

研究発表

福井 克法

背景

- 効率的なソフトウェア開発には適切な人材配置が重要
- 組織の規模がある程度大きくなると開発者すべての業務経験を人為的に管理することは困難



開発者検索の既存研究

	検索する対象	データセット	アプローチ
[Mockus et al 2003]	特定のモジュールに精通した開発者	バージョン管理システム	カウントベース
[Robbes et al 2013]	特定のモジュールに精通した開発者	バージョン管理システム	カウントベース
[Campbell et al 2003]	特定の技術に関する投稿をした開発者	メーリングリスト	HITS
[Dom et al 2003]	特定の技術に関する投稿をした開発者	メーリングリスト	HITS, PageRank, Power, Affinity
[Yang et al 2013]	質問に対する回答者	StackOverflow	LDA
[Yang et al 2014]	質問に対する回答者	StackOverflow	PMF
[Wang et al 2016]	質問に対する回答者	StackOverflow	LDA

StackOverflowにおけるエキスパート検索

- 背景

- StackOverflowのようなCQAサイトでは、毎日膨大な数の質問と回答が投稿されている
- 様々なドメインの質問が存在する

- 課題

- 現在のStackOverflowでは、質の低い回答やスパム回答が存在する
- 質問が、もっとも適切な回答をしてくれる人に届かない可能性がある。また、回答が集まるまで時間を要する。

既存手法のアプローチ

- トピックモデルベースのアプローチ[1][2]
 - 質問と回答のタイトルと本文を利用して，回答者を推薦する手法
- Tagベースのアプローチ[3]
 - 質問につけられるTagを利用して，回答者を推薦する手法

[1]L. Yang, M. Qiu, S. Gottipati, F. Zhu, J. Jiang, H. Sun, and Z. Chen, "Cqarank: jointly model topics and expertise in community question answering," in *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, 2013, pp. 99–108.

[2]Lin Wang, Bin Wu, Juan Yang, Shuang Peng, "Personalized recommendation for new questions in community question answering", *Advances in Social Networks Analysis and Mining (ASONAM) 2016 IEEE/ACM International Conference on*, pp. 901-908, 2016.

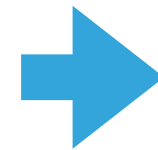
[3]B. Yang, S. Manandhar, "Tag-based expert recommendation in community question answering", *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, pp. 960-963, Aug. 2014.

Tagベース手法のアプローチ

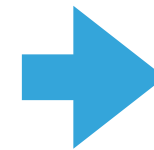
学習 (A3と対応する質問)

	Tag1	Tag2	Tag3	...
User1	3	4		
User2			5	
User2	1	1	1	
...				

Tagのスコアの平均値

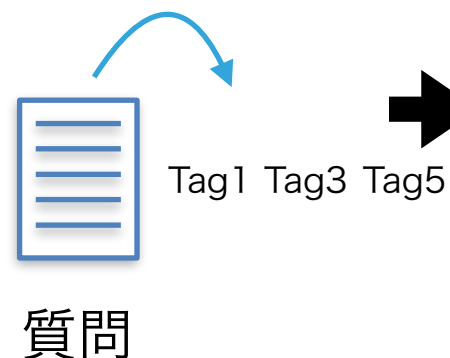


Probabilistic Matrix Factorization

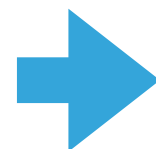


	Tag1	Tag2	Tag3	...
User1	3.2	4.3	1	
User2	2.2	2.3	5	
User2	1.1	0.9	0.9	
...				

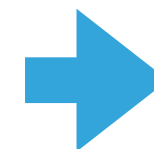
テスト (A5と対応する質問)



Res(u, q)



	Tag1	Tag2	Tag3	...
User1	3.2	4.3	1	
User2	2.2	2.3	5	
User2	1.1	0.9	0.9	
...				



予測

順位	ユーザ
1	User1
2	User3
3	User8
4	User3
5	User6
6	User9
...	...

$$ReS_{our}(u, q) = \frac{1}{N_t} \sum_{t=1}^{N_t} Ept(u, t)$$

研究の目的とアプローチ

- 目的

- 既存手法を改善し，精度向上

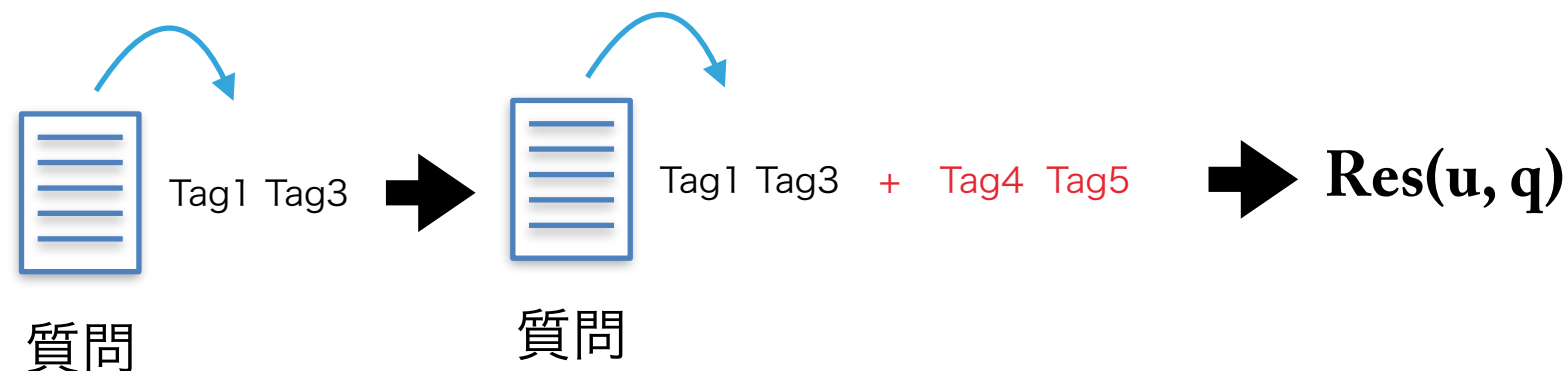
- アプローチ

- 既存手法の改善ポイント

- 質問にはTagが複数ついているとは限らない，Tag数が少ない場合と多い場合で精度が変わる可能性がある

- 改善案

- Tag数が少ない場合に，類似したTagを追加する



既存手法の改善案

- 分散表現を用いたアプローチ

- 質問に付けられたタグセットを1文として学習



- PMFの学習時のノイズ削減（分散表現で効果が出なさそうなら）

- 問題になり得る点

- 質問につけられるタグは質問者が任意でつけるため、必ずしも適切とは限らない（関係のないタグが付けられている可能性がある）

- 改善案

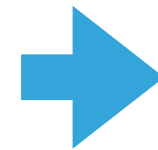
- pmf, word2vecの学習時に相関ルールマイニングなどを利用して、頻繁に一緒に使われているタグセットのみを使うようにしノイズをへらす

既存手法の実験の再現

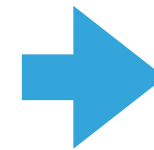
学習 (A3と対応する質問)

	Tag1	Tag2	Tag3	...
User1	3	4		
User2			5	
User2	1	1	1	
...				

Tagのスコアの平均値

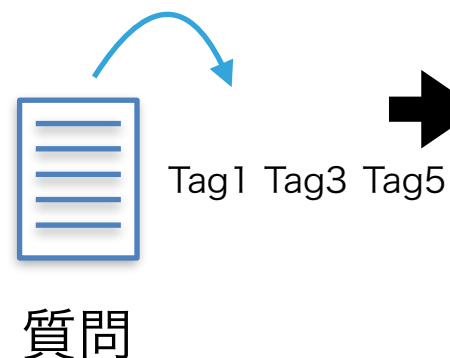


Probabilistic Matrix Factorization

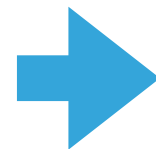


	Tag1	Tag2	Tag3	...
User1	3.2	4.3	1	
User2	2.2	2.3	5	
User2	1.1	0.9	0.9	
...				

テスト (A5と対応する質問)

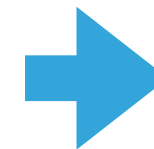


Res(u, q)



	Tag1	Tag2	Tag3	...
User1	3.2	4.3	1	
User2	2.2	2.3	5	
User2	1.1	0.9	0.9	
...				

$$ReS_{our}(u, q) = \frac{1}{N_t} \sum_{t=1}^{N_t} Ept(u, t)$$



予測		正解	
順位	ユーザ	順位	ユーザー
1	User1	1	User1
2	User3	2	User4
3	User8	3	User8
4	User3	4	User4
5	User6	5	User5
6	User9		
...	...		

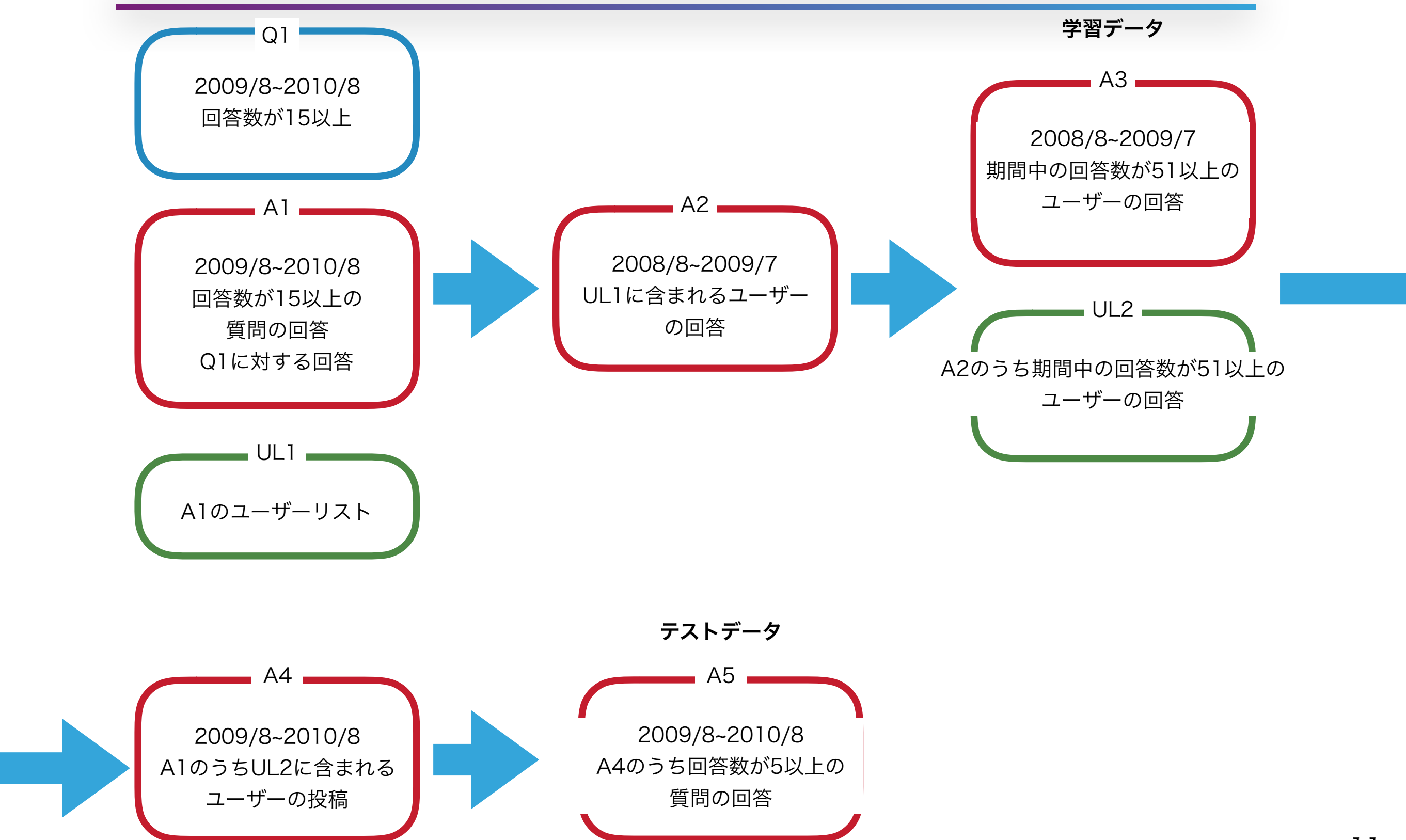
nDCGにより精度を算出 9

データセットの分割

- データセットの分割手順

1. 2009/8~2010/8の回答数が15以上ある質問の集合を, Q1とする
2. Q1に対応する回答の集合をA1として, A1を投稿したユーザーの集合をUL1とする
3. 2008/8~2009/7の全ての回答のうちUL1に含まれるユーザーの回答集合をA2とする
4. A2のうち, 期間内に回答数が50よりも多いユーザーのみの回答をA3とし, A3を投稿したユーザーの集合をUL2とする
5. A1のうちUL2に含まれるユーザーの投稿の集合をA4とする
6. A4のうち, 回答数が5よりも大きい質問の回答のみをA5とする
7. A3とA3に対応する質問を訓練データ, A5とA5に対応する質問をテストデータとする

データセットの分割



進捗

- 完了済み
 - データの整形および、学習・テスト用データの準備
- 進行中
 - 既存手法：タグベースPMFの実装
 - ・ 最終の評価部分 nDCGの部分に苦戦中...
- 残りタスク
 - 精度を確認した後、改善案を適用
 - 最新のLDAベースの手法の実装と比較

Word2Vecの学習モデル

- 2008/8~2009/7の全ての質問のタグセット
 - ウィンドウサイズ:5 次元数:100
- 出力例（類似度上位10件）

jquery	database	database + python	python
<prototypejs>	<rdbms>	<rdbms>	<turbogears2>
<mootools>	<database-design>	<data-modeling>	<perl>
<dhtml>	<query-optimization>	<sqlalchemy>	<django-flatpages>
<javascript-events>	<denormalization>	<django-models>	<django-views>
<jquery-ajax>	<data-modeling>	<freebase>	<python-3.x>
<jquery-plugins>	<ddl>	<database-design>	<django-authentication>
<iframe>	<entity-attribute-value>	<denormalization>	<templatetags>
<yui>	<normalization>	<plpgsql>	<django-urls>
<scriptaculous>	<foreign-keys>	<postgresql>	<django-models>
<lightbox>	<multi-tenant>	<foreign-keys>	<reverse>