

自然言語処理による面接結果の自動推定

Automatic assessment of Interviews using Natural Language Processing

加口 由樹¹ 黄 宏軒^{1*}

¹ 福知山公立大学情報学部

¹ Faculty of Informatics, The University of Fukuchiyama

Abstract: 面接には主に2つの課題がある。1つは人的コストがかかることであり、もう1つは、面接官の感覚によって評価が変わることである。自動で面接を採点可能な人工知能を開発することを通して、これらの問題の低減、解消を目指す。本研究では、特に発言の内容を用いて、自然言語処理を通じた面接の採点への応用を目指す。

1 はじめに

現代社会では、コミュニケーション能力が重要視されている。例えば、経団連が毎年行っている「新卒採用に関するアンケート調査結果」[1]によると、「選考時に重視する要素」という項目では16年連続で「コミュニケーション能力」が1位となっている。そして、企業が採用の際にコミュニケーション能力を判断するために行っているのが面接である。

しかし現状、面接には「面接を一人ずつ行うのはコストも手間もかかる」、「面接官によって面接の評価が異なる」といった課題がある。「面接官によって評価が異なる」という問題については、仮に全ての就職希望者に対して全ての面接官が面接すれば解決可能だが、手間とコストがかかるため、小規模の場合を除いて現実的ではない。

これらの問題は、機械学習モデルによって面接を評価させ、面接の自動化或いは面接の補助を行わせる事ができると解決可能である。

まず、「面接を一人ずつ行うのは手間もコストもかかる」について、面接の評価を機械学習モデルで行って補助することで、このコストを抑えられる。動画や文章で面接の評価が信頼できる精度で推定できるならば場所や時間を指定して面接官と面接の対象者を集めて面接を行う必要はない。また、面接の対象者は機械学習モデルのフィードバックを利用して、コミュニケーション能力を向上させる訓練を行うこともできるだろう。面接官は、面接の対象者の何を評価すればよいのか、どういった表情が望ましいのかを、その機械学習モデルを利用することで参考にする事ができる。類似の事例として、接客の際にでいい表情を作れているかど

うかを評価し、表情の改善に利用する試み [2] がある。

次に、「面接官によって面接の評価が異なる」について、機械学習モデルを用意して評価させるならば、単一の評価基準で評価された事になる。人間がそれを行うのは労力がかかりすぎるため困難であるが、機械学習モデルを利用するならば現実的なコストで実現可能である。また、機械学習モデルは時間や場所を問わず利用できるという点も利点である。

これらの問題に対して、面接の評価を適切に行う機械学習モデルを開発すれば、面接の自動化や人間の面接官が行う評価の補助を行う事が可能となる。面接における「コストや手間」、「面接官によって面接の評価が異なる」といった問題を軽減可能である。

また、近年機械学習の技術が高まってきた事で、様々な分野で活用が進んでいる。活用が進む分野の1つに自然言語処理があり、Google 検索や Google 翻訳の精度向上、スマートホームアシスタント、電話の自動音声などへの実用化がなされている。その技術的な背景として、深層学習を用いた自然言語処理分野の精度向上があり、BERT[3] や GPT-3[4] の様な技術が注目を集めている。これらは、以前は出来ていなかった文脈に対応した応答を可能にしたことにより、自然言語処理タスクにおける最高スコアを更新するなど [5] 成果を上げている。大規模言語モデルの成果が現実世界への応用が実現されている [6]。

そこで本研究では、面接の自動化や補助を行える機械学習モデルの構築を目標として、これらの自然言語処理の技術を扱い、面接の言語情報を用いて、面接の評価を推定する研究を行うこととした。

*連絡先:

E-mail: hhhuang@acm.org

2 手法について

これまで、分散表現を用いた自然言語処理は、多くの場合で Word2vec を用いて行われてきた。Word2vec を用いて文章を扱う場合、文章中の単語の持つ Word2vec で得られた分散表現を合計して、単語数で割ることで文章の分散表現する。(文章中の単語の分散表現の平均を取る。)しかし、この手法では、分散表現が文章中に出現する単語の前後関係を保持できないという問題がある。ここで、文章中の単語の前後関係を考慮して分散表現を獲得可能なモデルを用いることで、この問題が解消できる。

2.1 Word2vec

言語情報を用いた推定では、Word2vec[7] という手法が広く使われる。Word2vec は単語の分散表現を得るモデルである。この手法では、単語が同じ文章に出現する単語と近い座標となる多次元のベクトルを得るように学習する。この手法を文章に適応する場合、学習させた単語の分散表現を用いて文章の分散表現を獲得する。しかし Word2vec では、文章中にどの単語が出現しているかに注目しているため、出現する単語の前後関係が与える意味への影響を考慮出来ていない。

例：「This is a pen. 」と 「Is this a pen?」が同じものを意味する。

2.2 Universal Sentence Encoder

Universal Sentence Encoder[8](以下 USE と表記する)とは、Google が公開している事前学習済みモデルである。このモデルは文章を直接分散表現に変換することができる。本モデルは Transformer モデルの Encoder 部分を利用して学習されており、Attention 機構によって文中の単語同士の関係を双方向に学習可能なモデルとなっている。そのため、Word2vec では保持できない、文章中の単語の前後関係を考慮した分散表現が得られる。Word2vec と比較して高い精度が得られると期待できる。

3 学習の手順

Word2vec の場合、以下の手順で分散表現を獲得した。

1. 全てのスクリプトを文章単位に分割する。
2. Natural Language Tool Kit から得た Stopword のリストを使って、Stopword を削除する。
3. 全ての文章で word2vec モデルを学習させる。(Gensim というライブラリを用いて、200 次元と 512 次元に

埋め込むモデルを学習させた。)

4. 回答 1 つずつのスクリプトを、単語単位にそれぞれ分割する。

5. 単語を分散表現に変換する。

6. スクリプトに含まれる単語の分散表現の平均をそのスクリプトの分散表現とする。

ここで得られた分散表現を後述の学習に用いた。

USE を用いる学習では、スクリプトごとに USE のモデルに文章を読ませて、分散表現が獲得した。そしてその分散表現を後述の学習に用いた。

分散表現から面接の評価を推定するモデルを学習するに当たって、以下のネットワークの構成で学習を行った。各層の詳細とパラメータ数については図 1 と図 2 に示した。

- ・全結合のニューラルネットワーク
- ・入力：データを USE で分散表現にしたもの、word2vec で変換した単語の分散表現の平均
- ・中間層：4 層
- ・活性化関数：Relu
- ・epoch：200
- ・損失関数：最小二乗誤差
- ・最適化アルゴリズム：adam
- ・データの 9 割を学習、1 割をテストとした

3.1 補足

USE は学習済みモデルのため、得られる分散表現の次元数は 512 次元で固定である。対して、Word2vec のモデルは今回使うデータセットで学習を行うため、出力される分散表現を 512 次元に合わせて学習させた。また、慣習的に Word2vec の次元数は 200~300 次元で学習が行われているため、それに合わせて 200 次元のモデルも学習させた。USE と、Word2vec の 512 次元のモデルでは共通のネットワークで学習させたが、Word2vec の 200 次元のモデルでは、分散表現の次元の大きさに合わせてネットワークサイズをチューニングした。

4 学習結果と総括

学習したモデルをテストデータで評価した。表 1 の結果から、時系列を用いて学習してあるモデルである USE の精度が最も高くなった。また、次元数が高い分散表現になると精度が向上することが分かる。

今回の研究で、単語の時系列を用いて学習された USE は、そうではない Word2vec に対して優位性があることが示された。今後は他の有望な手法とも比較検討を行う。また、分散表現の獲得の手法以外も検討を進めく事が重要である。また、今回は質問文を入力として

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 128)	25728
dense_1 (Dense)	(None, 64)	8256
dense_2 (Dense)	(None, 8)	520
dense_3 (Dense)	(None, 1)	9

Total params: 34,513
 Trainable params: 34,513
 Non-trainable params: 0

図 1: Word2vec200D

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 150)	76950
dense_1 (Dense)	(None, 32)	4832
dense_2 (Dense)	(None, 8)	264
dense_3 (Dense)	(None, 1)	9

Total params: 82,055
 Trainable params: 82,055
 Non-trainable params: 0

図 2: Word2vec512D USE

表 1: 学習結果

	$RMSE$ (※)
<i>Word2vec200dim</i>	0.143804
<i>Word2vec512dim</i>	0.127815
<i>USE</i>	0.112359

※ $RMSE$ は、Root Mean Squared Error. 最小二乗誤差 (0~1 の値を取る) の平方根. 0~1 の範囲を取る.

含めていないが、質問文を何らかの形で与えることで精度の向上に寄与する可能性がある。今回は言語しか検討出来ていないが、それ以外マルチモーダルな特徴量を用いた予測も重要である。

5 謝辞

本研究で使用したデータセットを提供していただいた ETS の Chee Wee Leong 氏に深く感謝いたします。

参考文献

- [1] 経済団体連合会. 新卒採用に関するアンケート調査結果, 2020.
- [2] 資生堂. 資生堂、顔・表情研究から生まれた「笑顔アプリ」を実用化— ニュースリリース詳細 — 資生堂 企業情報, 2017.
- [3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019.
- [4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.
- [5] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019.
- [6] Caswell and Bowen Liang, Software Engineers, Google Research. Recent advances in google translate, 2020.
- [7] Greg Corrado Tomas Mikolov, Kai Chen and Jeffrey Dean. Efficient estimation of word representations in vector, 2013.
- [8] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder. *CoRR*, abs/1803.11175, 2018.