

金融文書のための別タスク学習による教師なし重要文判定

平野 正徳 *

坂地 泰紀

松島 裕康

和泉 潔

東京大学 大学院工学系研究科

{hirano@g.ecc, {sakaji, matsushima, izumi}@sys.t}.u-tokyo.ac.jp

1 はじめに

近年、ITC技術の向上により、様々な文書が電子的に公開されるようになった。特に、金融においては、インターネット上の即時開示システムを用いて、多くの文書が投資家に公開されるようになってきている。投資家にとって、多くの文書を手に入れることができるようになったことは良いことである一方で、多く文書中の情報を処理することの難しさも表面化してきている。

そこで、自動的にテキストを分析する技術が重要となってきている。例えば、Sakaiらは日本語のニュースのうち、業績要因を抜き出す手法を提案しており、ブートストラッピング的な手法により、自動的に業績要因を抜き出すための手がかり表現を抜き出した[4]。Sakajiらはニュース記事から統計的な手法により、自動的に景気動向を示す表現を抜き出す手法を提案した[6]。さらに、Sakajiらは決算短信の中から自動的に稀な因果関係を抜き出して来る手法を提案した[5]。Kitamoriらは決算短信の中から業績に関する文をニューラルネットワークモデルを利用して取り出し、分類する半教師あり学習の手法を提案している[3]。

これらの手法は投資家の投資判断に役に立つが、必ずしも業績に直結する文だけが欲しいわけではない。業績に繋がるかもしれないが、現在ではまだわからない事象や各会社の取り組みなども必要である。そのため、単純に、重要文を抽出することに一定の重要性がある。[7]も金融文書における重要文判定のタスクをBiLSTMをベースとして行っているが、このような重要文判定において、教師あり学習を適用することは非常に難しい。アノテーションの量的困難さがあるだけでなく、重要文かどうかの基準も難しいため、教師あり学習はあまり向いていないと考える。

そこで、本研究においては、アノテーションの必要のない別のタスク(銘柄判定、価格変動)を学習させることで、教師なし学習により、重要文判定を行う手法を提案する。具体的には、Attentionを用いること

で、銘柄判定や価格変動に繋がる要因をもつ文章に重みを強く持つようなAttentionを獲得することが可能であると考え、これを重要文判定に転移することを考えた。

2 別タスク学習(事前学習)

まず、本研究においては、決算短信の文からいくつかの情報を予測させるという事前学習を行った。ここでは、3節でAttentionを用いることを前提に、ニューラルネットワークを構築した。以下では、データの前処理から順に説明をしていく。

2.1 データと前処理

まず、本研究で使用した決算短信は、期間が2012/10/9-2018/5/11で、ファイル数は63,592であり、株式会社日本取引所グループの提供する、Timely Disclosure Network (TDnet)より取得した¹。

データ削減の観点から、各文の長さは100単語まででカットした。それぞれの決算短信には多くの文が含まれているが、非常に文の多い決算短信は一つの決算短信の中に1,000文を含む場合などがあるため、1つの決算短信あたり100文までを分析対象とすることとした。決算短信の中に含まれる文の本数の分布の分析をしたところ、100文以下である決算短信の割合は87.5%であり、一定の妥当性があると考えている。

以上の前処理により、100単語以下の文を100本以下含むデータをそれぞれの決算短信に対し獲得することができる。

2.2 BERTによる文Embedding

続いて、前処理により獲得した文をBERT[2]によりEmbeddingする。日本語BERTの学習済みモデルとして、柴田ら[8]の公開しているモデル²を利用した。後に述べるが、文中の決算情報の数字などは学習の対象に直接関連のあるものになりうるので、ここで

¹<https://www.jpx.co.jp/equities/listing/tdnet/>

²<http://nlp.ist.i.kyoto-u.ac.jp/index.php?BERT> 日本語 Pretrained モデル

*Corresponding Author; <https://mhirano.jp/>

はあえて、数値データを全角に変更することなく、半角のまま処理することにより、未知語として情報を落とされている。文 Embedding としては、BERT で冒頭に埋め込む [CLS] タグに対応する、後ろから 2 個目の Transformer のアウトプットを文 Embedding として使うことにした。これは、学習済みモデルの隠れ層の次元に対応するので、768 次元である。また、決算短信内の対象文が 100 本に満たなかった場合は、空文として、"[CLS][SEP][PAD]..." という文を追加した。

2.3 事前学習タスクとモデル

ここまでで得られた 768 次元の文ベクトル 100 本をインプットとして利用し、以下のタスクを学習させた。

- Ticker 予測: 決算短信がどの銘柄のものか (約 4,000 クラス分類)
- Up/Down 予測: 決算短信発行 20 営業日前から 20 営業日後までの日経平均株価に対する超過リターンの正負 (2 クラス分類)
- Return 予測: 決算短信発行 20 営業日前から 20 営業日後までの日経平均株価に対する超過リターンの絶対値 (非負値回帰問題)

ここで、20 営業日前後の価格変化を確認したのは、決算短信の直前直後の株価変動ではうまくいかなかったからである。主な原因としては決算短信が発行される前後の株価には決算短信の内容がすでに織り込まれている場合や、決算短信の内容が悪くても、底入れ期待が起こる場合があるほか、投機的戦略が入り乱れるなど、複雑な要因が絡み合っているからである。

後に述べる実験では、これらの 3 つのタスクのうち、全て使った場合だけでなく、一部を使った場合についても検討を行っている。2 つ以上のタスクを学習させる場合には、マルチタスク学習という形を取っている、これらのタスクに対する学習モデルは、図 1 の通りである。

ここで使用している Multi-head Attention は [2] で提案されているものであり、本研究においては 8 Heads とした。Positional Encoding も [2] と同じである。学習にあたっては、Ticker 予測、Up/Down 予測には Negative Log Likelihood Loss を使用し、Return 予測には Mean Square Error を使用し、学習率は 10^{-5} とし、バッチサイズは 100 とした。

3 Attention を用いた重要文判定

重要文の判定にあたっては、2 節で説明した別タスク学習の結果得られた Attention の結果を利用した。

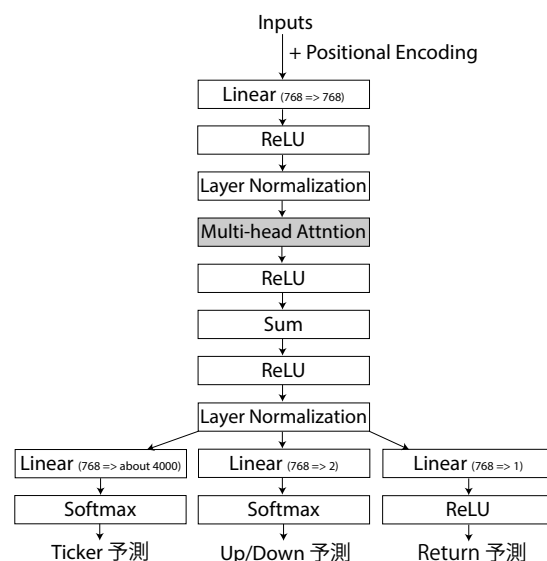


図 1: 学習モデル。"Sum" は Inputs の 100 系列に対する前の層からのアウトプットを全て合計し、 100×768 のインプットを 1×768 に変える。

具体的には図 1 中の灰色に塗りつぶされている Multi-head Attention の結果を用いた。Inputs が決算短信から抽出された 100 文となっているので、Multi-head Attention の各 Head においては、

$$\text{Softmax}(QK^T) \quad (1)$$

と Attention が計算される。ここで、Query(Q)、Key(K) とともに、前のレイヤーから出てきたアウトプットに 1 層の Linear 層を用いることで変換された 100×768 行列である。そのため、式 1 で得られる Attention は 100×100 の行列となっており、それぞれの文間での重み付けに相当する。最終的に事前学習のモデルにおいてはこれらの Attention が、前のレイヤーから出てきたアウトプットに 1 層の Linear 層を作用させた Value(V) を用いて

$$\text{Softmax}(QK^T)V \quad (2)$$

と計算されて、 100×768 のアウトプットを得た後、ReLU 関数で活性化し、 1×768 になるように列の合計をとる。そのため、Attention 行列 $\text{Softmax}(QK^T)$ の列ごとの合計を取ることによって、最終的な予測にそれぞれの文がどの程度寄与しているのかという情報が取れる。これは、予測においてどの程度それぞれの文が重要であったかということの意味すると考えられるので、重要文の判定に使えると考えた。これらの Attention は Multi-head Attention のそれぞれの

Head ごとに出てくるので、それらを平均し、さらに、100本の文に対する重み付けの合計が1になるように、1/100にすることで、最終的なそれぞれの文の重み付けを獲得した。さらに、重要かどうかの判定においては、全体の平均となる1/100を超えているかで判定を行い、1/100を超えていた場合には重要文と判定することとした。

4 実験

実験は、まず、別タスク学習(事前学習)を1,000 Epochs 行い、データセットを8:1:1に分割し、それぞれ、Train, Valid, Test とし、その中で Valid Loss の合計が最小になる時の重みを採用した。同時に、Test のデータでの予測精度についても検証を行った。さらに、その重みを用いて、Attention を計算し、Test のデータに対して重要文かどうかの判定を行った。その後、人手でアノテーションを行った重要文判定結果と照合し、評価を行った。ただし、現状では、人手のアノテーションの限界ゆえに全ての test データに対してアノテーションが行えておらず、709本(13決算短信)に対してのみ照合を行った。詳細については割愛するが、アノテーションにおいては、単純に重要文かどうかで判定を行うと、正確なアノテーションが困難であることから、文を全部で15種類(重要文8種類、非重要文7種類)に分類して行った。

5 結果

まず、別タスク学習(事前学習)における予測精度を示す。結果は表1の通りである。比較手法として、Bag of words を用いた SVM と Logistic Regression の結果を載せてある。これらは、表では便宜上同じ行に Ticker 予測と Up/Down 予測を載せているが、マルチタスク学習を行ったわけではない。Ticker 予測においては、Ticker のみを学習させた場合とさらに価格の上下を同時に予測させた場合のモデルが比較的良好な結果を叩き出しているが、そもそも、比較手法の SVM も Logistic Regression も比較的高い精度を出しているため、統計的に有意かどうかの検討をしなければ、アウトパフォームしているとは言い切れないと思われる。一方で、Up/Down 予測の方は、マルチタスク学習で、Ticker, Up/Down, Return 予測を同時に行ったモデルがもっとも良い精度を出しており、統計的に有意かの検証は行ってないものの、SVM と Logistic Regression との性能差は非常に大きいだけで

なく、Up/Down を単体で学習させた場合よりもかなり良い結果を出している。

次に、重要文の予測精度を示す。表2が結果である。通常の指標に加えて、アノテーションと予測の間での Cohen's κ [1] を計算することで、偶然の一致を除いた評価も載せた。3つのタスクをマルチタスクで行った場合の Attention を用いた場合がもっとも良い結果を出している。

6 考察

まず、別タスク学習(事前学習)について考察する。この学習においては、Ticker 予測というタスクは、Up/Down 予測や Return 予測の精度にも寄与していると考えられる。Ticker 予測においては Up/Down 予測、Return 予測を同時に行うことの効果は限定的であったが、Up/Down 予測および Return 予測においては、マルチタスク学習の効果が出ていた。また、Tickery 予測は、SVM 等でも精度が出ているように、非常に簡単なタスクであるということがわかった。さらに、Up/Down 予測は、SVM においては、適切に分類できず、完全に偏った判定をしてしまっており、Logistic Regression でも提案手法よりは明らかに低い性能しか出ておらず、こちらは難しいタスクであるということがわかったが、提案手法はそれなりの性能を発揮している。加えて、Up/Down と Return 予測は日経平均に対する超過リターンで計算をしているので、トレンドフォローなどの発生も否定できるので、かなり信頼性の高い結果だと考えられる。

次に、重要文判定についてだが、こちらはマルチタスク学習した場合の結果がもっとも良かった。しかしながら、次点が Ticker 予測のみの学習の時であり、マルチタスク学習の組み合わせによっては、Ticker のみを予測させた時に勝てないケースもあり、組み合わせには気をつけなければならない可能性を示唆している。今後の課題として、Up/Down と Return 予測は日経平均に対する超過リターンで計算をしているので、組み合わせで、Return の絶対値の予測をするのではなく、Return の値そのものを入れるなどして試してみたい。

また、Cohen's κ に着目すると、そこまで高い値が出ていないことが目に付く。これ自身は今回の結果が悪いことを意味するわけではないと考えている。そもそも、今回の重要文判定のタスクは非常に難しく、人手のアノテーションが怪しい部分もある。分類項目を分けるなどして、アノテーションの精度維持に努めた

表 1: 別タスク学習 (事前学習) の結果. Precision, Recall, F1 の計算方法はマクロである. 参考までに, 統計的に有意かの検討はしていないが, もっとも良い結果を太字で表示している.

Learning Task	Ticker				Up/Down				Return Loss
	Acc.	Prec.	Recall	F1	Acc.	Prec.	Recall	F1	
Ticker	0.985	0.979	0.980	0.977	-	-	-	-	-
Up/Down	-	-	-	-	0.793	0.780	0.774	0.777	-
Return	-	-	-	-	-	-	-	-	0.696
Ticker + Up/Down	0.985	0.980	0.980	0.977	0.825	0.814	0.813	0.813	-
Ticker + Return	0.981	0.977	0.977	0.974	-	-	-	-	0.712
Up/Down + Return	-	-	-	-	0.811	0.799	0.796	0.798	0.765
ALL	0.983	0.978	0.978	0.975	0.832	0.822	0.819	0.820	0.690
SVM	0.982	0.979	0.977	0.975	0.375	0.188	0.500	0.273	-
Logistic Regression	0.939	0.934	0.924	0.920	0.716	0.704	0.666	0.672	-

表 2: 重要文の予測の結果. Precision, Recall, F1 の計算方法はマクロである. 参考までに, 統計的に有意かの検討はしていないが, もっとも良い結果を太字で表示している.

Learning Task	Acc.	Prec.	Recall	F1	Cohen's κ
Ticker	0.703808	0.687869	0.690318	0.688942	0.378013
Up/Down	0.678420	0.658875	0.656536	0.657558	0.315260
Return	0.669958	0.661139	0.669105	0.661356	0.326429
Ticker + Up/Down	0.682652	0.668203	0.673156	0.669776	0.340405
Ticker + Return	0.698166	0.684641	0.686460	0.686460	0.373933
Up/Down + Return	0.661495	0.651607	0.658770	0.652019	0.307302
ALL	0.715092	0.701669	0.707800	0.703703	0.408234

が, 今後の課題として, 複数人でアノテーションを行い, アノテーションのさらなる高精度化や, アノテーター間での Cohen's κ を計算することで, アノテーションと予測の間での Cohen's κ を相対評価するなどしたいと考えている.

7 まとめ

本研究においては, 近年, 需要が高まっている金融文書の自動解析の一部である, 重要文判定を行った. 大量のアノテーションを必要とする教師あり学習には限界があると考え, 本研究においては, アノテーションの必要のない別のタスク (銘柄判定, 価格変動) を別タスクとして学習させた学習モデルの Attention を用いることで, 教師なしで重要文判定を行う手法を提案した. その結果, マルチタスク学習で複数の要素を学習させた場合ほどより良い結果を出すことがわかった. 今後の課題として, リターンの変化の予測などの学習させるアノテーションが不要な学習ターゲットを変更・追加することや, 正解データを増やすことにより, 性能評価をより正確に行うことなどが考えられる.

参考文献

- [1] Jacob Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, Vol. 20, No. 1, pp. 37–46, apr 1960.
- [2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. oct 2018.
- [3] Shiori Kitamori, Hiroyuki Sakai, and Hiroki Sakaji. Extraction of sentences concerning business performance forecast and economic forecast from summaries of financial statements by deep learning. In *Proceedings of 2017 IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2017)*, pp. 67–73, Honolulu, Hawaii, USA, 2017.
- [4] Hiroyuki Sakai and Shigeru Masuyama. Extraction of cause information from newspaper articles concerning business performance. In *Proceedings of the 4th IFIP Conference on Artificial Intelligence Applications and Innovations (AIAI 2007)*, pp. 205–212, 2007.
- [5] Hiroki Sakaji, Risa Murono, Hiroyuki Sakai, Jason Bennett, and Kiyoshi Izumi. Discovery of rare causal knowledge from financial statement summaries. In *The 2017 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFER 2017)*, pp. 602–608, 2017.
- [6] Hiroki Sakaji, Hiroyuki Sakai, and Shigeru Masuyama. Automatic extraction of basis expressions that indicate economic trends. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2008)*, pp. 977–984, 2008.
- [7] Shuhei Yamaoka, Seiichi Ozawa, Takehide Hirose, and Masaaki Iizuka. 自己注意機構付き LSTM を用いた景況感情報に基づく金融文書の重要文抽出. In *Proceedings of the 33rd Annual Conference of the Japanese Society for Artificial Intelligence*, p. 201J1301, Niigata, Niigata, Japan, 2019.
- [8] 知秀柴田, 大輔河原, 禎夫黒橋. BERT による日本語構文解析の精度向上. 言語処理学会 第 25 回年次大会 発表論文集, pp. 205–208, Nagoya, Aichi, Japan, 2019.