space:

tax	0.399
taxation	0.608
costs	0.652
expenses	0.656
income	0.660
benefits	0.681
charges	0.683
royalties	0.689
taxa	0.694
operations	0.697
expense	0.699
compensation	0.699
earnings	0.702
expenditures	0.706
fees	0.709
33	0.713
commissions	0.715
environmental	0.718
amounts	0.719
interest	0.719
losses	0.722
laws	0.725
distributions	0.728
items	0.728
rates	0.729
properties	0.729

Search: taxes | by
neighbors 100
distance COSINE EUCLIDEAN

Nearest points in the original space:

tax	0.337
taxation	0.525
costs	0.563
income	0.569
expenses	0.570
operations	0.594
charges	0.609
regulations	0.613
laws	0.616
taxi	0.619
services	0.625
expense	0.629
fees	0.630
rates	0.634
losses	0.636
payments	0.639
to	0.640
and	0.641
benefits	0.641
revenues	0.642
##s	0.642
,	0.645
amounts	0.647
the	0.647
for	0.648
obligations	0.649

Search: rates | by
neighbors 100
distance COSINE EUCLIDEAN

Nearest points in the original space:

rate	0.300
prices	0.492
levels	0.579
amounts	0.583
costs	0.594
price	0.608
fees	0.620
##s	0.626
pricing	0.626
terms	0.628
operations	0.634
facilities	0.635
values	0.637
services	0.639
payments	0.640
and	0.642
charges	0.643
revenues	0.643
programs	0.646
cost	0.646
amount	0.650
changes	0.650
markets	0.650
the	0.651
,	0.651
products	0.652

Search: rates | by
neighbors 100
distance COSINE EUCLIDEAN

Nearest points in the original space:

rate	0.225
prices	0.468
levels	0.502
costs	0.513
##s	0.518
and	0.523
,	0.526
pricing	0.535
the	0.538
products	0.539
on	0.546
for	0.547
income	0.548
in	0.549
factors	0.552
interest	0.552
of	0.552
or	0.552
operations	0.553
.	0.554
to	0.554
amounts	0.555
market	0.556
a	0.556
services	0.556
that	0.558

Figure 4. Nearest words of *cloud*, *taxes* and *rates* in 2019 and in 1999

6. Conclusion

In this paper, we introduced 3 FinBERT models, trained on different 10-K filings, to use in financial text mining and context changes over time. While BERT was built for general purpose language understanding, our models learn from technical financial language, and in the case of FinBERT Combo, BERT knowledge is transferred to it. We also proved that even in new financial datasets, FinBERT outperforms BERT in masked LM and Next Sentence Prediction. Finally, using TensorBoard for visualization, we were able to show selected changes from two decades brings contextual information from specific financial terms.

7. Appendices

Appendix A: Extract from financial documents

All appendices are extracted from the EDGAR database of 10K fillings posted in 2019.

1. *The 2017 U.S. tax reform introduced a one-time transition tax that is based upon the Company's total accumulated post-1986 prescribed foreign earnings and profits ("EP") estimated to be $8.9 billion, the majority of which was previously considered to be indefinitely reinvested and accordingly, no U.S. federal and state income taxes were provided. Upon enactment of the 2017 U.S. tax reform, the Company calculated and recorded a provisional tax amount of $181.1 million as a reasonable estimate for the one-time transition tax that was fully offset by foreign tax credits. During 2018, the Company determined that this amount should be reduced by $28 million and finalized its transition tax at $153.1 million, which was fully offset by foreign tax credits. Earnings of the Company's Peruvian branch are not subject to transition taxes since they are taxed in the United States on a current basis.*

2. *Share repurchase program: In 2008, our Board of Directors ("BOD") authorized a $500 million share repurchase program that has since been increased by the BOD and is currently authorized to $3 billion. Since the inception of the program through December 31, 2018, we have purchased 119.5 million shares of our common stock at a cost of $2.9 billion. These shares are available for general corporate purposes. We may purchase additional shares of our common stock from time to time based on market conditions and other factors. This repurchase program has no expiration date and may be modified or discontinued at any time. For further details please see Item 5 "Market for Registrant's Common Equity, Related Stockholder Matters and Issuer Purchases of Equity Securities SCC common stock repurchase plan." Dividend: On January 24, 2019, the BOD authorized a dividend of $0.40 per share paid on February 26, 2019, to shareholders of record at the close of business on February 12, 2019. In addition, as part of the settlement of claims brought on behalf of the Company and its shareholders against Grupo Mexico, AMC and certain current and former directors (together with Grupo Mexico and AMC, the "Defendants") a dividend of $0.44428 per share was payable on February 21,2019 to shareholders of record at the close of business on February 11, 2019, other than the Defendants. The settlement dividend, totaling $36.5 million is an obligation of Grupo Mexico and AMC and therefore, have been funded by them.*

References

[1] Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. Docbert: BERT for document classification. *CoRR*, abs/1904.08398, 2019. URL `http://arxiv.org/abs/1904.08398`.

[2] Stephen Balaban. Deep learning and face recognition: the state of the art. *CoRR*, abs/1902.03524, 2019. URL `http://arxiv.org/abs/1902.03524`.

[3] James Chen. Earnings call, Jul 2019. URL `https://www.investopedia.com/terms/e/earnings-call.asp`.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL `http://arxiv.org/abs/1810.04805`.

[5] Allyson Ettinger. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *arXiv e-prints*, art. arXiv:1907.13528, Jul 2019.

[6] Yair Even-Zohar and Dan Roth. A classification approach to word prediction. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, NAACL 2000, pages 124–131, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.

URL http://dl.acm.org/citation.cfm?id=974305.974322.

[7] Python Software Foundation. sec-edgar-downloader, Jun 2019. URL https://pypi.org/project/sec-edgar-downloader/.

[8] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *CoRR*, abs/1901.08746, 2019. URL http://arxiv.org/abs/1901.08746.

[9] LXML, Jun 2019. URL https://lxml.de/parsing.html.

[10] NVIDIA. Automatic mixed precision for deep learning, Jun 2019. URL https://developer.nvidia.com/automatic-mixed-precision.

[11] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[12] Michael A Peters. Deep learning, education and the final stage of automation. Jul 2017. URL https://www.researchgate.net/publication/318705811_Deep_learning_education_and_the_final_stage_of_automation.

[13] SSIX Horizon 2020 Project. Fiqa - 2018, Feb 2018. URL https://sites.google.com/view/fiqa/home.

[14] Leonard Richardson. Beautiful soup, 2019. URL https://www.crummy.com/software/BeautifulSoup.

[15] SEC. Form 10-k, Jun 2009. URL https://www.sec.gov/fast-answers/answers-form10khtm.html.

[16] SEC. Edgar company filings: Cik lookup, Dec 2009. URL https://www.sec.gov/edgar/searchedgar/cik.htm.

[17] Alexander Sergeev and Mike Del Balso. Horovod: fast and easy distributed deep learning in tensorflow. *CoRR*, abs/1802.05799, 2018. URL http://arxiv.org/abs/1802.05799.

[18] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. doi: 10.1109/iccv.2017.97.

[19] Tensorflow. Tensorboard: Visualizing learning : Tensorflow core : Tensorflow, 2019. URL https://www.tensorflow.org/guide/summaries_and_tensorboard.

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. pages 5998–6008, 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.