Vol.2018-DBS-167 No.26
Vol.2018-IFAT-132 No.26
2018/9/14

大学が有する潜在文書を活用した報告書推薦システムの構築

赤木里騎[†1,a)]　徐海燕[†1]

概要：本稿では，大学が有している潜在文書を活用することで，就職活動の準備段階から手助けとなる内定報告書推薦システムを提案する．本学では 2008 年度より内定報告書を電子化して Web 上で閲覧できるシステムを運用している．既存システムは企業名や業種，学科などの検索機能を有しているが，報告書の中身を考慮した推薦の能力を持ち合わせていない．そこで，word2vec を利用し分散表現や TFIDF を組み合わせた，自由記述の入力による，類似した報告書の推薦を可能にした．分散表現の学習は登録されている約 3,600 件の報告書のコーパスに，本学学生が将来のキャリアについて記述したキャリアポートフォリオを追加した．また，分散表現を用いた入力と報告書の類似性，LDA によって入力が持つ主題と類似した主題を持つ報告書を推薦する能力を検証する．これにより，利用者が欲しい情報が内容と主題どちらも類似する報告書を読み，就職活動の準備段階から先輩のアドバイスを生かすことが出来ると期待する．第三者が選抜した報告書の集合を用いた検証により，本推薦システムの有効性を示す．

キーワード：情報推薦，情報検索，word2vec，Latent Dirichlet Allocation

Reports Recommendation System Using Documents in the University

RIKI AKAGI[†1,a)]　HAIYAN XU[†1]

Abstract: In this paper, we propose a reports recommendation system supporting job hunting activities of students by using documents in the university. The system can recommend reports by estimating similarity between users' input and reports by using word2vec, distributed representation and TFIDF. In addition to about 3,600 reports that are registered, learning of distributed representation has also be taken the career portfolio described by the students as a corpus. We also validate recommend skill of reports that have similar contents or subject. By recommending the reports having either similar contents or subject, students can improve their activities related to job hunting process. Validity of the recommendation system show by verification using a set of reports selected by a third party.

Keywords: information recommendation, information retrieval, word2vec, Latent Dirichlet Allocation

1. はじめに

近年，自然言語処理の分野で分散表現を用いた研究がある[1][2]．Mikolov ら[3]によって提案された word2vec が分散表現を活用した研究を加速する 1 つの要因となっている．分散表現は文書中の単語を低次元の実数ベクトルで表し，単語間の類似度を算出できる．単語間の類似度が分かることで，情報推薦における入力として活用すれば推薦向上が期待できるため, 自然言語処理において重要な技術である．

本学では学生が就職活動を終えた時に内定報告書を登録して，Web 上で検索できるシステムを運用しているが，報告書は 3,600 件を超え，利用者が求める報告書を探すことは困難となっている．今後，データが増え続け，役に立つ報告書が埋もれてしまうことが考えられる．そこで，本稿では分散表現と TFIDF を組み合わせ，業種や職種などをメタ情報として設定することで報告書を推薦するシステムを構築し，過去の検索履歴ではなく，利用者が求めている情報と，報告書が持つ特徴を用いて報告書を推薦することを目指す．

自然言語処理を用いた本稿の推薦システムの構築における課題は以下のことが考えられる．

① 分散表現の学習に用いるコーパスの適合性
② 利用者や報告書が持つ特徴を活用した推薦
③ 推薦式の推敲

これらの課題を踏まえて推薦システムを構築し，評価実験や検証を行った．

①のコーパスの適合性については，報告書の後輩へのアドバイスと本学のキャリアポートフォリオ（以降，CPF と呼ぶ）2014 年からの 4 年間分のデータを利用した．分散表現の学習にはテキストコーパスのサイズが大きい方が良いとされる．本稿で用いるコーパスは十分な大きさとは言えないが，システムの利用者と同大学に属する利用者が将来のキャリアを記述した CPF を用いた．小さいコーパスでも中身が近く，記述者が類似している場合に分散表現の学習が報告書推薦において有効であることを示す．

②については，300 件位の報告書をランダムに選び，事前分析を行った．それにより，登録者の所属学部，学科や業種，職種にはあまり境界がなく似たようなアドバイスを

†1 福岡工業大学
Fukuoka Institute of Technology.

a) E-mail: mfm17101@bene.fit.ac.jp

Vol.2018-DBS-167 No.26
Vol.2018-IFAT-132 No.26
2018/9/14

記述していることが判明した．このため，LDA によるトピック分類によって報告書の分類を試みた．個々の報告書と所属するトピックに関する情報を提供するようにしている．

1 つの文書に複数のトピックを付与する LDA を用いてトピックの分布値を文書毎に付与し，その分布値から同一の主題で内容が異なる報告書を推薦する研究[4]がある．本稿では LDA を活用してシステムに存在する数少ない類似した報告書の推薦と，評価実験時の課題で，入力が 1 単語における入力と報告書の類似した主題の報告書を抽出する能力の検証を行った．

情報推薦にはコンテンツに基づくフィルタリングと協調フィルタリングがある[5]．協調フィルタリングは大量な利用情報を用いて利用者間の嗜好の類似性を求める．本稿で扱う報告書推薦システムは十分な利用情報が無いため，協調フィルタリングではなく，コンテンツ（報告書）が持つ情報や，利用者の情報を用いる．

③の推薦式については，業種と職種という報告書のメタ情報の重みを利用するかどうかを，利用者に選択できるようにしている．

本稿では分散表現の学習と類似度算出の手法を 2 節で述べ, 3 節で推薦システムの概要を述べる．4 節では推薦システムのアンケート評価実験の結果を述べる．また，4 節にて課題となる問題の解消と，コーパスの違いによる推薦能力の違いについて検証した結果を 5 節で述べる．6 節では，アンケート調査と検証の結果と将来的な報告書推薦システムの考察を述べる．

2. 分散表現の学習

自然言語処理におけるタスクの大半は前処理であるが，word2vec の分散表現の学習においても前処理で単語間の類似度は大きく変わる．

2.1 前処理

本稿ではテキストに対して以下の前処理を施した．

1. クリーニング処理
   1.1 多くの報告書に出現するテンプレートワードの削除
   1.2 HTML タグの削除
2. 単語の正規化
   1.1 英語を半角小文字へ変換
   1.2 カタカナを全角へ変換
3. 助詞と助動詞，記号，数字を削除する分かち書き
4. ストップワードの除去（SlothLibrary の活用[b)]）

報告書推薦システムは運用当初，報告書の後輩へのアドバイス (以降, Advice と呼ぶ) 欄に「[就職活動のポイント，合格の決め手]」のようなテキストを入れていて，多くの報告書の中に存在したため削除した．また，助詞と助動詞は同じ単語でも複数の意味をもつ曖昧な表現であり，分散表現学習の邪魔になる[6]ために削除した．

表 1　入力に対する類似単語とコサイン類似度

入力	面接		筆記試験	
1 位	筆記試験	0.73	spi	0.79
2 位	spi	0.72	筆記	0.74
3 位	おく	0.70	面接	0.73
4 位	面接官	0.70	対策	0.73
5 位	受け答え	0.70	scoa	0.69

表 2　Advice と CPF のテキストデータまとめ

	Advice	CPF
単語数	12 万 5677 個	37 万 2789 個
語彙数	7134 個	1 万 1598 個
文書数	3324 個	3 万 2673 個
平均文字数	130 文字	39 文字

word2vec は分かち書きされたコーパスを入力とする．分かち書きは形態素解析ツールの MeCab[c)]を利用し，分かち書き用の単語辞書には mecab-ipadic-NEologd[d)]を使用した．更新頻度が多く，更新時に新しい固有表現が登録されるため, 比較的新しい報告書に対応できると考えたからである．

2.2 word2vec

分散表現の学習には Python のトピックモデルのライブラリである gensim[e)]から word2vec を活用する．word2vec には分散表現学習モデルとして C-BOW と Skip-gram の 2 つが提案されている．Mikolov らによると Skip-gram が C-BOW に比べ高精度であることが述べられている[7]．そのため，本稿では Skip-gram による学習を行う．word2vec のハイパーパラメータは以下の通りである．

- 学習モデル：skip-gram
- 次元数：100
- n 回未満の単語を破棄：1
- ウィンドウサイズ：10
- エポック：50
- ネガティブサンプリングサイズ：20

学習後の分散表現からある入力単語に対して出力される類似単語の例を表 1 に示す．先頭行が入力で，それ以降の行は類似単語とコサイン類似度を示す．

2.3 CPF

大学は様々なデータを有しているが，その中でも学生自身が将来の目標や学生の間の個人目標を記述した CPF に着目し，分散表現の学習コーパスとして加えた．CPF は 2014 年からの 4 年間分のデータを扱った．

表 2 は内定報告書検索システムに登録されている報告

b) http://svn.sourceforge.jp/svnroot/slothlib/CSharp/Version1/SlothLib/NLP/Filter/StopWord/word/Japanese.txt
c) http://taku910.github.io/mecab/
d) https://github.com/neologd/mecab-ipadic-neologd
e) https://radimrehurek.com/gensim/models/word2vec.html

書の内, Advice を記述しているデータと CPF についてまとめたものである. 単語数と語彙数は前処理後の数値である. CPF は Advice と比較して，1 つの文書に含まれる文字数は短いが，文書数は多いことが分かる.

2.4 TFIDF と分散表現を用いた報告書の類似度

TFIDF は文書中の単語の特徴を重みづけする手法である. 本稿で TF は報告書d内の単語vの出現頻度を示す(式 (1)).

$$TF(v, d_i) = \frac{n_{v,d}}{\sum_{e \in d} n_{e,d}} \quad (1)$$

ここで，$n_{v,d}$はある報告書d内の単語vの出現回数を示し，$\sum_{e \in d} n_{e,d}$は報告書d内の全単語数を示す．IDF はある単語が出現する報告書頻度を逆数とした値で示す．TFIDF は式 (1)に IDF の値を掛け合わせた値である（式 (2)).

$$TFIDF(v, d_i) = TF(v, d_i) \times \log \frac{N}{DF(v)} + 1 \quad (2)$$

ここで，$DF(v)$はある単語vが出現する報告書数で，Nは全報告書数である.

入力と報告書の類似度を計算するために，入力に対して出力される類似単語の類似度を活用する．ただし，同じ単語が複数ある場合には重複を削除した.

$$Similarity(v, d_i) = \sum_{e \in d} s_{e,d} \times TFIDF(v, d_i) \quad (3)$$

ここで，$s_{e,d}$はある報告書に含まれる類似単語の類似度を示す．vが入力単語自身の時には類似度を 1 とする．式 (3)は文書量の大きい報告書の場合に類似単語が出現する回数が増え，文章量の小さい報告書と比べて差が出てしまうため式 (4)のように対数を取ることで影響を小さくした.

$$LogSim(v, d_i) = \log(Similarity(v, d_i)) \quad (4)$$

式 (4)により，ある入力に対する報告書の類似度を算出する．また，式 (4)とは別に，入力と報告書の類似度を求める指標として，文章が入力されることを想定しコサイン類似度を活用する．コサイン類似度は式 (5)に示す.

$$CosSim(W_i, W_r) = \frac{W_i W_r}{|W_i||W_r|} \quad (5)$$

ここで，入力の持つ分散表現をW_iとして，報告書の持つ単語の分散表現を足し合わせた分散表現をW_rとする.

3. 報告書推薦システム

既存の内定報告書検索システムの拡張機能として，自由な入力による報告書推薦システムを構築した（図 1).

利用者の中には明確に気になる業界が決まっている人と，幅広く様々な業界のことを調べたい人がいると仮定して，推薦は業種や職種をメタ情報の重みw_mとして推薦に考慮することとした．推薦する報告書は入力と類似し，報告書の中に特徴のある単語を含むように，式 (4)と式 (5)とw_mを組み合わせて式 (6)とした.

$$LogSim(v, d_i) + CosSim(W_i, W_r)_{norm} \times w_m \quad (6)$$

コサイン類似度は-1 から 1 の値を取るので，正規化したものを$CosSim(W_i, W_r)_{norm}$とした．業種，職種が一致した場合にはそれぞれ重みを 1.2 とし，一致していない時には 1 とした．また，本稿では 1 文のみの報告書は利用者が求める情報を含む可能性は低くなると考えた．1 文の報告書は単語数が 10 以下の場合が多かったため，単語数が 10 以下の場合には推薦する報告書から外した.

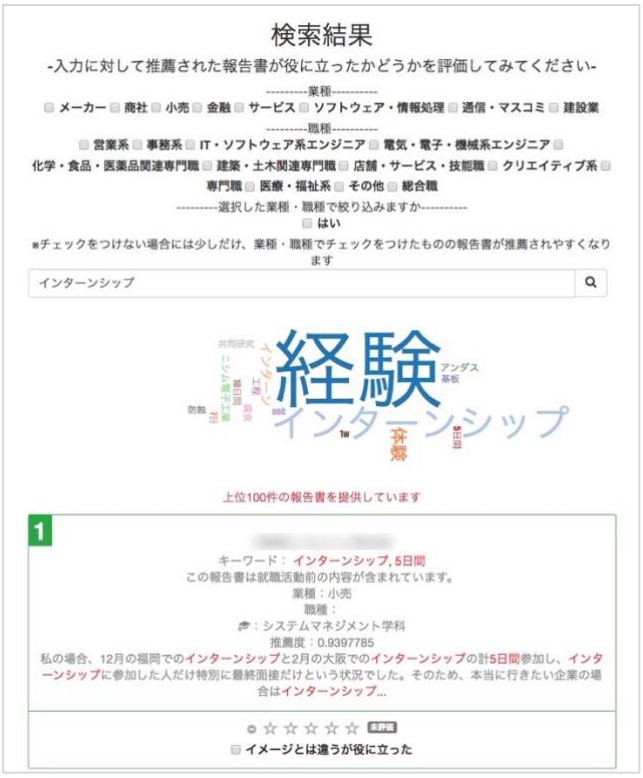

図 1　報告書推薦システム

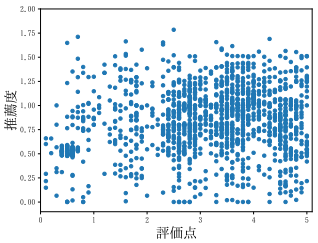

図 2　推薦度に対する評価点の分布

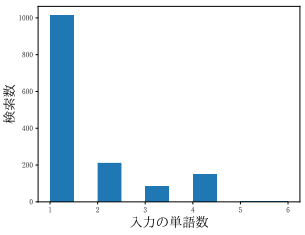

図 3　入力単語数

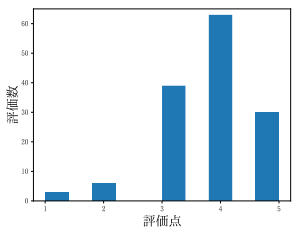

図 4　推薦システムの満足度調査

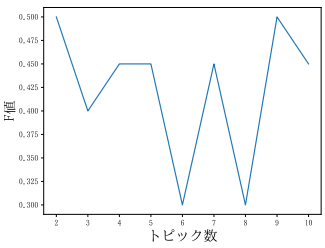

図 5　トピック数と F 値

表 3　頻出単語

入力単語	出現回数
資格	299
面接	296
質問	105
模擬面接	89
SPI	60
ゲーム	40
プログラム	35

Vol.2018-DBS-167 No.26
Vol.2018-IFAT-132 No.26
2018/9/14

表 4　各トピックの推定された単語

トピック	単語（出現確率）
トピック 1 選考過程	面接（0.139），質問（0.021）， 自己（0.017），説明（0.016）， 選考（0.016），練習（0.014）， 行動（0.013），積極（0.011）， 会社（0.010），大切（0.010）
トピック 2 選考前の準備	企業（0.033），勉強（0.031）， 就職（0.021），会社（0.020）， 内定（0.019），情報（0.014）， 知識（0.013），試験（0.012）， spi（0.012），対策（0.011）

報告書推薦システムは与えられた入力と業種，職種から式 (6)を計算し，上位から簡易な報告書の表示をする．図 1 の画面中央には報告書の Advice に含まれた入力との類似語と入力を集計してワードクラウドとして表示した．大きい文字ほど報告書に出現する回数が多いということを表す．画面下部にはカード形式で報告書を表示する．報告書中で TFIDF の高い単語はキーワードとして赤く表示する．検索後にワードクラウドで要約のような結果を確認し，再検索するときにワードクラウドを参考にしながら利用者自身が検索を改善することができる．また，図 1 の例のように入力を「インターンシップ」とした場合，「経験」の単語が最も大きくなっているように，就職活動のためだけではない，本質的な意味合いを集計結果から学ぶこともできる．

4. 評価実験

4.1 評価手順

本学学生，全学部全学年の 141 名を対象に推薦に対する報告書への評価実験とアンケート調査を行った．評価実験は以下の手順で行った．

1. 被験者が自由に検索し，推薦された報告書に対して評価（評価は 0~5.0 の 51 段階評価とし，入力のイメージと近い場合には 5.0 に近く，イメージと違う場合には 0 に近い点数として評価してもらった．）
2. 15 分の間で評価できるだけ評価
3. 評価の後には，報告書推薦システムに対する満足度のアンケート調査

アンケート調査は「推薦の内容は満足できましたか」について，1 を全く満足できなかった，5 をとても満足したとして，5 段階評価をつけてもらった．また，評価を低くつけた場合はその理由を記述してもらった．

4.2 評価結果

明らかに本実験と関係しない入力のデータを除き 1,466 件のデータを得た．図 2 は被験者が評価した点数とその時の推薦度の散布図を示す．縦軸が推薦度で横軸が評価点である．明らかな相関はないが，推薦度が中程度以上の場合に評価点が 3 以上となる場合が多く見受けられた．

また，被験者が入力した文字もしくは文章を MeCab によって分かち書きした後の単語数を図 3，出現単語を出現回数順に，表 3 にまとめた．入力には文章での入力による検索を想定していたが，多くの場合単語数は 1 つであったため，文章の単語の分散表現を足し合わせた報告書とのコサイン類似度が推薦に良い影響を与えているとは言えなかった．5 節ではこの結果を踏まえて入力が 1 単語でも対応できるように，予備実験として，Latent Dirichlet Allocation（以降LDAと呼ぶ）を活用した F 値検証を行う．

4.3 アンケート調査結果

システムの満足度調査として行ったアンケート調査の結果を図 4 に示す．被験者の約 7 割が 4 もしくは 5 点をつけた．低い評価をつけた理由について以下に 2 つ示す．

- アドバイスがどこの企業にも当てはまるようなものであった．
- 気になるキーワードがあったので報告書を見たけど求めているものと少し違うものであった．

5. 検証

5.1 LDA と JSD

評価実験より被験者の多くは 1 単語で入力をする傾向にあることがわかった．式 (6)のコサイン類似度は 1 単語では入力の意図と報告書との類似度を汲み取り切れない．そこで報告書と入力の主題を推定し，入力の意図を汲み取る能力の強化を図った．

LDAは文書中に複数の潜在的な主題を持つと仮定し，トピックを推定する手法である．本稿ではシステムに登録されている類似した報告書を推薦するために，類似した主題の報告書を抽出する能力を検証する．報告書と入力のトピック分布の非類似度を示す指標である Jensen-Shannon ダイバージェンス（以降，JSDと呼ぶ）を活用する．報告書のトピック分布をT_r，入力のトピック分布をT_iとする．

$$JSD(T_r, T_i) = \frac{1}{2}KLD(T_r \,||\, m) + \frac{1}{2}KLD(T_i \,||\, m) \qquad (7)$$

ここで，$KLD(T_r \,||\, m) = \sum_k T_{r,k} \log \frac{T_{r,k}}{m}$，$m = \frac{1}{2}(T_r + T_i)$とする．これにより，報告書と入力のトピック分布間の距離が求まる．類似トピックは 0 に近づき，非類似トピックの場合，値が大きくなる．本稿では全ての報告書に対してJSDを算出して正規化した後に，最も大きいJSDの値からそれぞれのJSDの値を引いた値を$JSD_{reverse}$として活用することで，類似トピックの値が 1 に近づくようにした．検証には式 (4)，(5)と$JSD_{reverse}$から以下の式 (8)を用いた．

$$LogSim(v, d_i) + CosSim(w_i, w_r)_{norm} \times JSD_{reverse} \qquad (8)$$

5.2 使用するトピック数の決定

LDAのトピック数を決めるために，分類精度である F 値を用いて検証する．F 値は以下の再現率・適合率から求めることができる．

Vol.2018-DBS-167 No.26
Vol.2018-IFAT-132 No.26
2018/9/14

表 5　入力を「資格」とした各コーパスの類似単語と類似度

類似度順位	CPF+Advice		WIki		Wiki+CPF+Advice	
1	取得	0.76	社会福祉主事	0.79	食品衛生管理者	0.83
2	土木	0.70	ビルクリーニング技能士	0.78	ワープロ検定	0.80
3	nr	0.68	ワープロ検定	0.78	日商簿記３級	0.79
4	自動車整備士	0.68	日本野菜ソムリエ協会	0.78	第１級陸上特殊無線技士	0.78
5	cg 検定	0.68	食生活アドバイザー	0.77	第２級陸上無線技術士	0.78

類似度順位	Twitter		Twitter+CPF+Advice	
1	実業高校	0.79	実業高校	0.81
2	広丸	0.77	食品衛生管理者	0.78
3	クロスボウエキスパートシャープシューター	0.77	gmes	0.78
4	取得	0.76	クロスボウエキスパートシャープシューター	0.78
5	moscad	0.76	取得	0.77

表 6　コーパスの違いによる F 値検証

表示数	CPF+Advice	Wikipedia	Wikipedia+CPF+Advice	Tweet	Tweet+CPF+Advice
5 件	0.24	0.08	0.11	0.21	0.16
10 件	0.36	0.24	0.16	0.29	0.18
15 件	0.44	0.32	0.27	0.32	0.25
20 件	0.45	0.35	0.33	0.37	0.25

$$再現率 = \frac{表示された適合報告書数}{適合報告書数}$$

$$適合率 = \frac{表示された適合報告書数}{表示報告書数}$$

$$F値 = \frac{(2 \times 適合率 \times 再現率)}{(適合率 + 再現率)}$$

検証に必要な適合報告書 20 件と非適合報告書 80 件を第三者が分類し，計 100 件の報告書を対象に検証を行った．表 3 にて最も入力の多かった「資格」という単語で検索した時に報告書の内容が適するかどうかを選別してもらった．

トピック数を 2~10 とし，表示する報告書数を 20 とした時の F 値を図 5 に示す．検証の結果，トピック数が 2 と 9 の時 F 値が最大になった．トピック数が 2 の時に各トピックに出現する単語を表 4 に示す．トピック 1 は選考過程，トピック 2 は選考前の準備と解釈した．全報告書の中のトピック 1 の値が 2 より大きくなったものは 1,644 件，残りの 1,680 件はトピック 2 に属する値が大きくなった．つまり，どちらのトピックも半分程度の報告書があることを考えると，就職活動生，就職活動前の学生どちらにも役立つ報告書があると考えられる．

5.3 トピック数 2 の場合の F 値検証

本稿では F 値が最大になったトピック数 2，9 の内，特にトピックの解釈がわかりやすかったトピック数 2 の場合において，F 値検証を行った．Advice と CPF のコーパスから word2vec を用いて作成した分散表現以外に 4 つのコーパスを作成して計 5 つ分の分散表現を作成し比較を行った．本稿では，Advice+CPF と比較するために，汎用的な文章の多い Wikipedia と，TwitterAPI を用いて就職活動に関するキーワードを検索キーとして収集した Tweet をコーパスとして用意した．それぞれのコーパスで 100MB 分のテキストをランダムに抽出して 2 つのコーパスを作成した後に，Advice+CPF のコーパスを Wikipedia と Tweet それぞれに追加することで 2 つ分のコーパスを追加作成した．これにより計 5 つのコーパスを作成し，それぞれの分散表現を学習した．入力を「資格」としたときにそれぞれのコーパスのコサイン類似度を表 5 に示す．CPF+Advice をコーパスとした時に上位には自動車整備士や nr，cg 検定などの工業大学ならではと言える資格名が上位に出たが，Wikipedia の場合には汎用的な資格名が多く出た．また，Wikipedia と Twitter に CPF+Advice を加えた場合には，それぞれに含まれる資格に加え，食品衛生管理者や第 1 級陸上特殊無線技士のような本学で取得されている資格名が上位に出現した．

5 つのコーパスを用いた F 値を表 6 に示す．F 値はそれぞれのコーパスで分散表現の学習と検証を 3 回し，平均を

取った値である． 表 6 からわかるように，20 倍近く大きいコーパスを使用した場合でも, Advice+CPF の分散表現のモデルの F 値が常に大きい値となった．また，Wikipedia と Tweet のコーパスに Advice+CPF を加えた場合では, 加える前と比べて F 値が下がる結果となった．

6. 考察

本節では，はじめにで述べた本稿の 3 つの課題について検討する．

6.1 コーパスの適合性

コーパスの適合性については，5 節の検証時に，CPF+Advice コーパス以外に汎用的で大きめの Wikipedia，就職活動に関する内容で大きめの Tweet コーパスを比較することで，CPF+Advice コーパスの有用性を示した．Wikipedia においては学習結果としてはわかりやすい類似語が出現していて，一般的な「資格」という単語とのイメージは近く感じたが，本システムの報告書の内容とはそぐわない場合があったために CPF+Advice よりも F 値が低くなったと考えられる．また，Tweet においては，就職活動に関する情報は取れていたが，本システムの利用者に比べ，Twitter の利用者が年齢，状況が多岐にわたりまとまりのないコーパスとなったことが CPF+Advice よりも F 値が低くなったと考えられる．

6.2 利用者や報告書の特徴を活用した推薦式の推敲

利用者や報告書が持つ特徴を推薦に生かすことと，推薦式の推敲については，本稿では業種や職種によって利用者の特徴をメタ情報として推薦式へ考慮した．

利用者の特徴として活用したメタ情報は他にも，学年や就職活動の状況，希望する勤務地，性別，学部，学科などが考えられる．本稿ではメタ情報として業種と職種の重みを設定したが，前もって利用者が設定できるようにして，推薦式はメタ情報の重みを抜くことで推薦に必要な情報がシンプルになり, 推薦能力が改善されることが考えられる．評価実験や検証は実施していないが, 業種か職種を選択し，業種か職種が一致する場合には推薦式に重みを付与，もしくは，一致しない場合には推薦から外す機能を実装した．利用者自身が推薦から外すのかそれとも重みを変えるのかを選択することで，探している報告書の業種・職種がすでに決まっていても，いなくても対応でき柔軟な推薦を行うことができると考える．この機能は他のメタ情報に対しても設定することができ，拡張可能性が高い．報告書推薦において, メタ情報のような条件は利用者が好きな方を選び, それ以外の分散表現やトピック分布の値を用いることで，より個人に合った推薦が可能になるのではないかと考えられる．

報告書の特徴については LDA を活用してシステムが持つ報告書の主題を推定することでその報告書の特徴とし，入力と報告書の主題が類似するかどうかを$JSD_{reverse}$を活用して算出した．トピック数が 2 の場合に選考過程と選考前の準備段階の報告書に分けられたが，例えば，利用者が選考前と進行中の情報どちらが欲しいかを検索時に選ぶことで推薦度を計算する前に報告書にフィルターをかけることも考えられる．

7. まとめと今後の課題

本稿では就職活動の手助けとなる報告書推薦システムを構築した．分散表現と TFIDF，コサイン類似度から入力と報告書の類似度を計算した．推薦度と評価点に明らかな相関があるわけではないが，これは入力が 1 単語しか持たず，推薦に必要な情報が足りないことが考えられた．そこで，入力の意図を更に汲み取るために，LDA により潜在的な主題が類似した報告書と入力の類似度を$JSD_{reverse}$として算出し，評価実験で用いた推薦式に加えて，推薦する能力を F 値の活用により検証した．

本システムの有効性を示すため，第三者が選別した報告書を用いて適合率と再現率を算出し F 値を求めた．その結果, 2 節で示した CPF+Advice の分散表現モデルが Wikipedia と Tweet の分散表現モデルに比べ常に F 値が高くなる結果を得た．

本稿では，システムに登録されている Advice+CPF のコーパスによる分散表現のモデルの有効性を示し，大学が有する潜在文書の活用の一歩を踏み出した．推薦式の推敲において利用者や報告書が持つ特徴をどのように活用していくか，また検索時に業種や職種だけでなく学科や，卒業年度のような，利用者が指定して検索できるようなシステムの構築が今後の課題である．

謝辞 本稿で使用したキャリアポートフォリオデータ提供にご協力頂いた福岡工業大学情報基盤センター，FD 推進室，運用にご協力頂いている就職課に感謝いたします．

参考文献

[1] 開地亮太，檜垣泰彦，“観光地推薦システムへの単語分散表現の適用”，電子情報通信学会技術研究報告，Vol.115，No.486，pp.45-50，2016.
[2] 野中尚輝，中山浩太郎，松尾豊，“Wikipedia の編集履歴から学習したベクトル表現によるコンテンツの人気予測”，電子情報通信学会論文誌，Vol. J101-D，No.4，pp.657-668，2018.
[3] Tomas Mikolov，Ilya Sutskever，Kai Chen，Greg Corrado，Jeffrey Dean，“Distributed Representations of Words and Phrases and their Compositionality”，In NIPS，pp.3111-3119，2013.
[4] 加藤嘉浩，石井隆稔，宮澤芳光，植野真臣，“Latent Dirichlet Allocation を用いたレポート推薦システム”，電子情報通信学会論文誌，J99-D，Vol.2，pp.152-164，2016.
[5] 土方嘉徳，“嗜好抽出と情報推薦技術”，情報処理学会論文誌，Vol.47，No.4，pp.1-10，2006.
[6] 西村陸，松本忠博，“日本語単語ベクトルの精度向上のための前処理手法の検討”，言語処理学会第 23 年次大会発表論文集，Vol.23,P11-4，2017.
[7] Tomas Mikolov，Kai Chen，Greg Corrado，Jeffrey Dean，“Efficient Estimation of Word Representations in Vector Space”，ICLR Workshop，2013.