

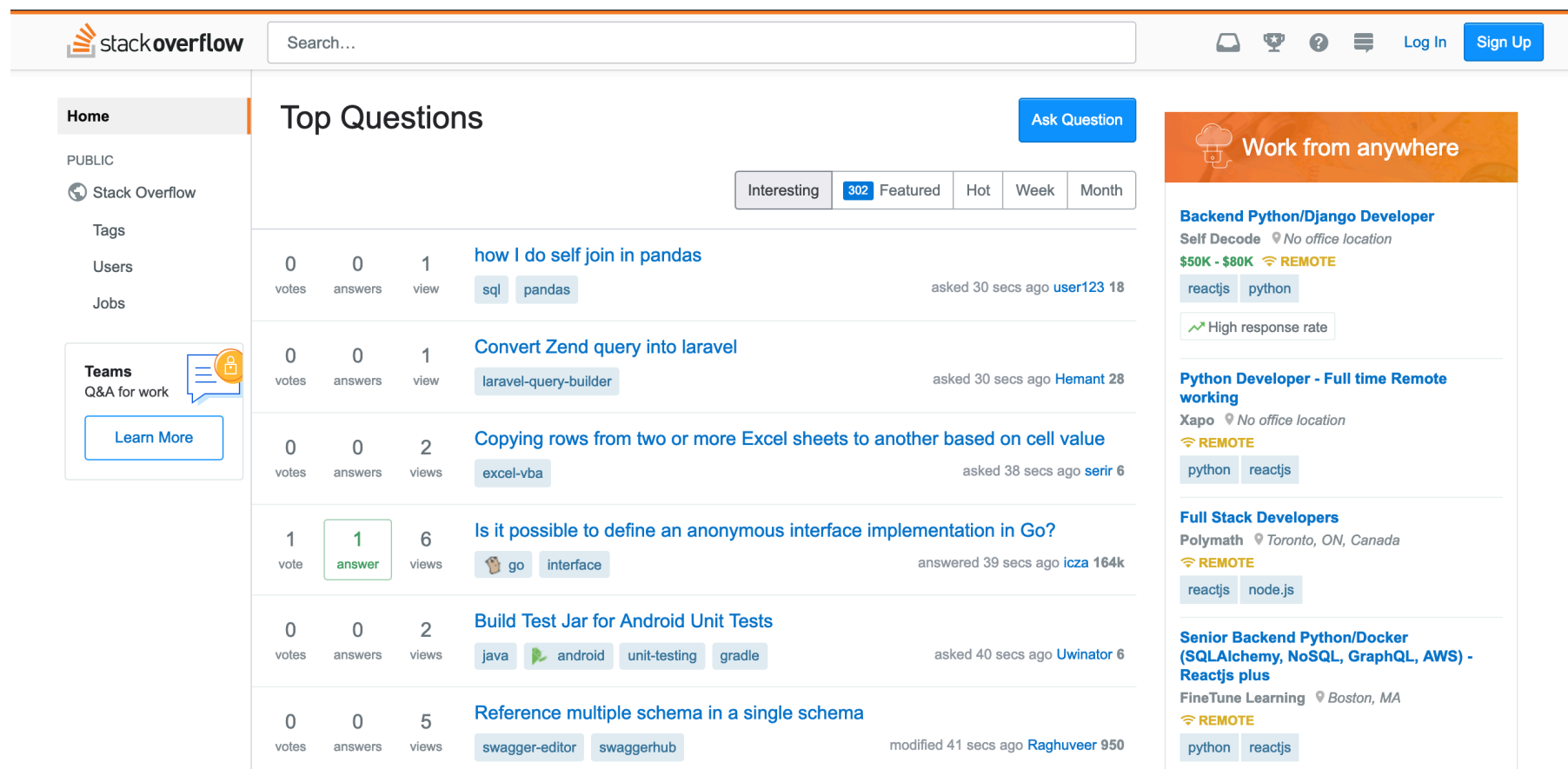
QAサイトにおける協調フィルタリングと 単語の分散表現を利用した解答者推薦手法

オープンソースソフトウェア工学研究室

M1 福井 克法

背景

- 開発者が技術的な問題を解決するために、QAサイトは必要不可欠な存在になっている
- 近年プログラミングに関する話題を扱う StackOverflowが人気を集めている



QAサイトの課題

- 投稿される解答は質が高いことが望ましいが、質の低い解答も多数投稿されている
- StackOverflowでは全質問の約半数が未解決である[1]
- 未解決の質問は増加傾向にある[2]

[1]L. Nie, X. Wei, D. Zhang, X. Wang, Z. Gao, Y. Yang, "Data-driven answer selection in community qa systems", IEEE Trans. Knowl. Data Eng, vol. 29, no. 6, pp. 1186-1198, June 2017.

[2]Muhammad Asaduzzaman , Ahmed Shah Mashiyat , Chanchal K. Roy , Kevin A. Schneider, Answering questions about unanswered questions of stack overflow, Proceedings of the 10th Working Conference on Mining Software Repositories, May 18-19, 2013, San Francisco, CA, USA

関連研究

- ベストアンサー予測[3][4]
 - 役に立つ解答が分からない質問者の支援を目的
 - 投稿された質問の解答の中から最も良い回答を予測する
- 解答者推薦
 - 質の高い解答の投稿を増加させることが目的
 - 新しく投稿された質問に対して、適切な解答者を推薦する

[3]Calefato, Fabio & Lanubile, Filippo & Novielli, Nicole. (2018). An empirical assessment of best-answer prediction models in technical Q&A sites. Empirical Software Engineering.

[4]Fabio Calefato , Filippo Lanubile , Nicole Novielli, Moving to Stack Overflow: Best-Answer Prediction in Legacy Developer Forums, Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, September 08-09, 2016

解答者推薦手法

- トピックモデルベースの手法[5][6]
 - 質問本文などのテキストデータを扱う
- 協調フィルタリングを利用した手法[7]
 - 質問に付けられたタグのみを利用
 - モデルの学習時間が短い
 - ・ 1イテレーションあたりトピックモデルでは121.43秒 協調フィルタリングの手法（PMFアルゴリズム）では1.03秒

[5]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.

[6]Lin Wang , Bin Wu , Juan Yang , Shuang Peng, Personalized recommendation for new questions in community question answering, Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM '16)., pp. 901-908.

[7]Baoguo Yang and Suresh Manandhar. 2014. Tag-based expert recommendation in community question answering. In Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM '14). IEEE Press, pp. 960-963.

StackOverflow

質問

How long is a 302 redirect saved in browser?



20



3

Due to a misconfiguration of our webserver the main domain sent a 302 redirect to a new location. We fixed that issue. When emptying the browser cache everything works fine now.

For the "normal" client who does not empty his cache: How long is the 302 redirect kept in the browser?

I'm looking for **specific cache times** (if any) for each of the major browsers (Chrome, Firefox, Safari, Edge, IE 12) under default settings.

[redirect](#)

[browser](#)

[browser-cache](#)

[http-status-code-302](#)

質問につけられたタグ



[Chuck Le Butt](#)

27.7k ● 45 ● 149 ● 225



[Ole Albers](#)

3,869 ● 6 ● 35 ● 102

asked Aug 31 '12 at 9:45

This question has an open [bounty](#) worth **+150** reputation from [Chuck Le Butt](#) ending in **6 hours**.

The question is widely applicable to a large audience. A detailed canonical answer is required to address all the concerns.

Looking specifically for how each of the major browsers behaves with default settings. Browsers SHOULDN'T cache at all... but DO THEY?

[Redirect caching deep dive](#) article by Steve Souders contains a link to a relevant web tests. – [Leonid Vasilev](#) 8 hours ago

[add a comment](#)

3 Answers

[active](#)

[oldest](#)

[votes](#)



23



It shouldn't be cached at all unless there's also a `Cache-Control` or `Expires` header returned by the web server. According to [RFC 2616, section 10.3.3 302 Found](#)

The requested resource resides temporarily under a different URI. Since the redirection might be altered on occasion, the client SHOULD continue to use the Request-URI for future requests. This response is only cacheable if indicated by a Cache-Control or Expires header field.

解答

解答に対するスコア

answered Aug 31 '12 at 9:53

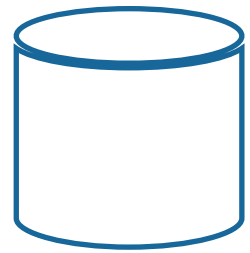


[Jon Lin](#)

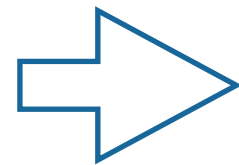
123k ● 24 ● 166 ● 182

協調フィルタリングベースの先行研究

学習



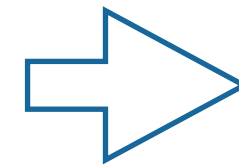
学習データの収集
過去の質問と回答データ



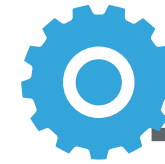
ユーザー

タグ		Tag1	Tag2	Tag3
ユーザー		1		1
			3	
		1		2

各解答者がタグ毎に得た
スコアの平均値を計算



協調フィルタリング

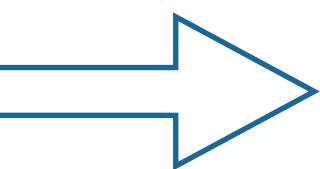


ユーザー

タグ		Tag1	Tag2	Tag3
ユーザー		1.3	0.8	1
		1	3	1.4
		1.3	0.8	2

評価値の予測

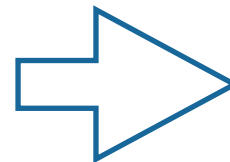
予測



新しい質問

Tag1 Tag3 Tag5

➡ **Res(u, q)**



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

ユーザーuごとにタグtのスコアを算出

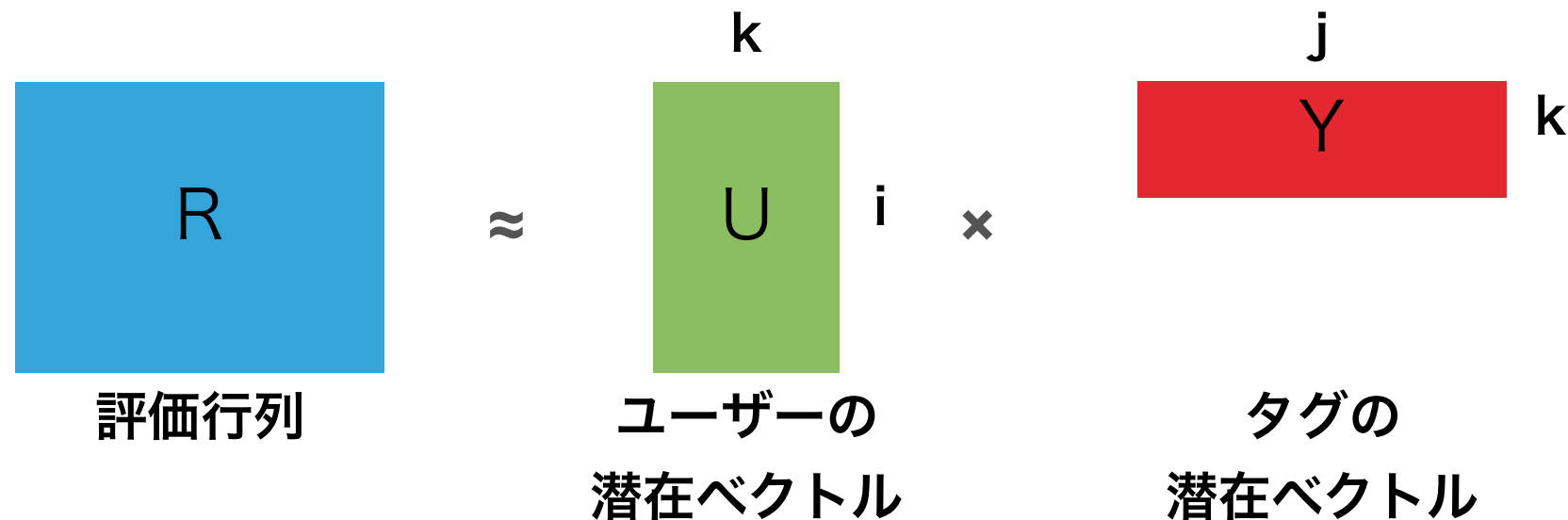
順位	ユーザー
1	User1
2	User3
3	User8
4	User3
5	User6
6	User9

各ユーザーのスコアに応じてランク付け

協調フィルタリング (PMF)

- PMF(Probabilistic Matrix Factorization)

- 行列分解を確率モデル的に考える手法



- 評価行列 R に近似するように潜在ベクトル U, Y を求める

$$L(S, U, Y) = \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^N \mathbb{I}_{ij}^S (s_{ij} - U_i^T Y_j)^2 + \frac{\lambda_U}{2} \sum_{i=1}^M \|U_i\|_{\mathcal{F}}^2 + \frac{\lambda_Y}{2} \sum_{j=1}^N \|Y_j\|_{\mathcal{F}}^2$$

評価の有無 $\mathbb{I}_{ij}^S$ 正解の評価点 s_{ij} 予測の評価点 $U_i^T Y_j$

ユーザーの潜在ベクトル U_i タグの潜在ベクトル Y_j

目的関数 L で表せる
2乗誤差+正則化項が
最小となる U, Y を求める

先行研究の問題点

- タグは質問者が任意で付けるため、タグに表記ゆれがある
 - バージョンによる違いなど
 - Java, Java8, Java9
- 質問者が初心者の場合、StackOverflowであまり利用されていないタグ、一般的でないタグが付けられている可能性がある
 - 出現頻度が低いタグの場合、協調フィルタリングの精度が低下する可能性がある

本研究の目的・アプローチ

- 目的

- 既存の協調フィルタリングを利用した手法を改善することで推薦精度の向上を目指す

- アプローチ

- タグをクエリ拡張することで精度の変化を確認する
 - 類似度の高いタグをクエリ拡張することで、推薦のスコアリングに使うタグを増やす

単語の分散表現を利用したクエリの拡張

- アプローチ
 - 単語の分散表現を利用して，入力タグと類似度の高いタグを取得し拡張する
- 単語の分散表現を利用したモデル
 - 単語の分布仮設に基づく手法
 - 類似した文脈に出現する単語の意味は似ている
 - I want eat to an **apple** everyday.
 - I want eat to an **orange** everyday.
 - 単語の分散表現の獲得
 - Word2Vec, FastTextを利用

Word2Vec

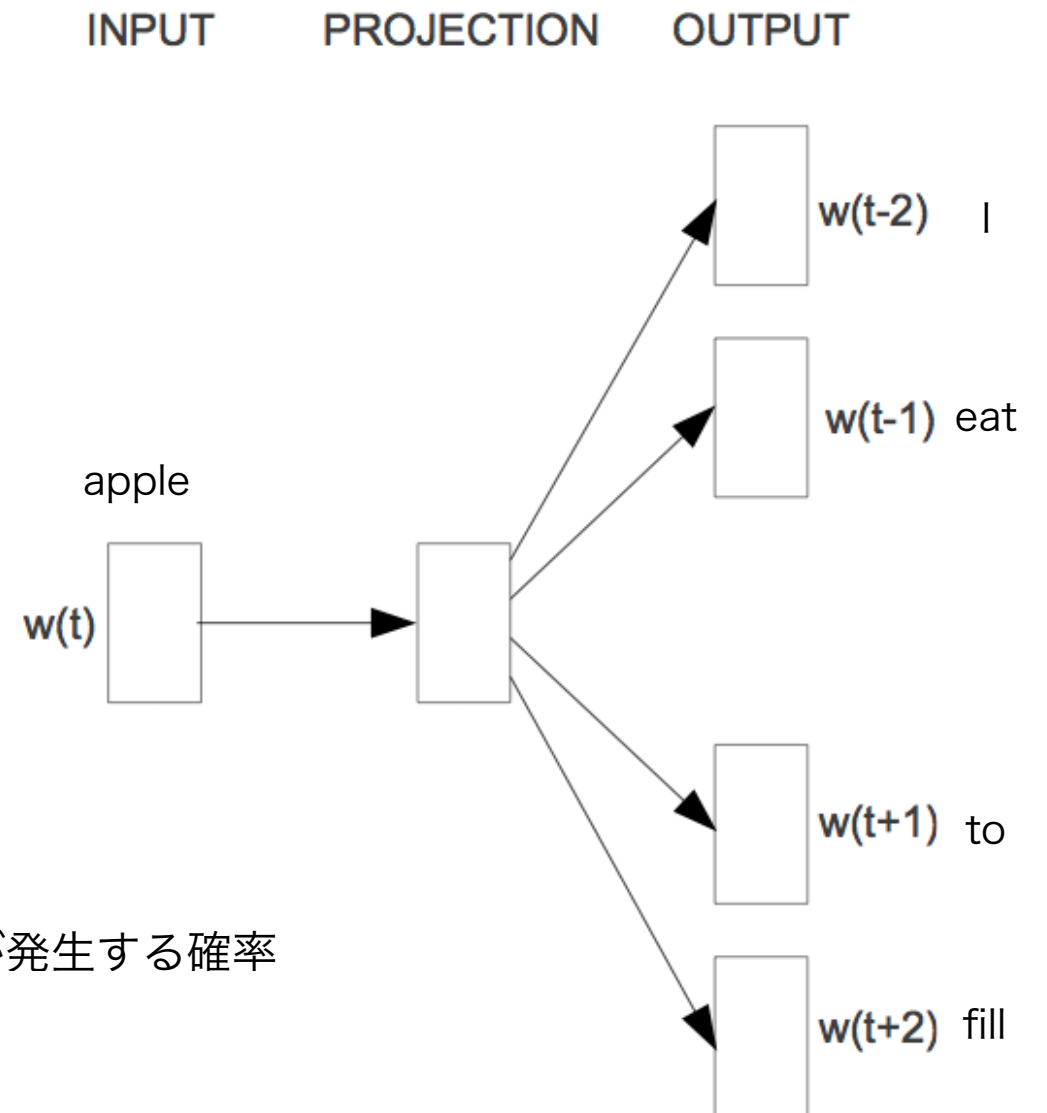
- Word2Vec
 - 2層のニューラルネットワーク
 - Skip-gramモデルを使用

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} | w_t)$$

平均対数確率 w_t : 単語
 T : 学習単語の総数
 c : 学習単語の周辺単語数
 $p(w_{t+j} | w_t)$: w_t のとき w_{t+j} が発生する確率

$$p(w_O | w_I) = \frac{e^{(v'_{w_O} \Gamma v_{w_I})}}{\sum_{w=1}^W e^{(v'_{w_O} \Gamma v_{w_I})}}$$

ソフトマックス関数 w_I : 学習単語
 w_O : 学習単語の周辺単語



Skip-gram

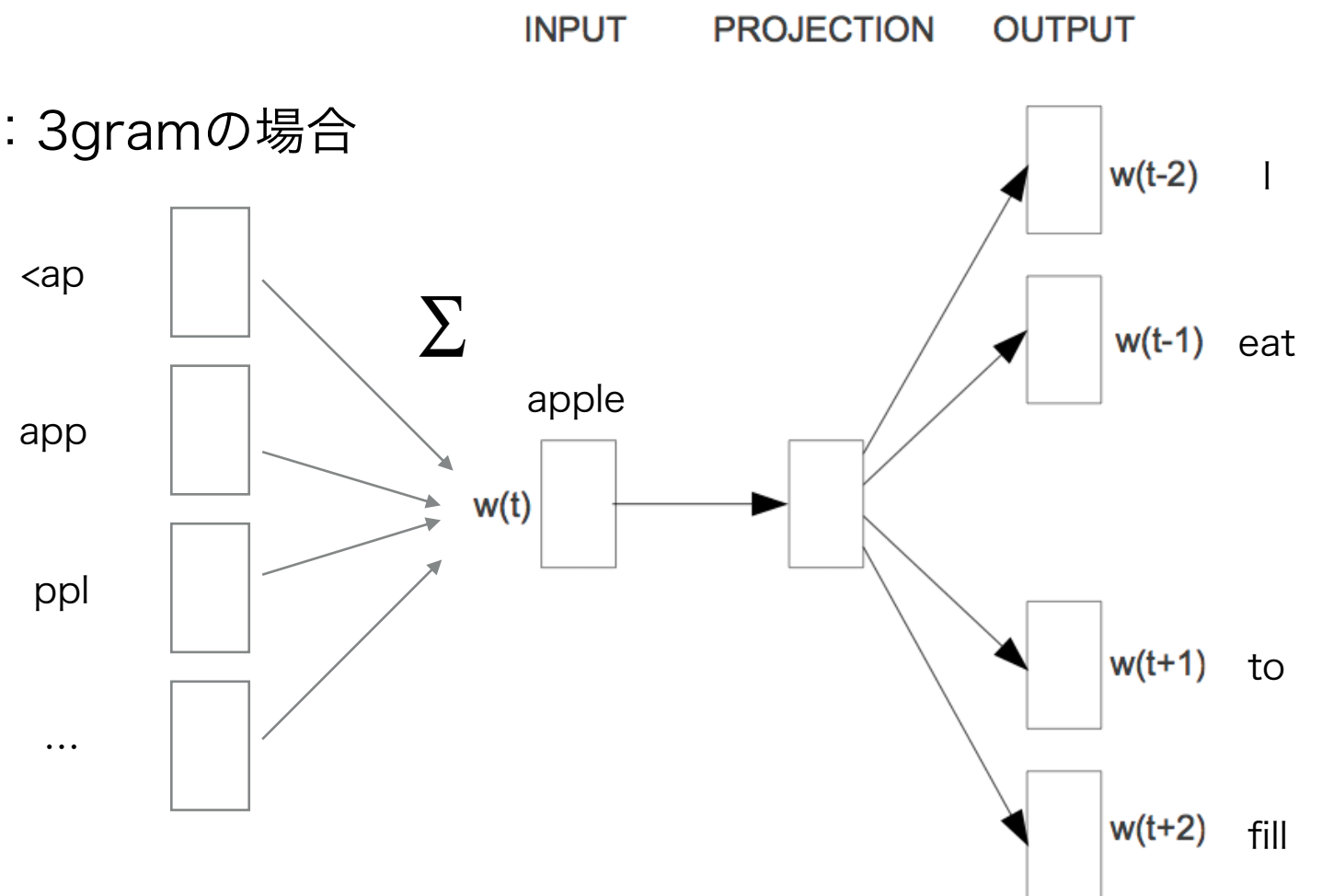
FastText

- Word2VecにSubwordモデルを取り入れたもの
 - 各単語ごとに単語ベクトルを計算
 - ・ 単語を構成する character n-gramのベクトルの和

$$s(w, c) = \sum_{g \in \mathcal{G}_w} \mathbf{z}_g^\top \mathbf{v}_c.$$

$\mathbf{z}_g$: 単語wの文字n-gram
・ に対する特徴ベクトル

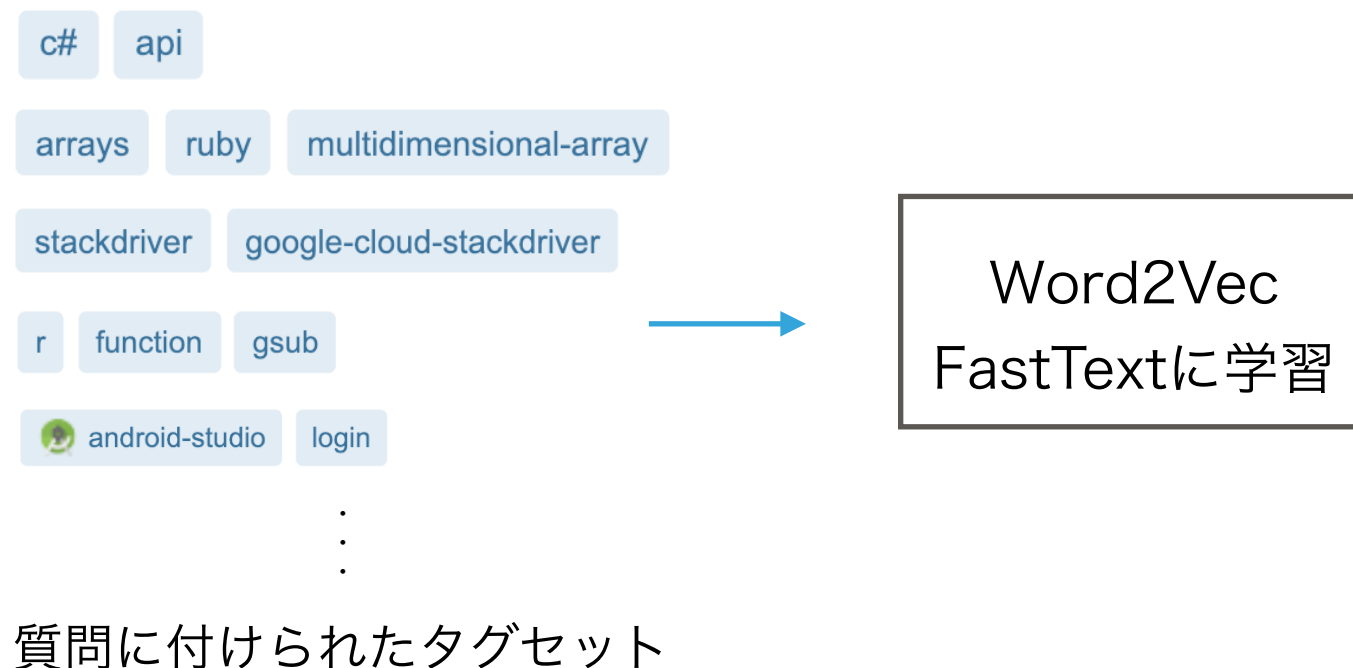
例：3gramの場合



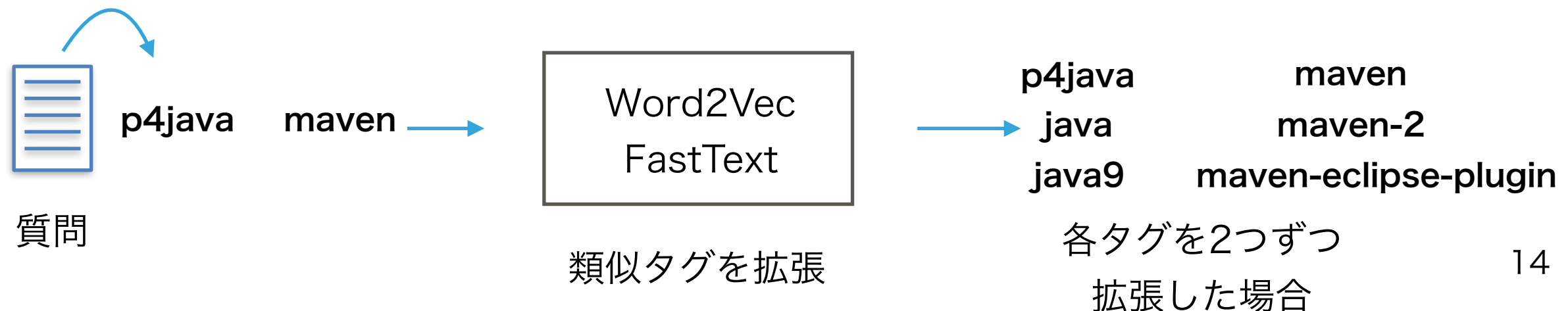
単語の分散表現を利用したクエリの拡張

- 質問に付けられたタグセットを学習し，近い意味のタグを取得

学習



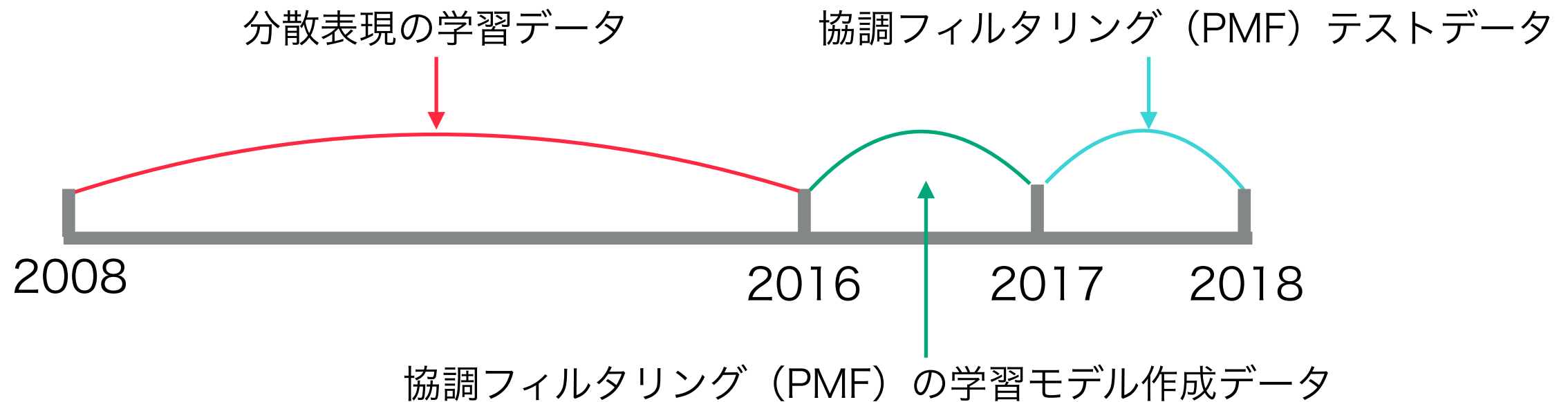
クエリ拡張



データセット

- StackOverflowの質問と解答データ
 - 2008~2017年
- PMFの学習と評価に利用するデータ
 - 2015~2017年
 - PMF学習データ：2015~2016年
 - 評価データ：2016~2017年
- 単語の分散表現の学習データ
 - 2008~2015年
 - 質問に付けられたタグセット：800万セット

データセットの分割



- データのフィルタリング(2016~2018)

- 学習データ

- ・ 解答数が1以上の質問とその回答
 - ・ 期間中解答数が50以上のユーザー

- テストデータ

- ・ 学習データに含まれるユーザーが6人以上解答している質問と学習データに含まれるユーザーの解答

評価指標 (nDCG)

- 予測ランキングがどの程度正しいかを評価する指標

$$DCG = rel_1 + \sum_{i=2}^n \frac{rel_i}{\log_2 i}$$

$$nDCG = \frac{DCG}{DCG_{ideal}}$$

i : 順位
rel : 適合度

例

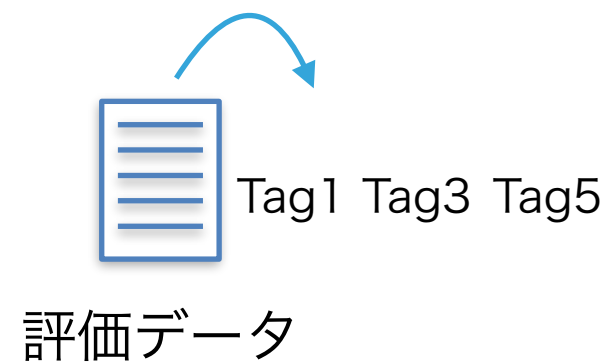
ユーザー	正しい順位	rel(適合度)	予測順位	rel(適合度)
user1	1	6	1	6
user2	2	5	5	2
user3	3	4	3	4
user4	4	3	4	3
user5	5	2	2	5
user6	6	1	6	1

DCG_ideal = 40.122
DCG = 34.448
nDCG = 0.858

評価の方法

- 評価内容

- 実際にユーザーが得たスコアのランキングと予測スコアによるランキングがどの程度一致しているかを評価する



質問に解答者	実際スコア	順位
user1	5	1
user2	4	2
user3	3	3
user4	3	3
user5	1	5
user6	0	6
user7	0	6

正解ランキング

一致度を評価



user1~7が得る
スコアを予測

手法による
スコアの予測

質問に解答者	予測スコア	順位
user1	8	1
user2	2	2
user3	0	5
user4	1	3
user5	0.5	4
user6	0	5
user7	0	5

予測ランキング

結果

各タグに付きクエリ拡張	ndcg_fasttext	ndcg_word2vec
ベースライン	0.82498	
1つ拡張	0.83398	実験中
2つ拡張	0.82578	
3つ拡張	0.82775	
4つ拡張	0.83133	

- 評価用の質問57件に対してスコアを算出
- 各10回ずつ試行した場合のndcgスコアの平均

今後の予定

- データのとり方を変えることで、評価データを増やす
- 試行回数を増やす
 - 実験用プログラムの改良（高速に実行できるように）
- 別の評価指標の導入（ピアソン順位相関係数 etc）
- **精度が向上するタグと低下するタグの分析**