自動収集した学習データを用いた 文書分類器に基づくFAQ検索システム

牧野 拓哉　　野呂 智哉

株式会社富士通研究所
{makino, t.noro}@jp.fujitsu.com

概 要

コールセンターを運営する企業ではコスト削減を目的として，想定質問を FAQ として蓄積していることがある．FAQ の質問は，オペレータが意味的に似た問い合わせ履歴をまとめて代表的な表記で作成したものである．そのため，その FAQ で回答できるような問い合わせであっても語彙が一致せず，全文検索エンジンによる FAQ 検索システムでは，適切な FAQ を上位にランキングできないことがある．本稿では，自然文を入力として受け付ける FAQ 検索システムに文書分類器を利用することで，問い合わせと語彙が一致しなくても，適切な FAQ をより上位にランキングする方法を提案する．文書分類器を学習するためには，過去の問い合わせがどの FAQ で回答されたかという情報が必要であるが，本稿で扱うデータには明示的にどの FAQ で回答されたという情報がない．そこで，本稿では FAQ で回答できる問い合わせの集合を自動で収集し，FAQ ごとに二値分類器を学習することで，問い合わせがその FAQ で回答できるかどうかを予測する．実験をおこない，FAQ ごとの二値分類器を用いることで，FAQ と問い合わせの語彙が一致しないような場合でも FAQ のランキング性能が向上することを示す．

1　はじめに

コールセンターにおける FAQ 検索システムは，入力の問い合わせに対して，FAQ 集合の中から適切な FAQ を提示することが求められる．FAQ 検索システムは，基本的に問い合わせに含まれる語彙と FAQ の質問部分や回答部分に含まれる語彙との重複率に基づいて計算される類似度をもとに FAQ をランキングして出力する [6]．しかしながら，このような方法における課題は，一般的な文書検索や質問応答と同じく言い換えの扱いである．以下の例を見ていただきたい．

- 問い合わせ: ○○カードの再発行をしたい．今から出張だが、カードが見当たらない．どうしたらよいか．
- 正解の FAQ の質問部分: ○○カードを紛失・盗難・破損した場合の手続き方法... (後略)

実際のデータは社内情報であるため，作例によるものである．FAQ の質問は，オペレータが意味的に似た問い合わせ履歴をまとめて代表的な表記で作成したものである．そのため，その FAQ で回答できるような問い合わせであっても語彙が一致しないことがある．解決方法の一つは “見当らない” と “紛失” が意味的に同じであるという言い換え表現を人手で作成することであるが，コールセンターのドメイン依存性を考えると，言い換え表現のメンテナンスは高コストにならざるを得ない．

この問題に対して，問い合わせに出現する語と FAQ に出現する語が意味的に同じであるか否かを判別するために WordNet の類義語を教師ありデータとして文脈や表記の類似性から同義語を推定する研究がある [11]．また機械翻訳における翻訳モデルを応用し，質問と回答を対訳文とみなして関連語を獲得する研究 [7, 9, 10] や，類似する質問を収集し，質問と質問を対訳文とみなして関連語を獲得する研究 [5] がある．

本稿では，問い合わせと正解の FAQ の語彙が一致しないという問題がある一方で，Yahoo!知恵袋のような Q&A コミュニティサイトと比べると FAQ は検索対象の数が限られるということと，ある FAQ が正解となるような問い合わせは似たような内容が多いという直感から，問い合わせがある FAQ で回答できるかどうかを予測する分類器を学習できるのではないかと考えた．そのため，本稿では FAQ 検索を文書分類問

題としてとらえる．例えば，“無くす”，“壊れる”，“再発行” のような表現が出現する問い合わせに対して，FAQ ごとの分類器を用いて，その FAQ で回答できるかどうかを予測する．

分類器を学習するためには問い合わせに対してどの FAQ が正解であるかというデータが必要であるが，本稿で扱う問い合わせ履歴にはどの FAQ を参照して回答されたかという明示的なログは残されていないため，FAQ 検索システムを開発する前に，オペレータにヒアリングをして問い合わせに対する対応手順を調査した．その結果，オペレータは FAQ で回答できる問い合わせの際には，FAQ の回答の一部をそのまま対応履歴として記入する傾向があることがわかった．この特性に着目して，過去の問い合わせがどの FAQ で回答されたかという情報を，問い合わせ履歴の回答と FAQ の回答部分の類似度をもとに収集する．本稿で扱う過去の問い合わせの数は FAQ よりも多く，FAQ に対して複数の問い合わせがペアとなりうる．語彙などを素性として利用して FAQ ごとに特徴的な言語表現を学習することで，問い合わせに対してどの FAQ がより正解らしいかを予測する．ただし，FAQ の二値分類器のみを利用して FAQ をランキングすると，正例を収集できなかった場合や，収集した学習データにノイズが多い場合にその FAQ が正解であるかどうかの分類器をうまく学習できない．そのため，本稿では単語の重複率などに加えて，FAQ 分類器の出力するマージンを素性としてランキング学習をおこない，FAQ をランキングするモデルを学習する．ランキング学習では，問い合わせに対して，正解の FAQ が不正解の FAQ よりもスコアが高くなるようにパラメータを学習する．

実験によって，自動で収集したデータをもとに学習した FAQ 分類器を用いることで FAQ のランキングの性能を上げられることを示す．

2　問題設定

FAQ を M 個の質問 Q と回答 A のペアからなる集合 $D_1 = \{(Q_1, A_1), (Q_2, A_2), ..., (Q_M, A_M)\}$ とする．FAQ に正解が存在する問い合わせ履歴を N 個の問い合わせ I と回答 R のペアからなる集合 $D_2 = \{(I_1, R_1), (I_2, R_2), ..., (I_N, R_N)\}$ とする．

本稿の目的は，問い合わせ I に対して，正解の FAQ の質問と回答のペア $(\hat{Q}, \hat{A})$ が一位になるようにランキングを出力することである．誤解を産まないように，本稿では FAQ の質問部分を質問，ユーザから受け付けた問い合わせを問い合わせと呼ぶ．

3　提案手法

提案手法の概要を図 1 に示す．提案手法は大きく 3 つの処理からなる．まず，FAQ と過去の回答履歴をもとに，どのような問い合わせがどの FAQ を使って回答されたという FAQ と問い合わせのペアを収集する．次に，FAQ とペアになる問い合わせを学習データとして FAQ ごとに分類器を学習することで，どのような問い合わせならばその FAQ が正解らしいかという知識を得る．最後に，得られた知識を用いて正解の FAQ をランキング形式で出力するためのモデルの学習方法を述べる．

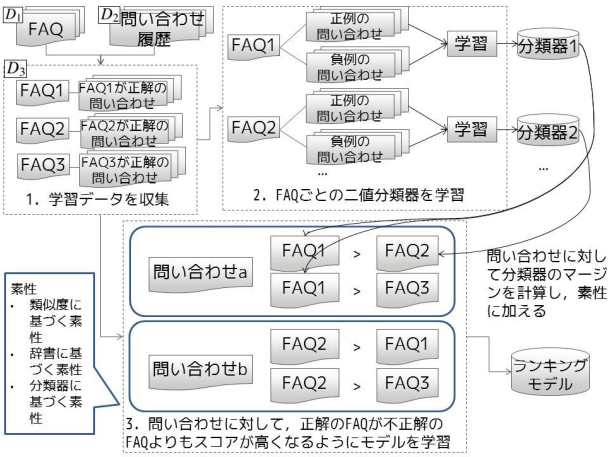

図 1: 提案手法のモデルを学習する処理の概要図

3.1　FAQ と問い合わせのペアの収集

過去の問い合わせがどの FAQ で回答されたかという知識は本稿のタスクにおいて非常に重要である．しかしながら，本稿で扱うデータの問い合わせ履歴にはどの FAQ を見て回答されたかという明示的なログが残っていない．また，人手による正解の FAQ のアノテーションはコストが高い．そこで，本稿ではオペレータの対応手順をヒアリングし，どのように FAQ を利用して回答するのかを調査した．オペレータの対応手順の概要を図 2 に示す．

コールセンターにおいて，オペレータは問い合わせを受け付けたのちに，FAQ から正解を検索して回答することがある．オペレータは対応後に，どのような回答をしたかをテキスト情報として残すのだが，FAQ で回答できるような問い合わせだった場合に，その FAQ の回答の一部をそのまま書き写すことがある．この特

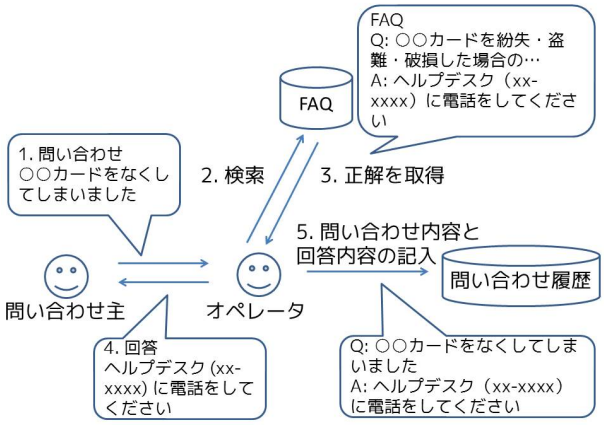

図 2: FAQ を使って回答する場合のオペレータの対応手順

徴をもとに，過去の問い合わせがどの FAQ で回答されたかという情報を，問い合わせの回答と FAQ の回答部分の類似度をもとに収集する．

そこで本稿では先行研究に従い，FAQ と問い合わせのペアをお互いの回答部分の類似度をもとに自動で収集する [5]．FAQ に正解が存在する問い合わせは，オペレータが FAQ の回答を一部書き写して回答することがある．そのため，問い合わせ履歴には FAQ の回答との類似度が高い回答がされた問い合わせが存在する．FAQ と似た回答がされている問い合わせは，その FAQ の質問と意味的に同じことを聞いているという仮定をおき，回答が類似する FAQ に問い合わせを紐づける．具体的には全文検索を使って，問い合わせの回答，FAQ の回答の内容語でお互いに OR 検索し，式 (1) によってスコア $hrank(A_i, R_j)$ を計算する．$rank_{A_m}$ は問い合わせ履歴の回答 R_n を入力として FAQ の回答を検索した場合の A_m の順位，$rank_{R_n}$ は FAQ の回答 A_m を入力として問い合わせ履歴の回答を検索した場合の R_n の順位である．$hrank(A_m, R_n)$ があらかじめ人手で設定した閾値を超えた FAQ と問い合わせのペアの集合 $D_3 = \{((Q_m, A_m), I_n) | 1 \leq m \leq M, 1 \leq n \leq N\}$ を作成する．

$$hrank(A_i, R_j) = \frac{1}{2}(\frac{1}{rank_{A_i}} + \frac{1}{rank_{R_j}}) \quad (1)$$

3.2 FAQ 分類器の学習

FAQ と問い合わせの語彙は一致しないことがある．一方で，ある FAQ が正解であるような問い合わせの集合は語彙の数はそれほど多くないと考えた．そこで，本稿では FAQ ごとの二値分類器を学習することで，ある FAQ が正解であるような問い合わせには，どのような表現が出現しやすいかを学習する．

具体的には，節 3.1 で収集した FAQ と問い合わせのペアの集合 D_3 を用いて FAQ ごとに正例と負例を作成して二値分類器を学習する．対象の FAQ とペアになる問い合わせの集合を正例，その他の FAQ とペアになる問い合わせの集合をすべて負例として学習データとした．ペアとなる問い合わせを持たない FAQ も存在するため，対象の FAQ そのものも正例に追加している．

例えば，“○○カードを紛失・盗難・破損した場合の手続き方法... (後略)” という FAQ の分類器を学習するときに，正例に “○○カードの再発行をしたい．今から出張だが、カードが見当たらない．どうしたらよいか．” という問い合わせがあった場合，“○○カード”，“再発行”，“見当らない” といった素性の重みを正の方向に大きく更新する．パラメータの更新には AROW[4] を用いた．作成した学習データは正例と負例のバランスが偏っており，この学習データで学習した分類器でこの FAQ で回答できる，できないの二値を予測することは難しいため，節 3.3 では予測したラベルを利用するのではなく，マージンを利用する．

素性には，内容語 (名詞，動詞，形容詞)，係り受け関係にある名詞，動詞の対の出現を二値を用いる．名詞句は同一の文節中に連続して出現する接頭詞と名詞とした．また，少なくとも片方が内容語であるような単語 bigram の出現も，同様に二値の素性として用いる．

3.3 ペアワイズランキング学習

ペアワイズランキング学習では，節 3.1 で収集した FAQ と問い合わせのペアの集合 D_3 を用いて，問い合わせに対して，正解の FAQ が，不正解の FAQ よりもスコアが高くなるように重みベクトルを更新する．ランキングのパラメータの学習には Stochastic Pairwise Descent を用いた [8]．

ランキングの重みベクトルの更新手順を Algorithm 1 に示す．問い合わせに対する正解の FAQ およびランダムに選択した不正解の FAQ から抽出した素性ベクトルを取得し，二つのベクトルの差をもとに重み

を更新する．ϕ_r は入力の問い合わせ I と FAQ の質問と回答のペア (Q, A) から抽出する素性ベクトルである．この方法では二値分類器用いて，ペアワイズランキング学習をおこなうことができる．重みの更新 UpdateWeight には AROW[4] を用いた．負例の数 K は 10 とした．

Algorithm 1 ペアワイズランキング学習

$\mathbf{w_r} \leftarrow \mathbf{0}$
for $((\hat{Q}, \hat{A}), I) \in D_3$ **do**
  $\phi_r(\hat{Q}, \hat{A}, I) \leftarrow \mathrm{GetFeatVec}(\hat{Q}, \hat{A}, I)$
  for k **do**1...K
    $(Q_k, A_k, I) \leftarrow \mathrm{GetRndFalsePair}(I, D_1)$
    $\phi_r(Q_k, A_k, I) \leftarrow \mathrm{GetFeatVec}(Q_k, A_k, I)$
    $\mathbf{x} \leftarrow \phi_r((\hat{Q}, \hat{A}), I_n) - \phi_r((Q_k, A_k), I)$
    $\mathbf{w_r} \leftarrow \mathrm{UpdateWeight}(\mathbf{w_r}, \mathbf{x})$
  end for
end for

ランキング学習で用いた素性は次のようなものである:

- **コサイン類似度 cos-q, cos-a**: 問い合わせと FAQ の質問に対する内容語のコサイン類似度，および問い合わせと FAQ の回答に対する内容語のコサイン類似度．これらの値は，問い合わせに出現する単語をより含み，出現する単語の異なり数が少ない FAQ ほど 1 に近い値を取り，そうでないほど 0 に近い値を取る．
- **係り受け関係にある名詞，動詞の対の一致 dep**: 係り受け関係にある文節に出現する名詞，名詞句，動詞の対の一致回数．
- **一致する名詞句の割合 np**: FAQ の質問と問い合わせに対して，出現する名詞句が一致する割合．
- **同義語の一致 syn**: FAQ の質問と問い合わせに対して，日本語 WordNet の同じ synset に属する単語が出現しているかどうか．
- **FAQ カテゴリの一致 faq-cat**: 問い合わせに対して，FAQ のカテゴリを予測し，予測したカテゴリのマージン上位 5 件に，FAQ に付与されている FAQ カテゴリが含まれれば 1，そうでなければ 0 を取る素性．FAQ には FAQ のカテゴリが付与されているため，FAQ の質問を学習データとして分類器を学習した．素性には，内容語の BoW を用いた．FAQ の質問には申請名のような特徴的な単語が出現することが多い．問い合わせがそのような表現を含んでいる場合に正解の FAQ とカテゴリ上に近いということを認識するために利用した．
- **FAQ 分類器の出力 faq-scorer**: 問い合わせに対して，該当する FAQ の二値分類器のマージン $\mathbf{w}_c\phi_c(I)$ を計算し，sigmoid 関数によって $[0, 1]$ へ変換した値を素性に用いる．この分類器は過去の問い合わせ履歴を使って，どのような表現が出現する問い合わせならばこの FAQ が正解らしいかどうかを学習したものである．そのため，この素性は問い合わせに対してこの FAQ が正解らしいほどスコアが 1 に近く，そうでないほど 0 に近い値を取る．FAQ 分類器を用いた素性抽出の例を図 3 に示す．

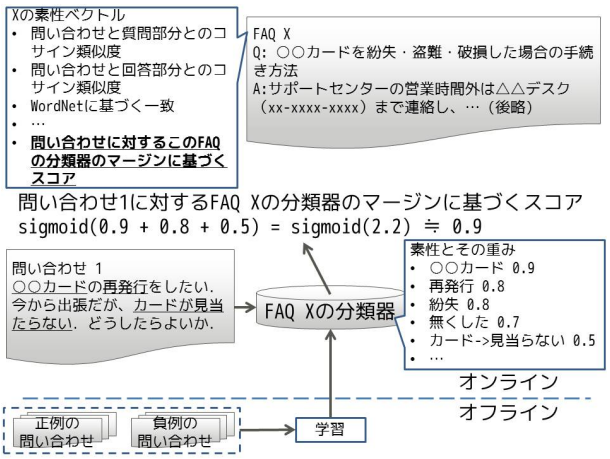

図 3: FAQ 分類器を利用した素性ベクトルの抽出の例

学習した重みベクトル $\mathbf{w}_r$ を使って未知の問い合わせ I に対して FAQ をランキングするときには，各 FAQ から抽出した素性ベクトル $\phi_r(Q, A, I)$ と $\mathbf{w}$ の内積を計算して，その値をもとに FAQ をソートする．

4 関連研究

FAQ 検索や特許検索に対して，検索漏れが起きないように同義語辞書を教師あり学習に基づいて同義語辞書を抽出する研究がされている [11]．本稿の提案手法では，単語間の知識を作成せずに，単語の出現が正解の FAQ にとって重要かどうかを学習している．

関連語を獲得するために，機械翻訳で用いられる IBM Model [1] を用いて単語単位の対応確率を学習す

る研究がされている [5, 7, 9, 10]．IBM Model は単語の対応確率を EM アルゴリズムで推定する手法である．この方法では，FAQ と問い合わせ間の単語の対応確率を学習することができるが，単語の対応確率を学習するには Yahoo! Answers のような大規模な回答済みの質問が必要になる．さらに，単語の対応確率が高くても，正解の FAQ を検索するために有効であるとは言えない．

Cao ら [2, 3] は Yahoo! Answers のカテゴリ情報を考慮した検索モデルを提案した．この手法は，入力の質問と検索対象の質問の単語の一致や，単語の関連確率に対して，入力の質問が，検索対象の質問に付与されたカテゴリに属する確率を重みとしてスコアを計算する．文書分類器を用いて検索をおこなうという観点で本稿と類似する研究であるが，単語の一致に対して入力の質問が検索対象のカテゴリに属する確率を重みを与える方法であるため，単語が一致しにくいような問い合わせに対して FAQ を検索するという問題には適さない．本稿の提案手法は，単語の一致を考えずに，ある素性の出現が検索対象の FAQ にとって重要かどうかを学習している．

5 実験

5.1 実験設定

実験には社内就業システムの FAQ および問合わせ履歴を用いた．問い合わせに対して全ての FAQ をランキングで出力し，得られた正解の順位で手法を比較する．問い合わせには人名や，従業員番号や口座番号が出現するため，個人情報保護の観点からパターンマッチによる秘匿化をおこなっている．その影響で本来は個人情報に当たらない内容も秘匿化されていることがある．

FAQ は 4,738 件存在する．問い合わせ履歴の中から FAQ で回答できるものを 286 件人手で収集した．具体的な情報が書かれる問合せに対して，抽象的な情報となっている FAQ から厳密な正解を定義することが困難であるため紐づけの基準は，この FAQ を見れば納得できる，と判断できた場合に紐づけをしている．評価データを作成する際に，FAQ で回答できる問い合わせの割合を調査したところ，42.2% であった．

データの自動収集で用いた閾値は人手で 0.6 とした．回答が短い FAQ は，誤った問い合わせが多くペアになりうるため，文字数が 10 文字以下の FAQ に対しては収集の候補から除外した．自動で収集した紐づけデータは 27,040 件得られた．問い合わせを紐づけられた FAQ は 1433 件であった．実験時には，評価データに含まれる問い合わせを自動収集したデータから除いた．

形態素解析器，係り受け解析器にはそれぞれ，MeCab[1]，CaboCha [2] を用いた．ユーザ辞書には秘匿化で用いたタグを追加し，秘匿化した際に用いたタグが分割されないようにしている．評価尺度にはランキングの評価で用いられる MRR (Mean Reciprocal Rank)，Precision@N (P@N) を用いた．MRR は正解の順位の逆数に対して平均を取った値であり，正解の FAQ を 1 位に出力できるほど 1 に近い値を取り，そうでないほど 0 に近い値を取る．P@N は正解が N 位以上になる割合である．正解が N 位以上に出力している問い合わせが多いほど 1 に近い値を取り，そうでないほど 0 に近い値を取る．

全文検索と Jeon ら [5] の翻訳モデルを比較手法とする．Jeon らの手法は入力の問い合わせ I を受け付け，式 (2) によって検索対象の FAQ Q をスコアリングする．

$$P(Q|I) = \prod_{w \in Q} P(w|I) \tag{2}$$

ただし，$P(w|I)$ は式 (3) のように計算する．

$$P(w|I) = (1-\lambda) \sum_{t \in Q} (P_{tr}(w|t) P_{ml}(t|I)) + \lambda P_{ml}(w|C) \tag{3}$$

式 (3) の $P_{tr}(w|t)$ は，節 3.1 で収集した D_3 における FAQ の質問と問い合わせを対訳部分とみなして GIZA++[3] を使って学習した単語 w と t の関連確率である．Jeon らの設定に従い，$P_{tr}(w|w) = 1$ というヒューリスティクスを加えている．λ は 0 から 1 まで 0.1 刻みで変えて，評価データに対して最も良くなる値を用いた．

MRR，P@1，P@5，P@10 に対して，paired t-test により有意水準 0.05 で有意差検定をおこなう．

5.2 実験結果

5.2.1 自動収集した問い合わせと FAQ のペアの質

自動収集したデータの中から，無作為に 50 件のペアを抽出し，人手で問い合わせとペアになっている FAQ

[1] https://taku910.github.io/mecab/
[2] https://taku910.github.io/cabocha/
[3] http://www.statmt.org/moses/giza/GIZA++.html

表 1: 人手による FAQ と問い合わせのペアの評価

ラベル	件数
正解	24
不正解	26

が正解らしいかどうかのラベルを付与した．結果を表 1 に示す．

おおよそ半分のデータは正解の FAQ とペアになっており，残りの半分は不正解の FAQ とペアになっている．FAQ の回答が短い場合には，類似する回答がされる問い合わせが多くなることがあるのと，回答の内容は同じであるが，FAQ の質問と，ペアになっている問い合わせの内容が一致しないような事例がみられた．

5.2.2 FAQ カテゴリ予測の精度

FAQ のカテゴリは最大で深さ 3 の階層構造になっている．今回の実験では深さ 2 のカテゴリを用いて実験をおこなった．深さ 2 のカテゴリを利用した場合，カテゴリ数は 107 である．FAQ の質問部分に FAQ のカテゴリを付与した 4,738 件のデータに対して，10 分割交差検定をおこなった．FAQ を学習データとしたのは，FAQ そのものに FAQ カテゴリが事前に付与されているためである．FAQ の質問には社内上の申請名が出現する場合が多くみられたため，申請名などの表現が出現した際に，正解の FAQ とカテゴリ上近いということを認識できると期待して，分類器の出力をランキングの素性として利用した．

表 2: FAQ の質問に対する FAQ カテゴリの予測精度

P@N	評価値
P@1	0.758
P@2	0.839
P@3	0.872
P@4	0.889
P@5	0.898

5.2.3 ランキングの評価

比較手法と提案手法の実験結果を表 3 に示す．全文検索には Elasticsearch[4] を用いた．内容語で OR 検索をして，得られたスコアを順に FAQ をランキングしている．全文検索の評価値が，語彙の一致のみに基づいて FAQ をランキングした場合の評価値である．提案手法は，翻訳モデル，全文検索と比べて MRR，P@1，P@5，P@10 の観点で向上している．

表 3: ベースラインとの比較．提案手法と有意差がある結果に † を付与した．

手法	MRR	P@1	P@5	P@10
提案手法	0.478	0.367	0.605	0.727
翻訳モデル	0.315†	0.238†	0.402†	0.476†
全文検索	0.276†	0.174†	0.388†	0.483†

提案手法の ablation test の結果を表 4 に示す．提案手法は faq scorer が性能向上の寄与が最も高い．また，faq cat による改善もみられる．syn を利用した場合に改善の寄与が見られなかったのは，利用しているデータがドメイン依存であり，一般的な類義語の一致があまり重要でなかったためと考えられる．np を除いた場合にやや評価値が下がっているのは，申請名などの名詞句が出現することが正解の FAQ を見つける根拠として重要であることを示している．

表 4: Ablation tests

手法	MRR	P@1	P@5	P@10
提案手法	0.478	0.367	0.605	0.727
w/o syn	0.478	0.367	0.601	0.727
w/o dep	0.478	0.363	0.612	0.731
w/o np	0.476	0.360	0.605	0.717
w/o faq cat	0.469	0.357	0.598	0.710
w/o cos-{q,a}	0.397	0.311	0.486	0.605
w/o faq scorer	0.346	0.220	0.486	0.601

提案手法の MRR の学習曲線を図 4 に示す．MRR の学習曲線をプロットするために，学習データとして，FAQ と問い合わせのペアを 1,000 件ずつ増やして FAQ 分類器およびランキングモデルを学習している．提案手法は，学習データの量に応じて MRR が向上しており，学習データの質がある程度ノイジーであっても，量を増やすことでランキングの性能向上に貢献していることがわかる．

5.2.4 FAQ 分類器の学習結果

作例であるが，「○○カードを紛失・盗難・破損した場合の手続き」という FAQ に対して学習した分類器

[4] https://www.elastic.co/jp/

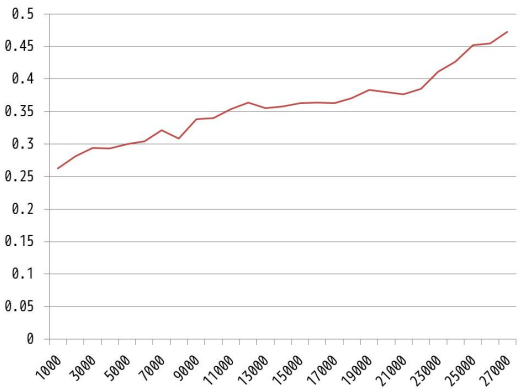

図 4: 提案手法の MRR の学習曲線

表 5: 正の相関がある素性

素性名	素性
係り受け	カード-> なくす
内容語	○○カード
名詞句	再発行申請
単語 bigram	磁気不良
係り受け	カード-> 盗難
単語 bigram	おとした
単語 bigram	財布を

の素性の中から，あまり社内のドメインに偏りすぎない内容であり，かつ重みに正の相関がある素性を人手で選んだものを表 5 に示す．表に示すような学習結果から，例えば “磁気不良”，“おとした” などの表現が出現する問い合わせに対して，この FAQ で回答できると予測することができる．

5.2.5 誤り分析

FAQ 分類器を用いることによって誤る事例の原因には，正解の FAQ に対して学習データとなる問い合わせが存在しない場合がある．評価データに出現する FAQ のうち，学習データで正例となる問い合わせが 0 件で正解の FAQ を 1 位に出力できた問い合わせ 1 件だったのに対して，正例となる問い合わせが 0 件で誤った FAQ が 1 位になった問い合わせが 25 件であった．正例の問い合わせが存在するのに，正解の FAQ を 1 位にできなかった事例については，誤って 1 位になった FAQ に，正解の FAQ の正例と似た内容の問い合わせが存在することがある．これは，回答が短い FAQ には誤った問い合わせが多く収集することによるものであり，回答の類似度による収集方法の改善が必要になる．また，今回は正解の FAQ が 1 つのみとしたが，複数の FAQ が正解になるような事例も見られた．この点については，評価データの設計の修正が必要である．

6 おわりに

自動で収集した FAQ と問い合わせのペアを用いて FAQ 分類器を学習し，FAQ 分類器の出力をランキング学習の素性として用いることで FAQ 検索の性能が向上することを確認した．FAQ という検索対象が限られた状況では問い合わせ履歴を用いて FAQ ごとの文書分類器を学習することで関連語を獲得する手法よりも良い結果が得られた．今後は誤って紐づけられたり，紐づけがない FAQ もあるために起きる誤り事例があるため，対応策を検討する必要がある．

参考文献

[1] Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Comput. Linguist.*, 1993.

[2] Xin Cao, Gao Cong, Bin Cui, and Christian S. Jensen. A Generalized Framework of Exploring Category Information for Question Retrieval in Community Question Answer Archives. In *Proceedings of the WWW*, 2010.

[3] Xin Cao, Gao Cong, Bin Cui, Christian Søndergaard Jensen, and Ce Zhang. The Use of Categorization Information in Language Models for Question Retrieval. In *Proceedings of CIKM*, 2009.

[4] Koby Crammer, Alex Kulesza, and Mark Dredze. Adaptive Regularization of Weight Vectors. In *Proceedings of NIPS*, 2010.

[5] Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. Finding Similar Questions in Large Question and Answer Archives. In *Proceedings of CIKM*, 2005.

[6] Valentin Jijkoun and Maarten de Rijke. Retrieving Answers from Frequently Asked Questions Pages on the Web. In *Proceedings of CIKM*, 2005.

[7] Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu Mittal, and Yi Liu. Statistical Machine Translation for Query Expansion in Answer Retrieval. In *Proceedings of ACL*, 2007.

[8] D Sculley. Large Scale Learning to Rank. In *NIPS Workshop on Advances in Ranking*, 2009.

[9] Radu Soricut and Eric Brill. Automatic Question Answering Using the Web: Beyond the Factoid. *Inf. Retr.*, 2006.

[10] Xiaobing Xue, Jiwoon Jeon, and W. Bruce Croft. Retrieval Models for Question and Answer Archives. In *Proceedings of SIGIR*, 2008.

[11] 森本康嗣, 柳井孝介, 岩山真. 文脈類似度と表記類似度を用いた教師あり同義語抽出. 言語処理学会 第 16 回年次大会 発表論文集, 2010.