

Predicting Best Answerers for New Questions: An Approach Leveraging Topic Modeling and Collaborative Voting

Yuan Tian
Pavneet Singh Kochhar
Ee-Peng Lim
Feida Zhu
David Lo

背景

- StackOverflowなどのCQAサイトには膨大な質問と回答が投稿されている
- CQAサイトの目的は、質の高い回答と様々な解決策を提供すること
- 質問者は回答を得るのに数日かかる場合もある。また、
質の高い回答が得られるとは限らない

目的・アプローチ

- 目的
 - 質問に対する回答のクオリティを上げることと，回答時間の短縮
- アプローチ
 - 新しい質問に対して適切な回答を得られるユーザーを予測するシステムの提案
 - ・ ベストアンサーを提供するユーザーの予測

関連研究

- ベストアンサー推薦は広く研究されている
 - PageRank, HITSを利用した手法[4]
 - LDAを利用した文書の類似度に基づく手法[7]
- 関連研究との違い
 - トピックを利用して, 回答者の専門性と興味の両方を考慮する点
 - StackOverflowで利用される投票機能のスコアを利用する点

[4]Bouguessa, M., Dumoulin, B., Wang, S.: Identifying authoritative actors in question-answering forums: the case of yahoo! answers. In: KDD, pp. 866–874 (2008)

[7]Liu, M., Liu, Y., Yang, Q.: Predicting best answerers for new questions in community question answering. In: Chen, L., Tang, C., Yang, J., Gao, Y. (eds.) WAIM 2010. LNCS, vol. 6184, pp. 127–138. Springer, Heidelberg (2010)

アプローチの概要とランキングモデル

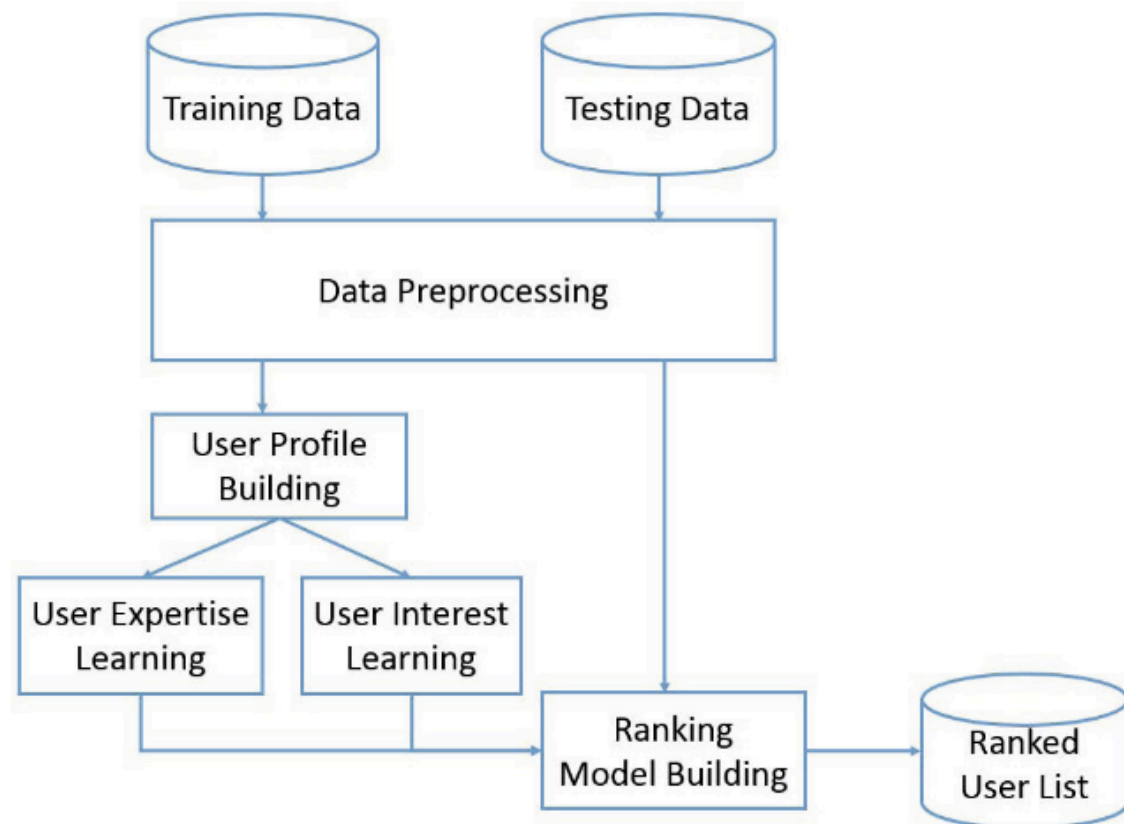


Fig. 1. Overall framework

アプローチの概要

- ランキングモデル

- ユーザー(u)が質問(q)に対してベストアンサーを提供する確率

$$P(u|q) = \frac{P(u)P(q|u)}{P(q)}$$

- $P(q)$
 - ・ ユーザーが質問(q)に関わる確率
- $P(u)$
 - ・ ユーザー(u)が質問に答える事前確率
- $P(q|u)$
 - ・ ユーザー(u)が質問(q)に答える確率

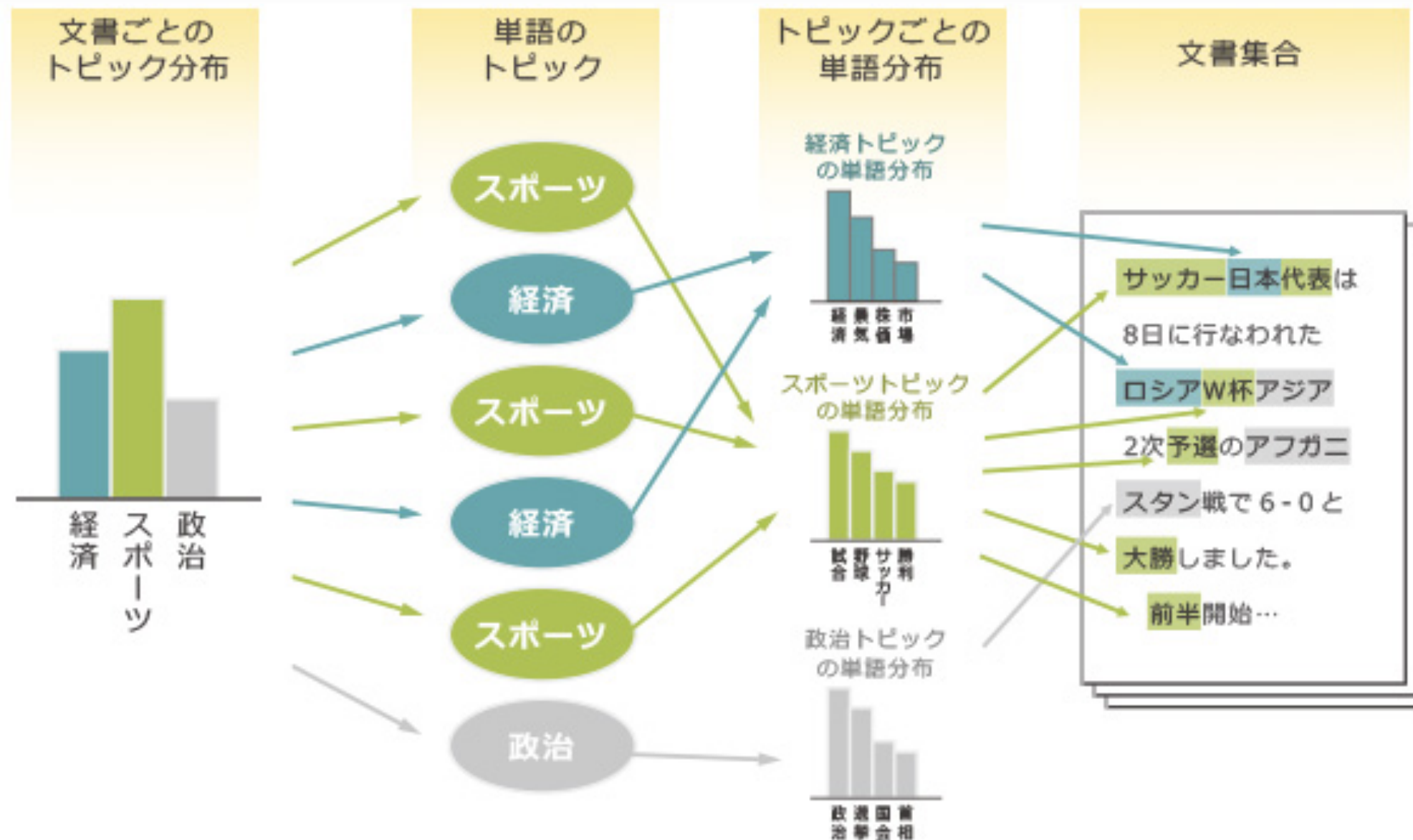
User Profile Building

- ユーザーが質問に回答するのは、ユーザーの興味と経験によるものとして考える
 - 質問に出現する単語 w に関するユーザーの興味と経験を確率で表す
 - λ はパラメータ

$$P(q|u) = \prod_{w \in q} P(w|\theta_u)^{n(w,q)} \quad (4)$$

$$P(w|\theta_u) = (1 - \lambda) \underset{\text{ユーザーの興味}}{P_{interest}(w|\theta_u)} + \lambda \underset{\text{ユーザーの経験}}{P_{expertise}(w|\theta_u)} \quad (5)$$

LDAとは



- 単語分布：トピック k で単語 w が出現する確率の分布
- トピック分布：ある文書を構成するトピック割合の分布

Modeling User Topical Interest

- ユーザーの過去の投稿を利用して，ユーザーの興味をモデル化する
 - 単語のマッチングだけでは，単語の意味を考慮できないためLDAによりトピックを利用する

$$P_{LDA}(w|\hat{\theta}, \hat{\phi}, \theta_u) = \sum_{z=1}^K P(w|z, \hat{\phi}) P(z|\hat{\theta}, \theta_u)$$

P_{interest}

- $P(w|z, \phi)$: トピック z における単語 w の出現確率 ϕ
- $P(z|\theta, \theta_u)$: ユーザー u の文書（過去の投稿）のトピック z のトピック分布
- W : ベストアンサーを予測する質問に出現する単語

Modeling User Topical Expertise

- ユーザーの経験を専門性を測定するために、回答につけることのできる投票機能を利用する
 - 高い評価を得ている回答を多く投稿しているユーザーは回答の質が良くベストアンサーを取りやすいと考えられる

$$P_{expertise}(w|\theta_u) = P_{expertise} \sum_z P(w|z, \hat{\phi}) P(z|\theta_u)$$

トピック z における
ユーザー u の専門性

Modeling User Topical Expertise

Procedure LearnUserExpertise

Inputs:

U : User Profiles

VM : Voting Matrix, contains voting information for each answer.

TM : Tassign Matrix

K : Topic Number

Output:

EM : User Topical Expertise Matrix

Methods:

1: For each user u in U

2: Let EV = a vector contains u 's expertise on each topic.

3: Let QA = answers provided by u and their corresponding questions.

4: For ($i = 1; i \leq |QA|; i++$)

5: Let $Score$ = the expertise of u on the question $QA[i]$;

6: $Score = getScore(VM, u, QA[i])$ $QA[i]$ についている評価の数を取得

7: Let $Probability$ = a vector contains the topic distribution of question $QA[i]$

8: $Probability = getProbability(Tassign, u, QA[i])$ $QA[i]$ のQのトピック分布を取得

9: For ($j = 1; j \leq K; j++$)

10: Add $Probability[j] \times Score$ to $EV[j]$

11: Add EV to EM

12: $NormalizeByTopic(EM)$

13: Output EM

各ユーザーのトピックごとの
専門性を計算する

$Q[i]$ のトピック j のトピック分布の割合と評価の数を掛け合わせ
スコアとする

Fig. 3. Learning user topical expertise

StackOverflowの評価機能

ホーム

質問

タグ

ユーザー

Anacondaに同梱されていないライブラリを入れるには？

質問する

▲ 1 私はWindowsなおかつ、Anacondaという汎用的なPython処理系を導入しています。でも、Anacondaには入っていないライブラリもありますヨネ。（例えばKivyやpygameとか）そのようなライブラリの導入方法が知りたいです。

▼ python windows python3 anaconda

★ 共有 この質問を改善する

編集日時: 1月18日 11:01

 nekketsuuu ◆
8,336 ● 5 ● 14 ● 48

質問日時: 1月16日 10:30

 keito940
77 ● 11

コメント追加

2 件の回答

アクティブ 古い順 票

▲ 5 Windows かつ Anaconda 環境の場合の一般的な質問のようなので、一般的な方法を紹介します。

▼ 1. [Anaconda Cloud](#) に無いか確認する。

▲ Anaconda は、公式では配布していないパッケージを第三者が配布するための仕組みとして、Anaconda Cloud を運営しています。有名なライブラリであれば、誰かが既にパッケージとして配布してくれているかもしれません。

▼ たとえば今 Kivy で検索すると <https://anaconda.org/krisvanneste/kivy> が出てきます。多少古いですが、Windows 64 bit、Python 2.7 向けの Kivy 1.8.0 でよければここからインストールできます。サ

質問を投稿 8 か月前

閲覧回数 1,595 回

アクティブ 今日

関連する質問

- 0 anacondaについて
- 2 ターミナルでのanacondaのインストールについて
- 0 AnacondaでインストールしたPython3がBracketsで認識されない
- 0 tensorflowをanacondaのJupyterで使いたいです
- 0 Neovim&deople-jediでAnacondaを使いたいのですが、エラーを消せません。
- 3 WindowsでanacondaのPythonに引数が渡せない？
- 0 anacondaでインストールしたIDLEがモジュールを認識しない
- 0 anacondaのmakeファイルについて

Prior Information of Answerers

- アクティブなユーザーの方が回答する確率が高いと考えられるため、回答する確率に重み付けを行う
 - 質問が投稿された時刻から、ユーザー(u)が最後に投稿した時刻を引いた値の逆数を利用
 - expは先行研究[7]に基づく

$$P(u) = Activity(u) = exp^{-\boxed{(t_q - t_u)}}$$

質問にユーザー(u)が回答する確率

最近、回答を投稿したユーザーほど回答する確率が高くなる

データセットと実験の設定

- StackOverflowのデータ
 - 2012年1月~12月
 - C#タグのついた質問を対象（C#タグの質問が一番多かったため）
 - 1月~11月をトレーニングデータ 12月をテストデータとする

Table 1. Data statistics

Tag	Questions	Askers	Answers
C#	99,166	37,363	191,338

- LDAのトピック数は100

リサーチクエスチョン

- RQ1
 - ベースライン手法（tf-idf手法）よりも提案手法は質問に対するベストアンサーを提供する回答者の予測精度が高いか？
- RQ2
 - パラメータ λ は精度に影響するか？

ベースライン手法

- TF-IDFに基づく手法

- 過去にユーザーuが回答した質問のtf-idfと新しく投稿された質問のコサイン類似度を算出
- 類似度が高いユーザーを回答者として推薦

$$s(\theta_u, q) = \frac{\sum_w tfidf(w, \theta_u) tfidf(w, q)}{\sqrt{tfidf(w, \theta_u)^2} \sqrt{tfidf(w, q)^2}}$$

質問qとユーザーuの
類似度

コサイン類似度

RQ1の結果

- RQ1: ベースライン手法と比較して提案手法は効果的か？

Table 2. Comparison of S@N rate

Approach	S@1	S@3	S@5	S@10	S@20	S@50	S@100
TF-IDF based	0.20%	0.32%	0.45%	0.65%	1.09%	2.17%	3.42%
<i>OurInterest</i>	1.18%	2.14%	2.66%	3.82%	5.53%	9.226%	13.76%
<i>OurCombine</i>	2.11%	4.27%	5.48%	7.76%	11.11%	16.63%	21.50%

RQ2の結果

- RQ2 : λ の値による影響はあるか？

Table 3. Varying value of parameter λ

λ	S@1	S@3	S@5	S@10	S@20	S@50	S@100
0.0	1.18%	2.14%	2.66%	3.82%	5.53%	9.226%	13.76%
0.3	2.08%	3.65%	4.74%	6.84%	9.53%	14.28%	18.92%
0.5	2.11%	4.27%	5.48%	7.76%	11.11%	16.63%	21.50%
0.7	2.14%	4.35%	5.71%	8.38%	11.86%	18.18%	23.06%
1.0	2.18%	4.15%	5.56%	8.33%	12.57%	18.34%	23.01%

まとめ

- CQAサイトの1つであるStackOverflowに着目して、新しく追加された質問に対する適切な回答者を推薦するシステムの提案を行った
- tf-idfベースの手法と比較実験を行い、有用性を示した

所感

- 基本的なLDAベースの手法が分かった
- 正解集合の作り方が参考になった
- 評価方法がtf-idf手法との比較で良いのか？
 - 既存手法との比較をしていない
- LDAのパラメータの設定でも精度が変化しそう
- ネットワーク手法とLDAを組み合わせた手法を詳細に調査する必要があるそう