Extracting sentiment from financial statements using neural networks

June 4, 2018

Murat Aydogdu

Department of Economics and Finance
Rhode Island College
Alger Hall 237
Providence, RI 02908
maydogdu@ric.edu

Abstract

Using a small sample of sentences, I test the performances of three models in predicting sentence sentiment (positive, neutral, or negative): a simple model based on a financial word list, a simple neural network model based on vector representations of words, and a sentence-based neural network model (Long Short Term Memory, LSTM). The simple neural network outperforms the word list based model in terms of prediction accuracy and similar metrics. While the LSTM model doesn't perform any better than the simple neural network, it is more likely to gain from a larger dataset and model calibration. My work contributes to the relatively small but growing literature of applying machine learning (especially neural networks) to the finance domain.

Preliminary and incomplete: please do not quote without permission.

I categorize 1,000 sentences randomly extracted from financial statements (specifically, the Management's Discussion and Analysis of Results of Operations and Financial Condition (MD&A) sections of 10-K filings from Security and Exchange Commission's Edgar database) into three sentiment categories: Negative (0), Neutral (1) and Positive (2). I use this dataset to build three models and use these models for sentiment prediction. The first model is a straightforward model based on Loughran and McDonald (2011) financial word list. The second model is a simple word-based neural network that relies on a more complex vector-based word representation (GloVe, explained below). The third model is also a neural network but it is a more complex model that precesses sequences, in this case, whole sentences (Long Short Term Memory, LSTM, model based on GloVe word representation). I find that neural networks (machine learning models) perform better in predicting sentence sentiment than the simple word list based model. With the given, admittedly small, dataset, the two networks' performances are about the same, although a larger dataset plus model calibration is likely to benefit the sentence-based model.

1 Background

My work is an exercise in using machine learning models to predict sentiment from financial statements. The goal is to develop robust models that analyze sentences in finance context and assign a sentiment (positive / neutral / negative) to them. These models can then be used for many purposes. For example, they can be used to assign an aggregate sentiment to a financial statement. The predictive power of those sentiment indicators could be an avenue to investigate (as has been done in prior research). Correlations between statements (cross-sectional or time-series) could be another avenue.

Sentiment analysis in finance is not a new field. There are a number of papers that survey this line of research: Kearney and Liu (2014), Li (2010a), and Loughran and McDonald (2016) are three such papers. Kearney and Liu (2014) provide a taxonomy of papers along two dimensions: information source (corporation-expressed sentiment, media-expressed sentiment, and internet-expressed sentiment), and methods of content analysis (dictionary-based / word list approach and machine learning). In my work, I use 10-K filings (i.e., corporation-expressed sentiment) and three models that span both methods of content analysis (i.e., word list approach and machine learning approach. The details of my analysis approach are in the next section.

Over the next few paragraphs, I present a brief overview of Kearney and Liu (2014)to put my work in context. Annual and quarterly financial statements and earnings press releases are some of the corporate disclosures that are studied in literature. Media-expressed sentiment use sources like the Wall Street Journal, analyst reports, and Thomson Reuters NewsScope service. And internet expressed sentiment papers look at sources like Yahoo message boards.

Corporate disclosures have the advantage of reflecting insiders' opinions and thus are very valuable. But they do not necessarily tell us everything the managers may know or they may be inaccurate in cases where managers try and manipulate public opinion regarding the company. Still these statements also are at least partially forward-looking and probably more reliable than, for instance, internet posts because of corporate regulations in place. On the negative side, financial statements are not issued very frequently and may not capture current news as well as other sources do.

Dictionaries and wordlists are used to categorize individual words into groups. The Loughran McDonald (2011)Loughran & McDonald (2011) list has become popular in recent years as it is domain-specific (i.e., it is based on financial statements as opposed to a general corpus. One drawback of these dictionaries and word lists are their focus on individual words, or "bag-of-words" approach (as is known in natural language processing literature). By splitting a sentence into a list of words, they lose context. Studies using dictionaries and word lists also consider whether they will put equal weight on each word or use a different weighing scheme (for instance weighing each word inversely proportional to its document frequency).

And finally, on the machine learning side, these studies use various models to engage in "supervised learning". This kind of study starts by "labelling" data: for instance messages labelled as positive / neutral / negative. Then, they split data into training and test sets and use the training set to "train" or build a model, for instance a Support Vector Machine model or a

Random Forests model. Training phase establishes model parameters. Then, test set is used to assess model performance. Once a model works reasonably well, it can be used to make predictions on new data.

Li (2010b) is one of the earlier works that uses machine learning and corporation-expressed sentiment on a large scale. It is based on sentences selected from the MD&A sections of more than 140,000 10-K and 10-Q from annual reports that span 1994 - 2007. This project first selects 30,000 forward looking statements, then labels these sentences as having positive / neutral / negative/ uncertain sentiment (or "tone"). Using these labelled sentences, Li (2010) trains a Naive Bayesian model which performs reasonably well on new data (about 60% test set accuracy). This model is then used to predict the sentiment for about 13 million sentences from the 140,000 filings. Sentence sentiments are averaged over a filing to obtain a sentiment measure for the filing. On of the main findings from this study is that the average sentiment of a financial statement predicts future earnings and liquidity.

2 Analysis framework and data

My data set is based on 10-K filings available from SEC's EDGAR database. Starting with a list of Standard and Poor's 500 Index, or SP500, constituent companies, I scrape EDGAR's website to obtain HTML filings from these companies. From those files, I extract the text of Item 7 (MD&A) portion of the filings. It is fairly common in sentiment analysis literature to focus on this section of filings.[1] The text of each filing, its sentences and words form the building blocks of my analysis.

I use three models to extract sentiment from 10-K sentences: two of them are word-based and one of them is sentence-based. My main focus is on the sentence-based LSTM neural network model (**NN model**), detailed below. I benchmark the performance of this model using two other models: the **LM model**, a word-based model that uses the Loughran and McDonald (2011) word list and the **GV model**, another word based model that relies on vector representations of words.

In this paper the unit of analysis is a sentence. It is further split into either words (LM Model) or "tokens", a combination of words and punctuation marks (GV and NN Models). I use words and tokens interchangeably for the rest of the paper.

2.1 LM Model

This model splits a sentence into words, and counts the number of words that appear in the Loughran and McDonald (2011) word lists as positive words (Pos) or negative words (Neg). It then constructs a scaled measure as: $(Pos - Neg)/(Pos + Neg)$. I use equal weighing when aggregating words in a sentence. This metric is easy to interpret: its value is between -1 and +1, with positive readings reflecting positive sentiment and negative readings reflecting negative sentiment. When I transform this metric into Neutral / Negative / Positive sentiment, I use $\pm 10\%$ cutoffs: any value above (below) 10% (-10%) is labelled Positive, coded as 2 (Negative, coded as 0). This model is pretty straightforward to use and it doesn't require a "training" or model building step.

The negative wordlist has 2355 words and positive wordlist has 354 words. Loughran and McDonald (2011) show that their dictionary captures sentiment better than a more generic but commonly used dictionary, the Harvard Psychosociological Dictionary. This is because words like as tax, cost, capital, board, liability are in the negative words list in Harvard dictionary but do not necessarily imply negativity in a financial context.

One drawback of this model is that a majority of words that will appear in 10-Ks are not in the positive or negative list. Thus, most words in a sentence do not contribute to its sentiment. Also, Loughran and McDonald (2011) state that many studies do not find much value in the positive words list, "likely attributable to their frequent negation." This may justify models that operate on full sentences rather than treat them as "bag of words".

[1]Loughran and McDonald (2011) find that the MD&A does not necessarily provide a clearer picture of the company than the whole filing does.

2.2 GV Model

The second benchmark model is also a word based model. It starts by splitting each sentence into tokens. Next, it converts each token into a 50-dimensional vector and computes an aggregate vector representation for the sentence by averaging token vectors. Finally, it runs this input through a simple neural network to predict a sentiment indicator. This model, in Kearney and Liu (2014) taxonomy, is a machine learning model.

2.2.1 GloVe: Global Vectors for Word Representation

For two of the three sentiment analysis models (NN and GV), I use a vector representation of words, GloVe (Pennington et al, 2014). This is an unsupervised learning algorithm that maps words into dimensions. This representation can then be used, for instance, to identify similar words. The difference between the dimensions of two similar words will be roughly a vector of zeros.

The representation I use in this paper was pretrained from the 6B token (total number of units) corpus of Wikipedia 2014 and Gigaword 5.[2] It is a vocabulary of 400K tokens (unique units) and has 50 dimensions. When I use vector representations, I first split a filing into sentences, then tokenize these sentences. Then, I obtain the vector representation of each word.

An advantage of this model over the LM model is its depth. First, it is a 400,000-word dictionary (as opposed to the less than 3,000 positive and negative words in the LM Model). It captures most of the words (and punctuation) in a filing so it has a more thorough word coverage. Next, instead of just looking at positive and negative (and by default, neutral words) it captures 50 dimensions, so it is more nuanced. A disadvantage could be that the word vectors are trained using a general corpus, not a finance-specific set of documents.

2.2.2 GV Model Neural Network

I use a simple neural network with no hidden layers for this model (python + keras with TensorFlow backend). The vectors of each token of a sentence are averaged first. The network accepts the averaged sentence vector as the input (50 dimensions) and runs the input through softmax activation to produce class probabilities. I train this network for 200 epochs using a minibatch size of 32, Adam optimizer with its default parameters (learning rate = 0.001, beta_1 = 0.9, beta_2 = 0.999), and the cross entropy loss function.

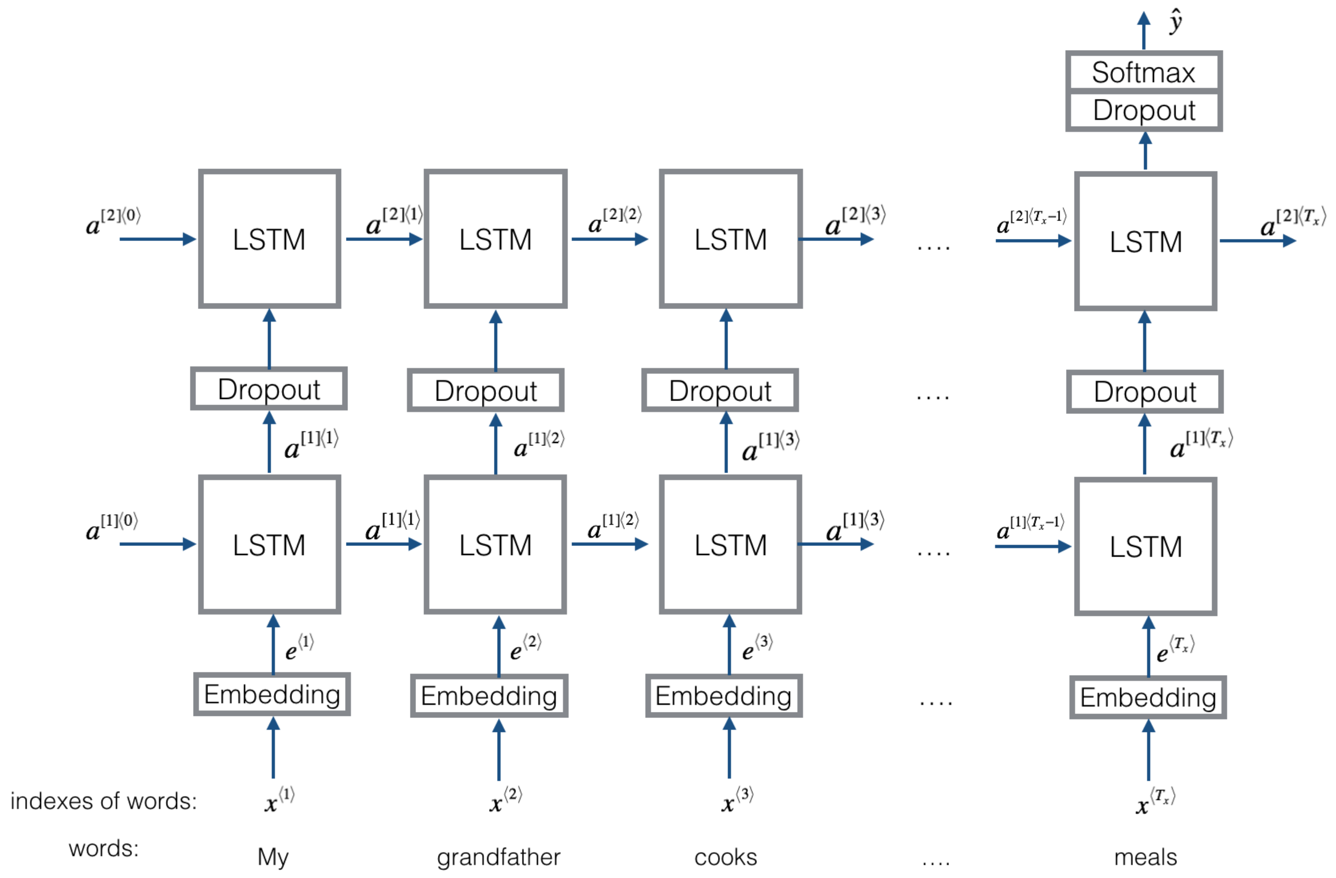
2.3 NN Model

The main model I use in this paper is an example of sequence models. These models have successfully been used in fields like language translation. The model accepts a sequence of data (a sentence), and produces an output, in this case, a sentiment indicator (Positive / Neutral / Negative). Since the model works on a sentence, it takes the context of words in consideration. In other words, it doesn't process each word in isolation. This model, in Kearney and Liu (2014) taxonomy, is a machine learning model.

[2]https://nlp.stanford.edu/projects/glove/

Figure 1: Precision and Recall over Test Period

This figure shows the LSTM network used in the NN Model. The input is a sentence whose words are converted into GloVe vectors (embeddings). The image is from coursera's Sequence Models course assignment (Emojifier).


$\hat{y}$
Softmax
Dropout
$a^{[2]\langle 0\rangle}$
LSTM
$a^{[2]\langle 1\rangle}$
$a^{[2]\langle 2\rangle}$
$a^{[2]\langle 3\rangle}$
$a^{[2]\langle T_x-1\rangle}$
$a^{[2]\langle T_x\rangle}$
Dropout
$a^{[1]\langle 1\rangle}$
$a^{[1]\langle 2\rangle}$
$a^{[1]\langle 3\rangle}$
$a^{[1]\langle T_x\rangle}$
$a^{[1]\langle 0\rangle}$
$a^{[1]\langle T_x-1\rangle}$
$e^{\langle 1\rangle}$
$e^{\langle 2\rangle}$
$e^{\langle 3\rangle}$
$e^{\langle T_x\rangle}$
Embedding
indexes of words:
$x^{\langle 1\rangle}$
$x^{\langle 2\rangle}$
$x^{\langle 3\rangle}$
$x^{\langle T_x\rangle}$
words:
My
grandfather
cooks
meals

2.3.1 NN Model Neural Network: LSTM Network

I provide a cursory review of NN model's structure here. Deep Learning by Goodfellow et al (2016)Goodfellow *et al.* (2016) is the standard textbook on deep learning, and has more detail. An LSTM (Long Short Term Memory) network is a Recurrent Neural Network (RNN) that has LSTM units as layers. RNNs are different from the more commonly used and simpler networks called feedforward networks. In feedforward networks, data flows from inputs to outputs in a single direction. RNNs allow outputs from one layer to be fed back into that layer, thus forming a ”cyclic graph”. This structure allows parameter sharing across the sequence.

I use the LSTM network shown in Figure 1which accepts a sentence as input. More specifically, the tokens of the sentence are converted into GloVe vectors first (embeddings). Those tokens pass trough two layers of LSTM cells with dropout used for regularization. The result then passes through a softmax activation function which outputs probabilities of three classes. The number of tokens in a sentence is set to 55 and sentences with fewer tokens are padded with 0s so that sequence length is fixed. This is a deep network since it has more than one hidden layers.

To train the NN network, I use Adam optimizer with its default parameters (learning rate = 0.001, beta_1 = 0.9, beta_2 =

0.999) and cross entropy loss function. I train the network for 30 epochs using a minibatch size of 32. Note that for this version of the paper, I have not performed model calibration.

2.4 Data set

I construct a dataset based on the Standard and Poor's 500 Index. At the beginning of May 505 CIKs[3] were listed in Wikipedia's "List of S&P 500 companies" article.[4]I then query SEC's Edgar database to obtain a list of all the 10-K filings for all the S&P500 CIKs; this query yields a list of 9,254 filings with filing dates ranging from 1/06/1994 to 5/15/2018. Next, I try to find the Item 7 (MD&A) in the HTML file by searching through this filing list. If it is found, I extract it into an HTML file. This step results in 4,143 Item 7 HTML files.

The next step in data set construction is extracting text from the Item 7 HTML file. I remove the tables from the filing and then split (tokenize) the remaining text into sentences. This step results in 4,026 text files.[5] Then, I randomly select 20% of the filings to construct a sentence-based dataset. From this random subset of filings, I obtain a list of tokenized sentences. For instance, "I am (not) sure this isn't right." gets tokenized as 'I', 'am', '(', 'not', ')', 'sure', 'this', 'is', "n't", 'right', '.'. Following this step, I filter out filings that have less than 2,500 tokens.[6]For the remaining documents, I randomly select 5% of the sentences, then make sure the selected sentences have at least 10 tokens and at most 50 tokens. This step eliminates many of the titles and subtitles (as well as very short sentences) and also very long sentences which may potentially be split incorrectly (for instance, two sentences wrongly processed as a single sentence). I record the remaining 13,465 sentences after shuffling them so that sentences from a filing are not in sequence.

2.5 Assigning sentiment (Positive / Neutral / Negative) to sentences

The goal is to label sentences as having positive, neutral, or negative sentiment. This is a subjective process as my views on what constitutes a positive sentence my differ from others'. While there may be agreement on some sentences, some sentences may be too ambiguous or controversial. Also, it is not easy to define what, exactly, constitutes positive sentiment. In some cases, a single sentence is not enough to put it in context. Overall, this is not a very straightforward process and will likely result in noisy measurement. With these caveats, I label 1,000 sentences as negative (193 sentences, coded as 0), neutral (479 sentences, coded as 1) and positive (328 sentences, coded as 2). While labelling sentences, I make sure I have a reasonably large number of observations in each class. This is important as there are more neutral sentences than positive or negative sentences.

Positive sentences are those that may have a positive impact on the company's future prospects. These are often forward looking but may sometimes refer to events that took place in the past. They often include words like increase in a positive way (sales increased but not debt increased).

- No impairment charges were recorded during the years ended December 31, 2011, 2010 and 2009.
- Base revenues increased 7.5% in 2011 versus 2010 as worldwide economic conditions strengthened.
- These expense increases were driven primarily by our business growth.
- Additionally, the Board of Director's approved a plan to increase the 2017 annual cash dividend to $1.38 , which is intended to paid in four quarterly installments.

[3]Central Index Key used to identify companies and individuals on Edgar.
[4]https://en.wikipedia.org/wiki/List_of_S%26P_500_companies accessed May 1, 2018
[5]Not all Item 7s have MD&A. See for instance https://www.sec.gov/Archives/edgar/data/4977/000000497704000030/afl10k03.htm#Item7
[6]Loughran and McDonald (2011) use 2,000 words as threshold.

- 2003 vs. 2002 NGL revenues increased $132 million in 2003.

Neutral sentences are often declarations or descriptions. They may clarify things but often do not make any positive or negative judgements. Sometimes they have a positive claim offset by a negative claim.

- Operational Risk Management is overseen by our Chief Operational Risk Officer, who reports directly to the CRO.
- In response to the declining economic conditions, the Federal Reserve lowered the Federal Funds rate eleven times since January 2001 to 1.75%, which represents a cumulative 475 basis point reduction.
- Each option was exercisable during the period starting from January 1, 2010, and ending on December 31, 2010.
- Both methods require the use of estimates which in turn contain judgments and assumptions regarding future trends and events.
- For our interest rate swaps and caps, the table presents the notional amount of the swaps and caps and the years in which they expire.

Negative sentences have fairly clear indications of things that may affect the company negatively. Some of these are also forward looking and some of them refer to events that took place in the past.

- The frequency and sophistication of such fraud attempts continue to increase.
- Other, net expense totaled $169 million during the year ended December 31, 2008, an increase of $113 million compared to the same period in 2007.
- The increase in 2015 was primarily due to lower investment earnings of $11 million and higher interest expense of $8 million.
- Cost of sales increased by $44.5 million, or 10.9%, to $453.9 million in 2016 from $409.4 million in 2015.
- These downgrades negatively affect our access to the capital markets and increase our cost of capital.

3 Results

In this section I will detail the performances of the three models based on my preliminary work using a dataset of 1,000 labelled sentences. I train each of these models 100 times by first randomly splitting the dataset into a training set of 700 sentences and a test set of 300 sentences and then recording model performances on both training and test sets. Below, I report main performance metrics resulting from these trials. Note that there is no training for the LM model.

3.1 f1-scores

In order to compare model performances, I compute and analyze three commonly used metrics: precision, recall, and f1-score. They are computed based on the counts of true positives (TP: 1 predicted as 1, 0 predicted as 0, etc), false negatives (FN: 1 predicted as 0 or 2, etc), false positives (FP: 0 or 2 predicted as 1, etc). Precision is defined as predictions of an outcome that are actually correct: $(TP/(TP+FP))$. Recall is defined as predictions of an outcome that are predicted as such: $(TP/(TP+FN))$.

Table 1: f1-scores

This table reports aggregate f1-scores. They are computed in two steps, by first starting with the counts of true positives (TP: 1 predicted as 1, etc), false negatives (FN: 1 predicted as 0 or 2), and false positives (FP: 0 or 2 predicted as 1). Precision is defined as percent predictions of an outcome that are actually correct $(TP/(TP+FP))$. Recall is defined as percent predictions of an outcome that are predicted as such $(TP/(TP+FN))$. F_1-score is the harmonic mean of these two metrics and summarizes model performance in these two dimensions.

$$F_1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

Training set results

model	mean	std	min	25%	50%	75%	max
LM	0.46	0.01	0.44	0.46	0.46	0.47	0.49
GV	0.65	0.01	0.61	0.64	0.65	0.66	0.70
NN	0.71	0.05	0.56	0.68	0.71	0.75	0.82

Test set results

model	mean	std	min	25%	50%	75%	max
LM	0.47	0.03	0.41	0.45	0.47	0.49	0.54
GV	0.58	0.02	0.52	0.57	0.58	0.60	0.64
NN	0.57	0.03	0.49	0.55	0.57	0.58	0.64

F_1-score is the harmonic mean of precision and recall. It summarizes a model's combined performance along these two dimensions. The "perfect" model would have a perfect precision of 100% (i.e., when the model predicts Positive, the outcome is indeed Positive, etc, for all three classes) and a perfect recall of 100% (i.e., the model predicts all Negative sentences as such, and similarly predicts the other two categories perfectly), which would result in an F_1-score of 100%.

$$F_1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

Looking at the training set results, we observe that the NN model has the best performance, with an average f1-score of 71%. GV models is a close competitor, with an f1-score of 65%. The LM model lags with a score of 46%. These results give an initial advantage to GloVe models. This panel also suggests that the NN model shows more variation across trials.

But model performance is ultimately measured by unseen data, so test set results are more important. When we look at these metrics, we see that, first, the LM model performance is pretty similar to its training set performance. This is not surprising since this model has no training and is based on simple counts of words. The only difference is in dataset size: each training set has 700 observations and each test set has 300 observations.

The two GloVe models perform better again, but their test performances are not as strongly better than the LM model as their training performances were. f1-scores are very similar for the two models: 58% (GV) and 57% (NN). The NN model performance has slightly more variance. The fact that both GV and NN models show significant differences in performance between the training and data sets may be an indication of overtraining. Model diagnostics will be carried out in the next iteration of the paper.

There are at least two takeaways from this analysis. First, GloVe representation results in a marked improvement in performance over the LM model, so that using the more complex representation of sentiment may be worthwhile. Second, a simple word-based network based on GloVe representation performs just as well as the more complex, sentence-based LSTM model, at least with this dataset and with no optimization. A larger dataset and an optimized network may give the NN model an advantage however.

3.2 Recall scores

In order to understand the variation in model performance, I look at the components of the f1-score in more detail. Specifically, I look at whether there are differences in model predictions across the three classes of sentences: negative (0), neutral (1), and positive (2) in terms of recall and prediction. This section focuses on recall. I review the results of precision analysis in the next section.

To repeat, recall measures how well a model identifies an outcome correctly. For instance, if there are 100 positive sentences, in a dataset, the perfect model will predict all of them as positive and none of them as negative or neutral. This will give the model a recall score of 100%. If a model predicts few positives when there are 100 positive sentences, or if it predicts many non-positive sentences as positive, it will have a low recall. The main dataset of 1000 observations has an uneven distribution of classes with 48% of sentences belonging to the neutral class, followed by 33% positive class, and 19% negative class. Because of this, training and test sets have more neutral sentences.

When we compare recall scores across classes, an interesting pattern emerges. First, all models do a reasonably good job in identifying neutral sentences. The GV model recalls 79% of all neutral sentences on average, followed by 75% for NN and 73% for LM. The NN model shows the most variation with a standard deviation of 7%. GloVe models have a small advantage in this category.

For negative sentences, LM model performs much better than the GloVe models. The Loughran McDonald dictionary has 2,355 negative words and 354 positive words; the model's superior performance implies that a simple count of negative words may be enough to identify negative sentences. The NN model performs poorly here with 18% recall on average with very wide fluctuations across trials. The GV model recall only about a quarter of negative sentences.

Finally, the NN model does well with positive sentences, recalling 61% on average (with again, a high standard deviation). This is where the sentence-based model shows strength. The poor performance of LM model suggests that many positive sentences may have little or no words (in relative terms) that are identified as positive in the LM dictionary. Similarly, individual words in a sentence may not quite capture the "positivity" of a sentence, giving GV a poorer performance than NN.

The overall result from recall analysis is that while each model has its strength, GloVe models are a bit better in identifying sentences belonging to positive and neutral categories. Even with their poor performance with negative sentences, their overall average recall score is about 59%-60%, above that of the LM model (50%).

3.3 Precision scores

Precision measures how precise a model's predictions are. For instance, if a model makes 100 positive predictions, all of them will actually be positives and none negatives or neutrals. This will give the model a precision score of 100%. If a model predicts few positives when there are many positive sentences, or if it predicts many non-positive sentences as positive, it will have a low precision.

Table 3shows that the GloVe models have somewhat higher precision (58% on average) than the LM model (52%). The LM model does well with positive sentences (61% precision) relative to the other two models. It also does fairly well with negative sentences, with its average precision of 40% edging ahead of the other two models. But it performs poorly with neutral sentences: when it predicts neutral, it is right about half the time only, compared to 75% for NN and 70% for GV.

Table 2: Recall scores across classes (Negative / Neutral / Positive)

This table reports the recall scores of three models used in the analysis. They are computed in two steps, by first starting with the counts of true positives (TP: 1 predicted as 1, etc), and false negatives (FN: 1 predicted as 0 or 2). Recall is defined as percent predictions of an outcome that are predicted as such ($TP/(TP+FN)$). The results show the distribution over 100 trials. In each trial, there are 700 observations (sentences) in the training set and 300 observations in the test set.

class	model	mean	std	min	25%	50%	75%	max
0 (Negative)	LM	0.42	0.05	0.25	0.38	0.42	0.45	0.54
	GV	0.25	0.08	0.08	0.20	0.24	0.30	0.45
	NN	0.18	0.20	0.00	0.03	0.09	0.27	0.70
1 (Neutral)	LM	0.73	0.03	0.65	0.71	0.73	0.76	0.81
	GV	0.79	0.05	0.66	0.75	0.79	0.82	0.89
	NN	0.75	0.07	0.52	0.72	0.75	0.79	0.91
2 (Positive)	LM	0.21	0.03	0.10	0.19	0.21	0.23	0.31
	GV	0.52	0.06	0.35	0.48	0.53	0.56	0.65
	NN	0.61	0.15	0.19	0.53	0.64	0.73	0.87
all	LM	0.50	0.03	0.44	0.48	0.50	0.52	0.56
	GV	0.60	0.02	0.52	0.58	0.60	0.61	0.65
	NN	0.59	0.03	0.50	0.58	0.60	0.61	0.67

Table 3: Precision scores across classes (Negative / Neutral / Positive)

This table reports precision scores of three models used in the analysis. They are computed in two steps, by first starting with the counts of true positives (TP: 1 predicted as 1, etc) and false positives (FP: 0 or 2 predicted as 1). Precision is defined as percent predictions of an outcome that are actually correct ($TP/(TP+FP)$). The results show the distribution over 100 trials. In each trial, there are 700 observations (sentences) in the training set and 300 observations in the test set.

class	model	mean	std	min	25%	50%	75%	max
0 (Negative)	LM	0.40	0.05	0.26	0.36	0.40	0.43	0.54
	GV	0.39	0.07	0.25	0.35	0.39	0.43	0.58
	NN	0.36	0.19	0.00	0.28	0.35	0.47	1.00
1 (Neutral)	LM	0.51	0.03	0.45	0.50	0.51	0.53	0.61
	GV	0.70	0.03	0.62	0.68	0.70	0.72	0.79
	NN	0.75	0.05	0.64	0.72	0.75	0.78	0.90
2 (Positive)	LM	0.61	0.07	0.37	0.56	0.61	0.66	0.76
	GV	0.51	0.05	0.41	0.47	0.50	0.54	0.63
	NN	0.47	0.04	0.38	0.45	0.47	0.50	0.57
all	LM	0.52	0.03	0.44	0.50	0.52	0.55	0.59
	GV	0.58	0.03	0.51	0.57	0.58	0.60	0.64
	NN	0.58	0.04	0.47	0.55	0.59	0.62	0.69

Table 4: Confusion Matrices for Test Sets

This table reports the confusion matrices obtained from the predictions of the three models over 100 trials. For each trial, the counts of each combination (Actual: 0, Predicted: 0, etc) are divided by the total number of observations in the test sets (300) to obtain percent figures. The table below reports the averages of these.

		Predicted			
LM Model		0	1	2	Total
	0	0.08	0.10	0.01	0.19
Actual	1	0.10	0.35	0.03	0.48
	2	0.02	0.24	0.07	0.33
	Total	0.20	0.69	0.11	

		Predicted			
GV Model		0	1	2	Total
	0	0.05	0.06	0.09	0.19
Actual	1	0.02	0.38	0.08	0.48
	2	0.05	0.10	0.17	0.33
	Total	0.12	0.54	0.34	

		Predicted			
NN Model		0	1	2	Total
	0	0.03	0.04	0.12	0.19
Actual	1	0.01	0.36	0.11	0.48
	2	0.04	0.08	0.20	0.33
	Total	0.09	0.48	0.43	

3.4 Confusion Matrices for the Test Sets

Next, we take a look at where models succeed and where they fail by looking at confusion matrices. These diagnostic tools provide counts of observations for each combination of Actual x Predicted category (Actual: Positive, Predicted: Neutral, etc). When looking at the grid of combinations, the perfect model would only have observations on the diagonal (so that Actual: Positive would only be predicted as Positive) and zero entries elsewhere. I scale the counts by the total number of observations in each cell of the grid, then average these over 100 trials. Table 4shows the results.

Starting with the LM model, we see that this model predicts 1 (Neutral) too often: 69% of its predictions are 1. Actual observations are Neutral 48% of the time. The model gets them right about half the time (35%) of the time. It often confuses 2s with 1s and predicts 1 for Positive sentences (24%). Its performance with 0s is better than the other two models: it gets them right 8% of the time. Although it confuses them with 1s 10% of the time. It predicts too few 2s and gets them right less often than the other two models do.

GV model does a better job in terms of prediction distribution although it predicts fewer 0s than the actual observations (12% versus 19%). It predicts 2s 34% of the time, though it gets about half of them right. And finally it's performance for Neutral statements is the best of three: it predicts 1s 54% of the time and it is often right (38%). It also predicts about half of Positive sentences (33% of the sample).

Finally, the NN model does a good job with Positive sentences and Neutral sentences. It predicts 2s too often (43% versus the actual observations of 33%) but is right for 20% of the observations (about half the time). Still, it confuses 2s with 0s (12%) and 1s(11%). Its performance with Neutral statements is strong: it predicts them 48% of the time and gets them right

75% of the time (36%). Its performance with negative sentences is weak: it predicts negative too few times and is wrong more often (1% and 4%) than right (3%).

4 Conclusions and future work

This preliminary and incomplete analysis yields some interesting insights into ways of extracting sentiment from financial statements. Using vector representations of words, either in a simple neural network context or in a sequence model where sentences are treated as units, can help predict the sentiment of a sentence better than straightforward word list based counts. This initial finding suggests the added complexity in this textual analysis exercise may be worthwhile.

The project is in its very early stages and will see a few iterations. This first challenge is to build a reasonably large labelled dataset of sentences. While my projects uses 1,000 of randomly selected sentences, Liu (2010b) Li (2010a)uses 30,000 forward-looking sentences. The data set size matters because neural networks tend to perform better than other machine learning models as the number of observations increase. A larger data set may be particularly advantageous for the NN model.

Second, training stage still needs model optimization through hyper parameter search. For instance, optimal values of learning rate and dropout rate can be obtained by randomly searching parameter space through a large number of iterations. Third, an analysis of whether the models can be used as an ensemble for improved prediction is another avenue that deserves investigation.

Finally, there's work to be done once a model is built based on sentences. Aggregating over sentences to obtain sentiment predictions will yield a variable that can be used, for instance, to predict future earnings, returns, or other variables of interest along the lines of Li (2010b)Li (2010a).

There is a growing interest in using machine learning models in finance both from the academia and the industry. This intersection of the two disciplines may prove very fruitful in the future.

References

Goodfellow, Ian, Bengio, Yoshua, & Courville, Aaron. 2016. *Deep Learning*. MIT Press.

Kearney, Colm, & Liu, Sha. 2014. Textual sentiment in finance: A survey of methods and models. *International Review of Financial Analysis*, **33**, 171–185.

Li, Feng. 2010a. The Information Content of Forward-Looking Statements in Corporate Filings - A Naive Bayesian Machine Learning Approach. *Journal of Accounting Research*, **48**(5), 1049–1102.

Li, Feng. 2010b. Textual Analysis of Corporate Disclosures: A Survey of the Literature. *Journal of Accounting Literature*, **29**, 143–165.

Loughran, Tim, & McDonald, Bill. 2011. When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks. *The Journal of Finance*, **1**, 35–65.

Loughran, Tim, & McDonald, Bill. 2016. Textual Analysis in Accounting and Finance: A Survey. *Journal of Accounting Research, Vol. 54, No. 4,*, **54**(4).

Pennington, Jeffrey, Socher, Richard, & Manning, Christopher D. 2014. GloVe: Global Vectors for Word Representation. *Pages 1532–1543 of: Empirical Methods in Natural Language Processing (EMNLP).*