検索における分散表現を用いた類似度定量化

齋藤祐樹, 田頭 幸浩, 小野 真吾, 田島 玲†

† ヤフー株式会社　〒 107–6211 東京都港区赤坂 9-7-1 ミッドタウン・タワー
E-mail: †{yukisait,yutagami,shiono,atajima}@yahoo-corp.jp

あらまし　情報検索のタスクにおいてクエリとドキュメントの類似度は検索精度に大きく影響を与える重要な指標の 1 つである. 一般的に, クエリとドキュメントの類似度として局所表現を利用し各単語に次元を割り当て, その各次元の重みを元にスコアを計算する手法が用いられる. 局所表現に基づく指標は疎性を利用して高速に計算できる一方, 言い換えや略記表記などクエリに含まれる文字列を明示的に含まないドキュメントに対して適切に評価を行うことが難しい. これは多様な商品名や型番が用いられる商品検索においては, 特に課題となっている. 本稿では単語を分散表現として扱い, 分散表現から得られる類似度をクエリとドキュメント間の類似度を表わす指標として用いる手法を提案する. 具体的にはクエリとドキュメントそれぞれに含まれる単語の分散表現の和を取り, それらのコサイン類似度を計算する. そのコサイン類似度をクエリとドキュメント間の類似度とし, 得られた類似度と既存の特徴量からランク学習によって予測モデルを学習する. このクエリとドキュメント間の類似度は意味的な近さを考慮したものとなっている. Yahoo!ショッピングの検索ログを用いて予測精度の評価を行い提案手法の有効性を検証した.

キーワード　情報検索, ランク学習, 機械学習, E コマース, 分散表現

1. はじめに

情報検索のタスクにおいてクエリとドキュメントの類似度は検索結果のクリック率などの精度に大きく影響を与える重要な指標の 1 つである. 一般的に, この類似度は各単語にそれぞれに異なる次元を割り当る局所表現を元にクエリに対するドキュメントのスコアを計算する. クエリに対するドキュメントのスコアは局所表現の各次元に対して単語の重みを算出し, その重みを元にスコアを計算する. まず, 単語の重み付け手法について述べる. これはそのドキュメントがどれくらい重要な情報を持っているかについて評価するために利用される. 単語の重み付けは出現頻度 (Term Frequency, TF) やドキュメント内の単語の出現回数と全ドキュメント内の出現回数の逆数の積で表される TF-IDF などが用いられる. これによって各ドキュメントに対して含まれる単語の重みを計算することができる. クエリに対するドキュメントのスコアは局所表現の内積やコサイン類似度などを利用することによって求めることができる. 本稿ではこれを局所表現に基づく類似度と呼ぶことにする.

しかし, 局所表現に基づく類似度は必ずしもクエリに対して意味的な近さを表しているわけではない. そのため, クエリに含まれる単語とは異なるが意味の近い単語を持つドキュメントに対して正しくスコアを計算することが難しい. 例えばクエリに含まれる単語を明示的に含まれない場合 (クエリ:車, ドキュメント:カローラ) や言い換え表現や略称 (PS, プレイステーション) などのクエリと近いまたは同じ意味を指している単語を含むドキュメントに対して正しくスコアを計算することができない. また, クエリに含まれる単語を含むが意図の異なる単語も含まれているドキュメントに対しても正しくスコアを計算することができない. 特に E コマースを対象にした場合, 商品タイトルに関連する単語を多くいれることなどもあり局所表現に基づく類似度がクエリとの意味的な近さからかけ離れてしまうことも多い. 例えばテレビというクエリに対してテレビ本体の商品タイトルが「32 型 ハイビジョン液晶テレビ ブラック」であるのに対して周辺機器が「テレビ用壁掛け金具/液晶テレビ プラズマテレビ テレビ金具」などであると局所表現に基づく類似度は後者のほうが高くなることがある.

Probabilistic Latent Semantic Analysis [3] や Latent Dirichlet Allocation [2] などの手法によってクエリや商品の意図を推定するアプローチもある. しかし, これらの手法ではクエリのように非常に単語数が少ないものを対象にした場合単語の共起関係をもとに学習を行うため意図の推定が困難で, ドキュメントとの類似度についても期待通りの計算が難しい.

そこで, クエリとドキュメントに意図を表わすものとして単語の分散表現を利用し, クエリとドキュメント間の類似度としてそれらの分散表現の和のコサイン類似度やユークリッド距離を用いる手法を提案する. 分散表現はクエリとドキュメントの類似度のスコアとして単語の足し算引き算などのアナロジータスクにおいても非常に高い精度で計算ができると報告されている [5]. 本手法ではクエリとドキュメントの意図や内容をそれらに含まれる単語の分散表現で得られるベクトルの和で決まるとし, クエリとドキュメントを表わす固定長のベクトルを得る. そして, ランク学習においてもクエリとドキュメントの分散表現ベクトルのコサイン類似度やそれらのユークリッド距離を特徴量として既存の特徴量に加えることによって，予測精度が向上することが期待される. 本手法の分散表現ベースと単語ベースの手法におけるベクトルの生成方法の違いについて図 1. に示す.

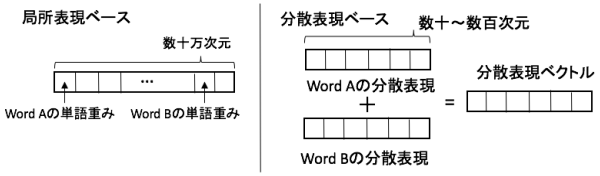

図 1　局所表現ベースと分散表現ベースのベクトルの生成方法の違い

本研究の貢献は以下の 2 点である.

- 局所表現に基づく類似度の代わりにクエリとドキュメントに含まれる単語の分散表現の和を用い, そのコサイン類似度やユークリッド距離をランク学習の特徴量として利用した.
- 提案手法を実データを用いて評価を行い, その有効性を確かめた.

2. 問 題 設 定

この章では本稿における問題設定について述べる. 検索エンジンではユーザーが与えた検索クエリに対して, 限られた時間の中で大量のドキュメントの中からそのクエリに関連したドキュメントを探しだし適切な順序で返す必要がある. 返却候補となるドキュメントの数が少ない場合, 全ドキュメントに対して予測モデルによるスコアリングを現実的な時間内に行うことができる. しかし検索対象のドキュメントの数が膨大な場合, 現実的な時間内にすべてのドキュメントに対して計算コストの高い予測モデルによるスコアリングをすることが難しい. そのようなとき図 2 のように全ドキュメントから適切なドキュメントを選ぶフェーズとそれらの選ばれたドキュメントの中からクエリに対して適切な並び順となるスコアを予測するフェーズを分離し, 2 つのフェーズによって検索結果を返却する手法がとられることがある [1] [7].

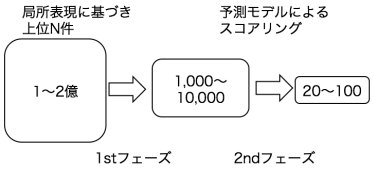

図 2　検索システムの概略図

本稿ではクエリごとに全ドキュメントに対してスコアを計算することは難しいので, 局所表現に基づく類似度で上位 N 件に絞りこんだあとのログに対して評価を行った.

またスコア計算時にクエリとドキュメント間の類似度の他にドキュメントなどのメタ情報などを利用する. このとき入力となるベクトルは図 3 に示す通りドキュメントのメタ情報とクエリとドキュメント間の類似度を結合して利用する.

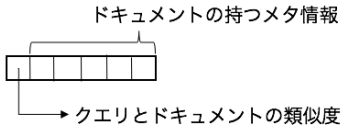

図 3　学習器への入力

3. 提 案 手 法

この章ではクエリとドキュメントの類似度として分散表現ベクトルを用いる提案手法について述べる.

クエリに対して適切な順序でドキュメントを並び替えるために, 局所表現に基づく類似度を用いることがある. これらの指標は非常に高速に計算が可能であり, ドキュメントの数が非常に多くかつ早い応答速度などが求められる場面においても現実的な時間で検索結果を返すことができる. しかし局所表現に基づく類似度ではどの語がドキュメントの中でより大きい重みを持つかどうかしか評価することができず, クエリの意図しているかどうかを評価することが難しいという問題がある. そこでクエリとドキュメントの意味的な近さを表現するスコアを利用する手法を提案する. このクエリとドキュメントの意味的近さを表現するために Skip-gram モデルを利用し, クエリの意図とドキュメントの意図はそれらの単語の意図をそれぞれの足しあわせとすることで意図を表現し, それらのコサイン類似度やベクトル空間上のユークリッド距離を意味的近さを表わすスコアとして利用する. このスコアを特徴量として予測の際に利用することによってクエリに対して適切なドキュメントを決定する. 本稿では予測モデルとして Gradient Boosting Decision Tree(GBDT) を用いた.

3.1　単語の分散表現の獲得

この節は単語にする低次元ベクトルの学習方法について述べる. 単語に対する低次元のベクトル表現を獲得するために分散表現を用いる. 分散表現の学習には Mikolov ら [5] の非常に学習効率のよい 2 つのニューラルネットをベースにした言語モデルの Continuous Bag-of-Words (CBOW) モデルと Continuous skip-gram (Skip-gram) モデルを用いた. CBOW モデルはある単語はその単語が出現した前後数個の単語から意味が推定されるというモデルになっている. 一方 Skip-gram モデルはある単語から前後数個の単語を推定するというモデルになっている. どちらのモデルも入力と出力の間には 1 つの projection 層のみで構成され隠れ層を持たない. この手法は既存のニューラルネットワークベースの言語モデルよりも大幅に計算コストを削減することができた. また, Negative Sampling も CBOW モデルと Skip-gram モデルの学習の効率化に用いられた. どちらのモデルも単語同士の類似度の評価のタスクにおいて精度がよかった. 本稿では分散表現の学習には同様のタスクで多く用いられる Skip-gram モデルを利用した.

3.2　クエリとドキュメントの分散表現の獲得

この節では学習した分散表現を元にクエリとドキュメントの意図推定をする手法について述べる. 本手法ではクエリやド

キュメントの意図がそれらに含まれる単語の意図の足しあわせであるとし, クエリやドキュメントの分散表現の和で表現する.

3.3 クエリとドキュメントの類似度計算

3.2 でクエリとドキュメントに含まれる単語からそれらの意図を推定した. この節ではこれらからクエリとドキュメントの意味的な近さを算出方法について述べる. クエリとドキュメントの意味的な近さを表わすスコアとして 3.2 で得たクエリとドキュメントの分散表現のコサイン類似度とユークリッド距離を用いる. w_q,w_d をそれぞれクエリの分散表現とドキュメントの分散表現とするとコサイン類似度とユークリッド距離は以下の表される.

$$Similarity(x'_q, x'_d) = \frac{x'^T_q x'_d}{\|x'_q\|\|x'_d\|}$$

$$Distance(x'_q, x'_d) = \sqrt{\sum_i (x'_{q,i} - x'_{d,i})^2}$$

4. 実験設定

この章ではデータセットと評価方法について述べる.

4.1 データセット

実験に用いるデータセットとして Yahoo!ショッピングの 2015 年 9 月の 1ヶ月分の検索ログの一部を利用する. 2015 年 9 月 1 日から 2015 年 9 月 20 日までの検索ログを訓練データとして,2015 年 9 月 21 日から 2015 年 9 月 30 日までを評価データとして利用する. ラベルとしてそのクエリに対して返却対象となったドキュメント (商品) がクリックされたかどうかを用いる. 実験にあたり 1ヶ月の間に一定以上の検索回数があったクエリに絞り込んだ. データセットのサマリは表 1 に記載する.

	訓練データ	評価データ
#query	309,425	123,824
#document	10,253,064	3,387,381

表 1　実験データのサマリ

分散表現の学習には word2vec(注1) を利用した. 単語の分散表現の学習にコーパスとして表 1 の訓練データを用いる. 分散表現を学習するためのコーパスの作成にはクリックされたかどうかに関わらず訓練データに含まれる商品タイトルのみを抽出した. そのため, 評価時に訓練データに出現しなかった単語に対して分散表現が存在しないことがある. このときは出現しなかった単語の分散表現として零ベクトルを利用する. 分散表現の学習には Skip-gram モデルを用い各単語に対して 100 次元のベクトルを学習する. 学習にあたってウィンドウ幅は 5, α は 0.025 とした. また, 今回はスコア関数の学習にドキュメントとクエリの類似度のほかにに商品に付与される他の特徴量を用いた. これらの特徴として商品のページビュー数, 価格, レビュー数, レビューの平均などの特徴量を用いた.

(注1) : https://code.google.com/archive/p/word2vec/

4.2 Gradient Boosting Decision Tree

この節では学習器として用いる GBDT について述べる. GBDT は Gradient Boosting を利用した決定木ベースの学習の 1 つで精度が高いことで知られている. 弱学習を複数組み合わせることで汎化能力を向上させるアセンブル学習の 1 つで, GBDT では損失関数が最も小さくなるような弱学習器を学習し, それをいままの学習器に追加する. GBDT は学習器として決定木を利用したものである. Gradient Boosting のアルゴリズムは N をデータ数, J を弱学習器の数, h を弱学習, F をアンサンブル学習器, $\mathbf{a}$ を学習器のパラメータとしたとき以下のように与えられる.

Algorithm 1 Gradient Boosting

$F_0(\mathbf{x}) = \arg\min_{\rho} \sum_{i=0}^{N} L(y_i, \rho)$
for $j = 0$ to J **do**
  $\tilde{y}_i = -[\frac{\partial L(y, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}]_{F(\mathbf{x})=F_{j-1}(\mathbf{x})}$
  $\mathbf{a}_j = \arg\min_{\mathbf{a},\beta} \sum_{i=1}^{N} |\tilde{y}_i - \beta h(\mathbf{x}_i; \mathbf{a})|^2$
  $F_j(\mathbf{x}) = F_{j-1}(\mathbf{x}) + \rho_j h(\mathbf{x}; \mathbf{a}_j)$
end for

また, 今回はクリックされたドキュメントとクリックされなかったドキュメントに対して損失関数を設定するために gbrank [10] を利用する. gbrank は Gradient Boosting におけるペアワイズの損失関数となっており, 予測後の順番が違うペアに対して損失が小さくなるように弱学習器を学習する手法である.

4.3 評価方法

ランク学習において nDCG と MRR という指標がよく用いられる. どちらの指標もリストに対するドキュメントの並び方に対して評価をする手法である. 本稿では評価実験にはスコアの上位 10 件までの nDCG(nDCG@10) と MRR の 2 つで評価を行う.

Normalized Discounted Cumulative Gain (nDCG) DCG はリストの並び順を評価する指標の 1 つで, よりクエリに対してより適切なドキュメントの順位を高く評価するほどスコアが高くなる. nDCG はそれをもし理想的な並び順になってたときの DCG(Ideal DCG) との比として表される. y_i をクエリに対するドキュメントの適合度を表わすラベルとしたとき以下の式で上位 k 個のドキュメントの並びに対する nDCG は計算される.

$$DCG@k = \sum_{i=1}^{k} \frac{2^{y_i} - 1}{\log_2(i+1)}$$

$$nDCG@k = \frac{DCG_k}{IDCG_k}$$

Mean Reciprocal Rank (MRR) MRR の nDCG と同様にリストの並び順を評価する指標である. MRR はリスト内で最初にクリックされたドキュメントの順位の逆数の平均として算出される.

5. 実　　験

この章では提案手法について行った評価について述べる.

本稿では Yahoo!ショッピングの検索ログを用いてクエリに対してクリックされたドキュメントの順位が高くなるように予測モデルを学習した. 本稿では 2 つの実験を行った. 1 つはクエリに対してクリックされた商品とそうでない商品が分散表現でどのような性質を持っているかを確認するために, 学習によって得られた分散表現からクエリとドキュメントの意図を推定し, それらのベクトルをラベル別にプロットした. 2 つめは本手法の有効性を確認するために局所表現に基づく類似度の代わりにそれらの分散表現から得られるベクトルのコサイン類似度を意味的近さを表わす特徴量で置き換え, スコア関数を学習し評価を行った.

5.1　クエリとドキュメントの分散表現の評価

クエリとドキュメントに含まれる単語の分散表現の和のベクトルをそれぞれの意図を表わすベクトルとして, クエリとクリックされたドキュメント, クリックされなかったドキュメントのベクトルの主成分分解の上位 2 軸をプロットした. クエリのベクトルとクリックされたドキュメントの距離が近いものを図 4 に, クエリとクリックされたドキュメントのベクトルの距離が遠いものを図 5 に示す. 図 4 はクエリの意図とタイトルの意図が近いドキュメントがクリックされていることを示している. これらのクエリは意図が明確であり, その意図に近いドキュメントがクリックされていることと考えられる. 一方図 5 はクエリの意図とタイトルの意図がドキュメントの意図が違うドキュメントがクリックされていることがわかる. これはクエリの意図が曖昧なクエリ, 複数の意図があるクエリなどに対してタイトルの意図が近いものが近いものがクリックされるわけではないことがわかる. これらのクエリに対しては分散表現から得られるベクトル同士のユークリッド距離やコサイン類似度を元に上位 N 件を返却するというランキングしても精度の向上に繋がるわけではないことがわかる.

また提案手法は既存手法と比べてスコアのみでランキングをした場合, 表 2 に示したように nDCG@10 で 5.1%,MRR で 4.2%の精度向上を確認することができた. これはクエリとドキュメント間の類似度のみでランキングをした場合でも単語ベースのアプローチよりもクエリの意図した商品を返していることがわかる.

	nDCG@10	MRR
局所表現に基づく類似度のみ	0.332	0.310
分散表現のユークリッド距離のみ	0.324	0.304
分散表現のコサイン類似度のみ	**0.349**	**0.323**

表 2　クエリとドキュメント間の類似度のみを用いた実験結果 (nDCG@10,MRR)

5.2　ランク学習の特徴量として用いた評価

実験にあたってベースラインではクエリとドキュメントの類似度として BM25 を用い, それ以外の特徴量として商品のページビュー, レビューの数, レビューの平均点, 価格などの商品の持つ特徴量を利用した. 評価に関して nDCG@10,MRR について評価を行った. その結果を表 3 に記載する.

	nDCG@10	MRR
局所表現に基づく類似度+商品に関する特徴量	0.445	0.423
分散表現のユークリッド距離 + 商品に関する特徴量	0.460	0.436
分散表現のコサイン類似度+ 商品に関する特徴量	**0.462**	**0.437**
すべての特徴量	0.454	0.434

表 3　実験結果 (nDCG@10,MRR)

提案手法はクエリとドキュメント間の類似度の他にドキュメントの持つ特徴量を加え, 予測モデルによってランキングした場合でも nDCG@10 で 3.8%, MRR で 3.3%の精度向上を確認することができた. これによってスコア関数の予測において単語ベースの類似度ではなく分散表現で得られる意味的な近さのほうが精度に寄与することを確認できた. クエリとドキュメントの類似度とクエリとドキュメントの分散表現のすべて加えたものを特徴量に加えた予測モデルに関して単体で追加したものに比べて予測精度が悪かった. これは訓練データに対して過学習をしており, 評価データに対する予測精度が落ちてしまっているものと考えられる. 過学習が起きている原因として考えられるのは特徴量をすべて加えて場合, 既存手法や局所表現に基づく類似度を提案手法のコサイン類似度に n 置き換えたものに比べて, 次元数がクエリとドキュメントの次元数だけ増加してしてしまっている. そのために特徴量の次元数に対して訓練に用いたデータセットの数は固定としたため訓練データに対して過学習をしてしまった原因と考えられる.

6. 関 連 研 究

意図推定. ドキュメントから意図を抽出する手法として単語の共起関係に基づく Latent Semantic Analysis(LSA) や, 生成モデルに基づく Latent Dirichlet Allocation(LDA) などが挙げられる. LSA や LDA はドキュメントからトピックを抽出する方法として自然言語処理の分野で多く用いられる. LDA は文章生成モデルの 1 つで T 個のトピックごとにディリクレ分布から単語出現を生成し, ドキュメントごとにディリクレ分布から単語生成確率を生成する. これらの手法ではドキュメントに対する類似度やトピックの分布を得ることができる. クエリとドキュメントのトピックの分布の類似度を局所表現に基づく類似度の代わりのスコアとして利用することができる. しかしクエリや商品タイトルなどは含まれる単語が少なく共起関係を元に学習するためクエリのようにクエリに含まれる単語数が非常に少ない場合トピックの推定の難しい. この問題に対して Yu ら [8] は Collapsed Gibbs Sampler をベースとした LDA を改良した Multivariate Bernoulli LDA という手法を提案した. これはドキュメントに結びつく単語が少ないコーパスでも多様性を目的に置いた情報検索タスクにおいて通常の LDA よりも精度よくトピック推定ができることを示した.

Deep Structured Semantic Model, DSSM. DSSM [4]

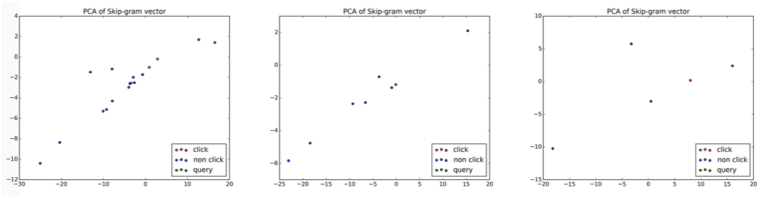

図 4　クエリに対して近いドキュメントがクリックされている例

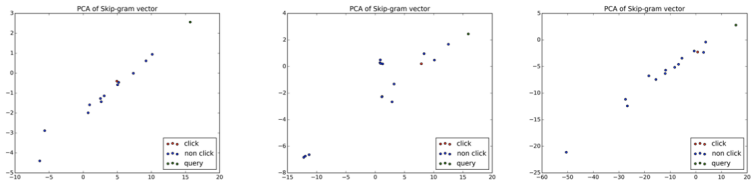

図 5　クエリに対して遠いドキュメントがクリックされていない例

[6] はニューラルネットワークを利用した情報検索のアプローチの 1 つである．DSSM はクリックの予測タスクにおいて LSA などのこれまでの手法と比べて精度の高い手法である．LSA や LDA などでは単語ベースでの意図の推定が難しいことが問題であった．そこで DSSM ではまず文字列を tri-gram によって分割する．具体的には tri-gram は先頭文字と終端文字に '#' を識別文字を加えて 'cat' を '#ca','cat','at#' という文字列に分割する．これによってボキャブラリのサイズが大きくなっても特徴量となる分割後の文字列は文字の種類の 3 乗で抑えることができ, また未知の単語に対しても予測を行うことができる．tri-gram による分割後に得られた特徴量としてクエリに対してドキュメントがクリックされるかどうかを判定する識別器を学習する．クエリに対するドキュメントがクリックされるかどうかの事後確率は以下の類似度とソフトマックス関数で計算する．Q,D はそれぞれクエリとドキュメントで γ はハイパーパラメーターを表わす．

$$R(Q,D) = cosine(y_Q, y_D) = \frac{y_Q^T y_D}{\|y_Q\|\|y_D\|}$$

$$P(D|Q) = \frac{\exp(\gamma R(Q,D)}{\sum_{D' \in D} \exp(\gamma R(Q,D'))}$$

この手法はクエリに対して対象となるドキュメント集合の中でどれが一番クリックされやすいかという問題を直接最適化しているアプローチである．DSSM ではでは tri-gram による文字列分割を行っており, この手法はデータ内の文字の種類が少ない場合は次元圧縮を可能とする．しかし例えば日本語が含まれるドキュメントを対象としたとき常用漢字でも 2000 文字程度存在し, これの tri-gram による分割を行っても次元数が 80 億程度となり次元圧縮にはならない．次元圧縮となるような文字列の分割方法を利用すれば我々の問題にも適用可能である．損失関数がコサイン類似度のスコアのソフトマックス関数の交差エントロピーに関してもクエリに対してドキュメントのスコアを計算するときに文字列以外の特徴量, 例えばドキュメント内の単語の重複数などやタスク固有の特徴量なども利用できるようにば適用可能である．また提案手法ではクエリと分散表現の獲得に教師なし学習とベクトルの和を用いたが DSSM のクリックから教師あり学習で表現を獲得ことができれば精度向上に繋がると思われる．

Answer sentence selection．自然言語処理の研究分野の 1 つに自然文で与えられる質問に対して適切な解答を抽出する Answer sentence selection という分野がある．Answer sentence selection にはクエリに対して自然文を生成する手法と与えられたドキュメントの中からよりクエリの解答に近いセンテンスを選ぶ手法の 2 つある．後者の手法はドキュメントの中から適切なセンテンスを選ぶという点で情報検索のタスクと見なすことができる．

これまでに精度がよい手法ではドキュメントやクエリの文章の構造に関する情報を利用する手法が多く用いられてきた．近年はこれらの手法とは異なり,Yu ら [9] が構造に関する情報ではなく分散表現を利用した意味的な情報を利用するアプローチを提案した．

7. おわりに

本稿では情報検索のランキングモデルにおいて返却候補となるドキュメントが多い場合におけるクエリとドキュメントの類似度について一般的に用いられる局所表現に基づく類似度に比べ, 分散表現を用いた意味的近さを表した類似度を予測の特徴量として用いる手法について提案した．提案手法では単語の分散表現として skip-gram モデルによって学習し, そのうえでクエリとドキュメントの類似度にそれらに含まれる単語の分散表現のベクトルの和のコサイン類似度を用いた．また提案手法を Yahoo!ショッピングの検索ログを用いて評価を行い, 予測精度が向上することを確認した．

予測に関して GBDT ではなくニューラルネットを用いた手法も提案されており, 今後はより精度の高い学習方法の適用や過学習が今後の課題として挙げられる．

文　献

[1] Deepak Agarwal and Maxim Gurevich. Fast top-k retrieval for model based recommendation. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*, WSDM '12, pp. 483–492, New York, NY,

USA, 2012. ACM.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, Vol. 3, pp. 993–1022, 2003.

[3] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 50–57. ACM, 1999.

[4] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, pp. 2333–2338, New York, NY, USA, 2013. ACM.

[5] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pp. 3111–3119. Curran Associates, Inc., 2013.

[6] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14 Companion, pp. 373–374, Republic and Canton of Geneva, Switzerland, 2014. International World Wide Web Conferences Steering Committee.

[7] Yukihiro Tagami, Toru Hotta, Yusuke Tanaka, Shingo Ono, Koji Tsukamoto, and Akira Tajima. Filling context-ad vocabulary gaps with click logs. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pp. 1955–1964, New York, NY, USA, 2014. ACM.

[8] Jun Yu, Sunil Mohan, Duangmanee (Pew) Putthividhya, and Weng-Keen Wong. Latent dirichlet allocation based diversified retrieval for e-commerce search. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pp. 463–472, New York, NY, USA, 2014. ACM.

[9] Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. Deep Learning for Answer Sentence Selection. In *NIPS Deep Learning Workshop*, December 2014.

[10] Zhaohui Zheng, Keke Chen, Gordon Sun, and Hongyuan Zha. A regression framework for learning ranking functions using relative relevance judgments. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 287–294. ACM, 2007.