

大規模組織における適切な人材配置のための 開発者検索手法

福井 克法

背景

- ソフトウェア開発は大規模化している
- ソフトウェア開発を円滑に行うためには、開発者の特性を理解して過去の実績や関わった開発を把握しておくことが望ましい

ソフトウェア開発現場での問題点

- 大規模な開発現場では、各開発者の今までに関わったプロジェクトや開発をすべて把握することは難しい
 - ▶ プロジェクトに適任の開発者を割り当てることが難しくなる
 - ▶ 企業が案件を提案する際に、開発者の知識や経験を生かせずに案件を逃してしまう。

先行研究の問題点

- ソースコードの変更履歴に着目したモジュール単位での適任開発者を推薦する手法が提案されている[1][2]
- 問題点
 - ▶ しかし、実際のソフトウェア開発では実装以外の開発工程も存在する。
 - ▶ ソースコードを各工程以外の開発工程が考慮されていない

[1] Audris Mockus and James D. Herbsleb. Expertise browser: A quantitative approach to identifying expertise. In Proceedings of the 24th International Conference on Software Engineering, ICSE '02, pp. 503–512, New York, NY, USA, 2002. ACM.

[2] Romain Robbes and David R'othlisberger. Using developer interaction data to compare expertise metrics. In Proceedings of the 10th Working Conference on Mining Software Repositories, MSR '13, pp. 297–300, Piscataway, NJ, USA, 2013. IEEE Press

本研究の目的

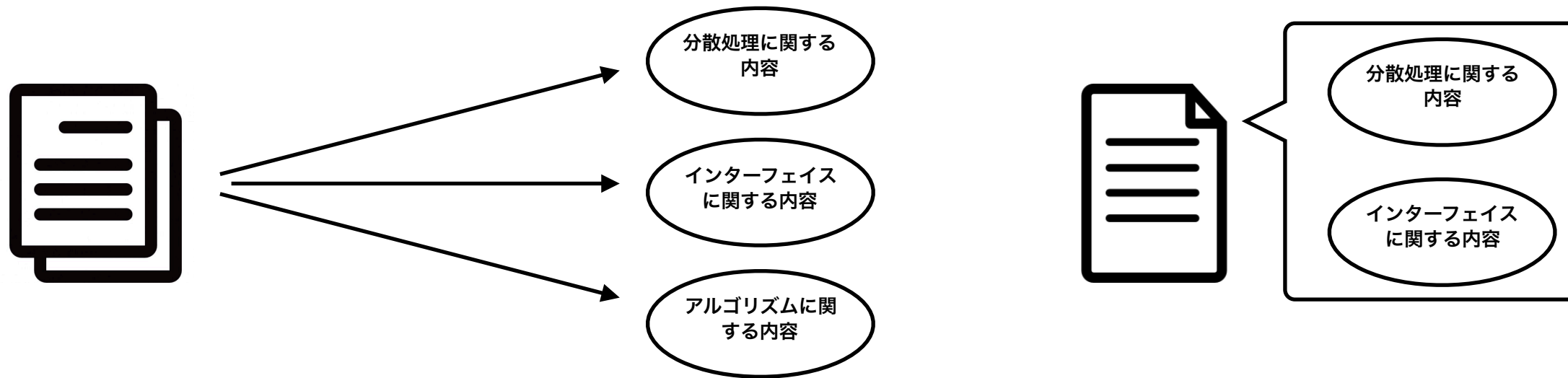
- 要件定義や設計などソースコードを変更しない工程も含めた、開発に関わっている人物を検索する手法を提案する

コミュニケーションデータの利用

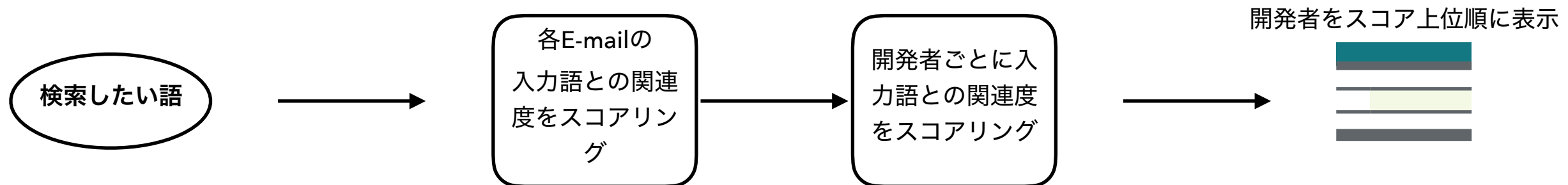
- ソフトウェア開発を行う上で、開発者・開発に関わっている人物はコミュニケーションツールを使用する.
 - ▶ メーリングリスト・Github・Slackなど
 - ▶ コミュニケーションデータには開発者の今までに関わった開発に関する情報が含まれていると考えられる
- 本研究では、代表的に使われているメーリングリストに着目する

提案手法の概要

1. メーリングリスト全体に含まれる内容・各E-mailの内容を特定

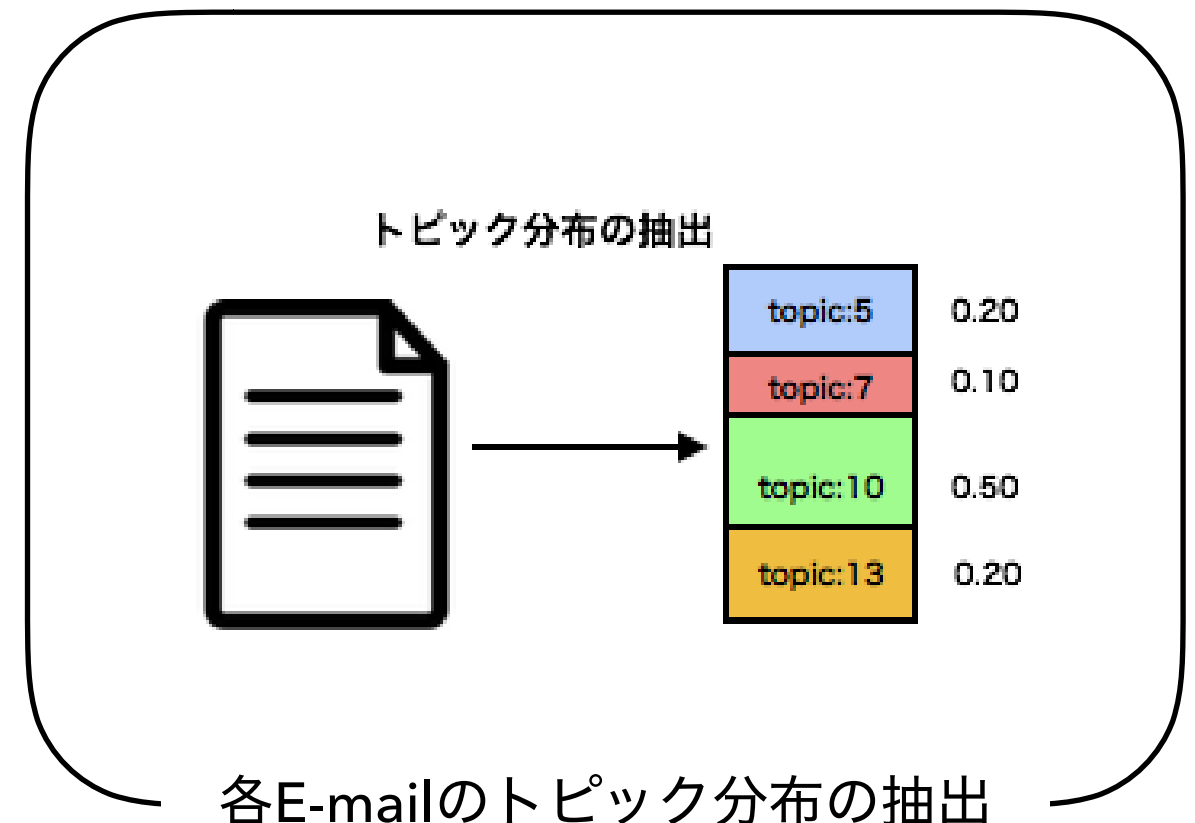
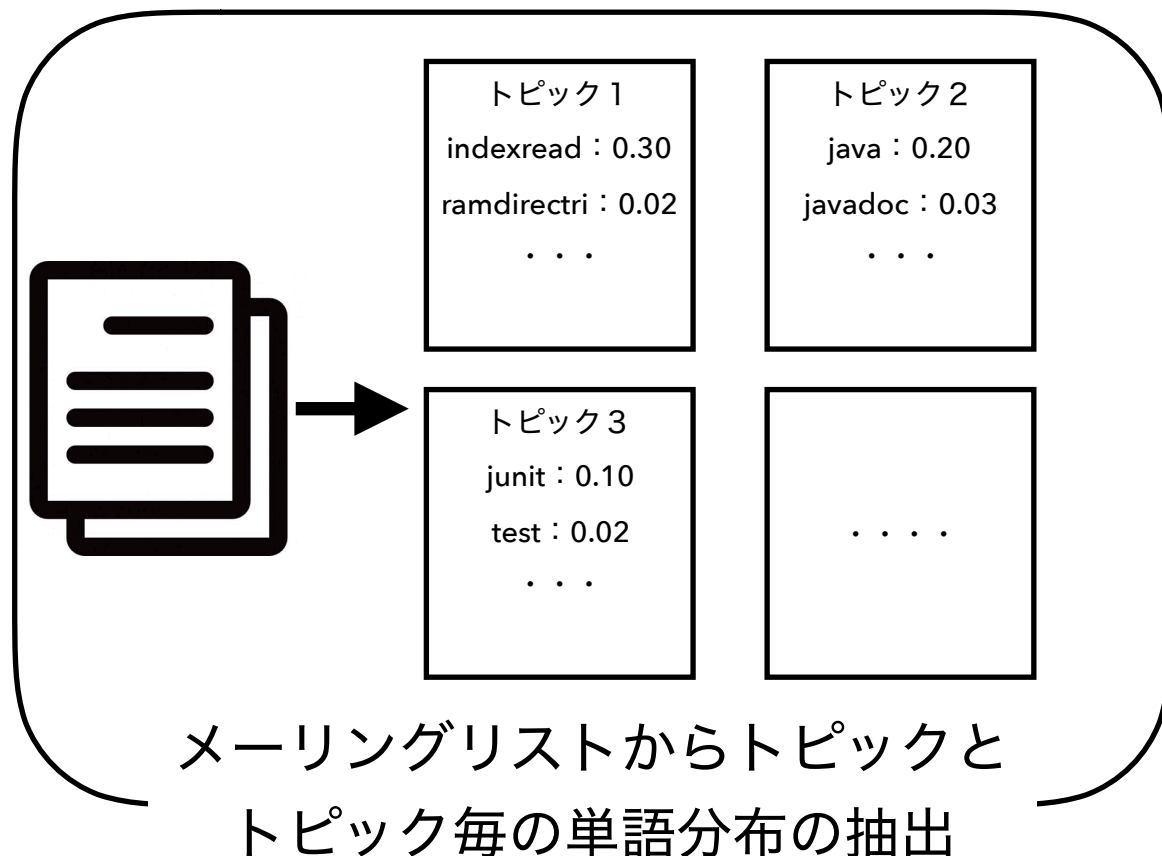


2. 技術用語や特定のプロジェクトに関する用語を入力として、開発者ごとに送信したメールと入力語との関係をスコアリング



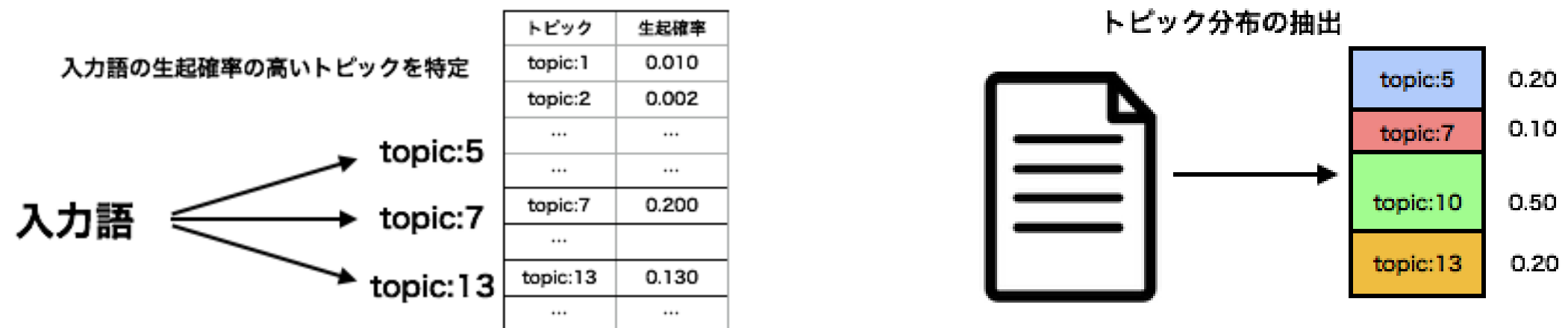
E-mailから内容と用語の抽出

- トピックモデルの潜在的ディリクレ分布（LDA）を用いる
- メーリングリストに含まれる話題，各E-mailの内容をトピック分布を利用して抽出



開発者のスコアリング

- 入力語との関連度をトピックの単語分布と各E-mailのトピック分布を利用して算出する



- 入力語に対する各E-mailのスコアリング

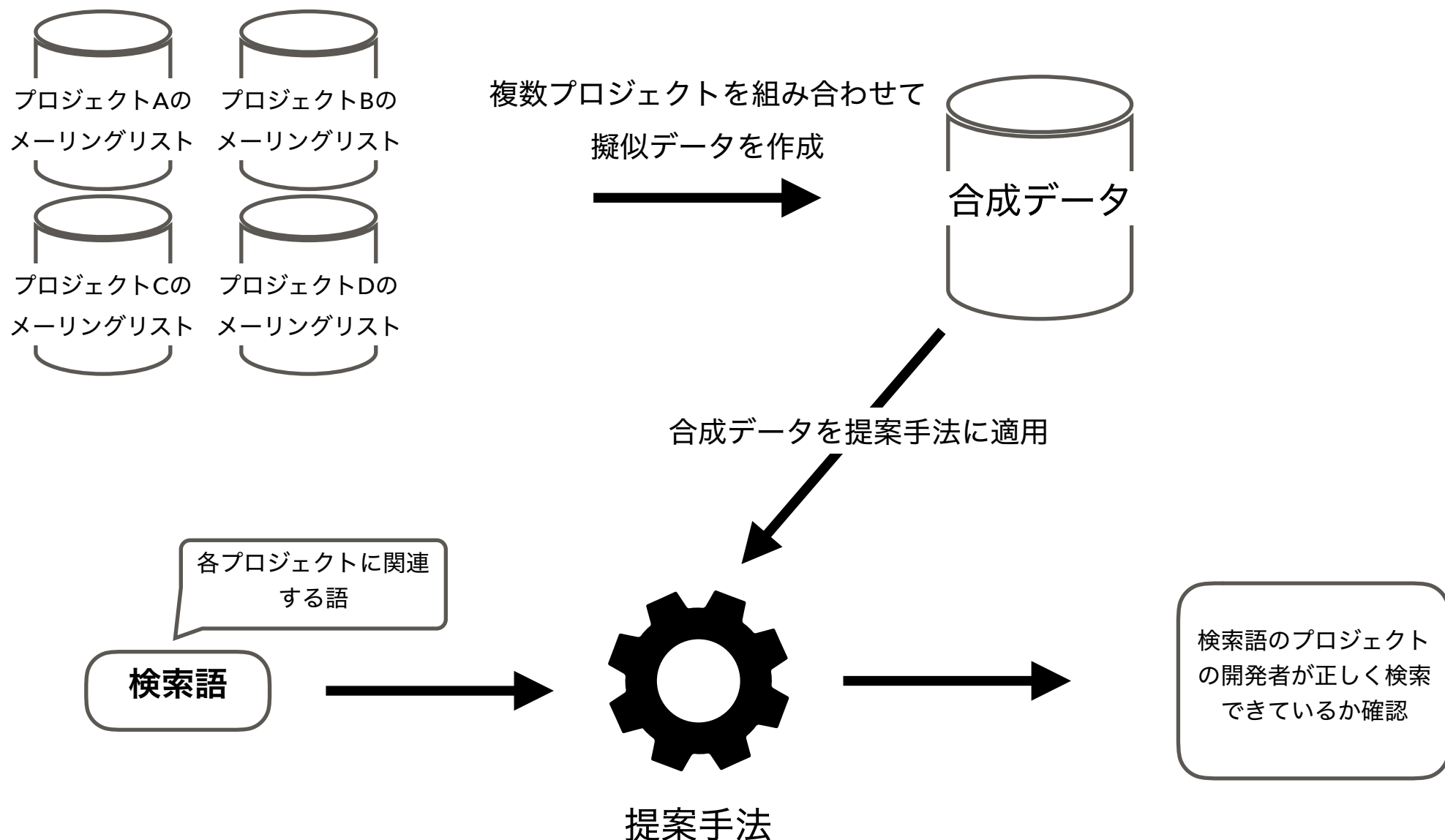
$$DocumentScore_d = \sum_{k=topic}^n \text{入力語の生起確率}_n \times \text{トピック分布}_n$$

- 入力語に対する各開発者のスコアリング

$$AuthorScore = \sum_{d=DocumentNumber}^n DocumentScore_d$$

実験の概要

- 複数プロジェクトのメーリングリストを合成することで、複数の内容が混ざったメーリングリストを人工的に作成
- 各プロジェクトに関連する語を入力としたとき時、関連語のプロジェクトの開発者が正しく検索されているか確認する



実験手順

1. OSSプロジェクトを組み合わせて合成データを作成
2. 各OSSプロジェクトのドメイン用語を抽出
 - ドメイン用語：プロジェクト内で使われる特徴的な単語
3. 合成データに対して手法を適用
4. 入力ドメイン用語のプロジェクトの開発者を正解集合として、スコアの高い上位5名と上位10名の正答数で評価を行う

対象プロジェクト

.....

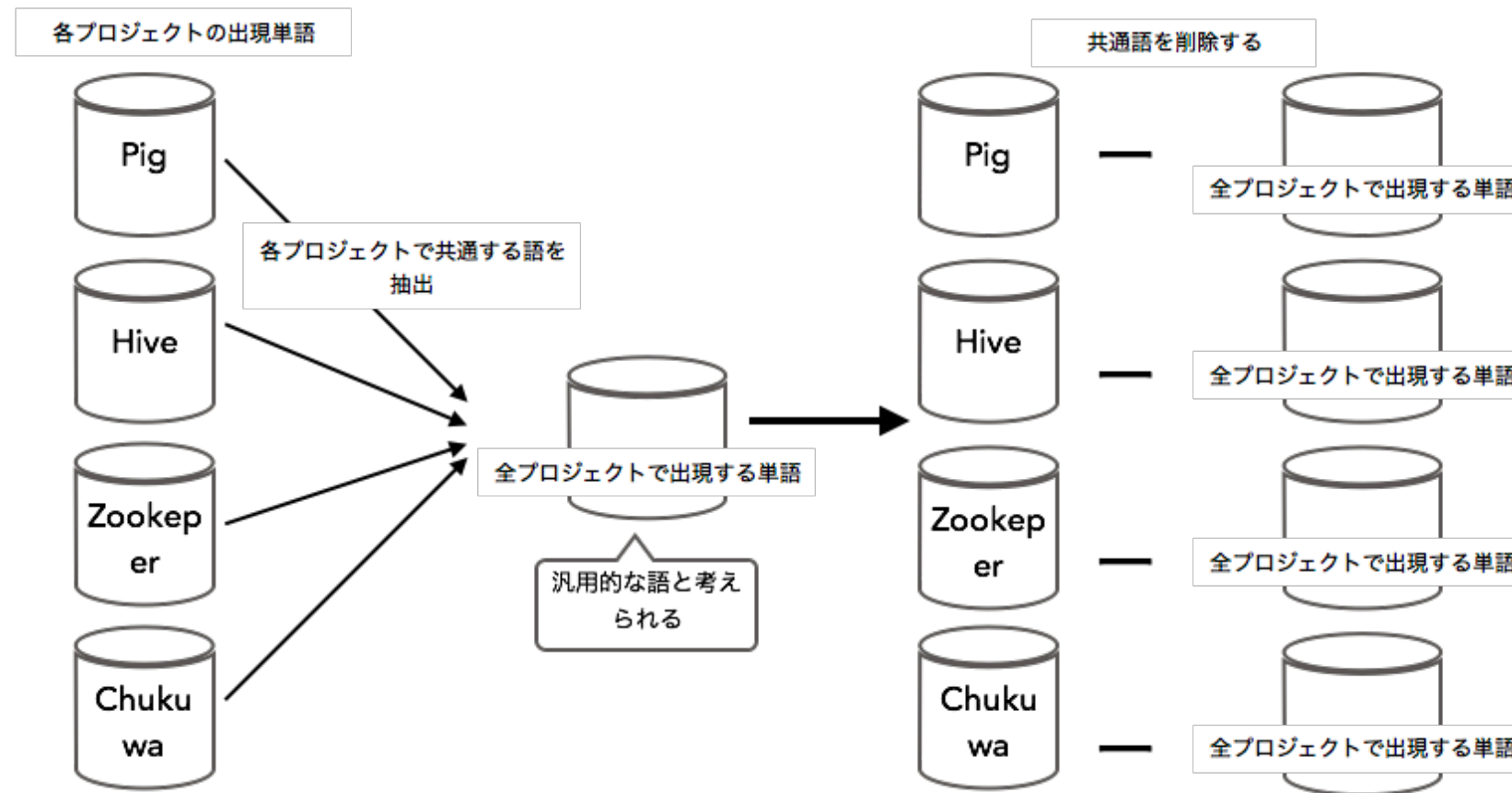
表 4.2: データセットの情報

プロジェクト	期間	件数	開発者数
Apache Hive	2008/11/11~2010/10/07	1,881	28
Apache Pig	2007/10/31~2010/09/28	1,662	44
Apache Zookeeper	2007/10/31~2010/09/28	1,506	23
Apache Chukuwa	2009/03/10~2013/10/08	877	25
合計		4,209	118

- Bugzilla JIRA Jenkinsなどの通知は削除済み

ドメイン用語の抽出手順

1. 各プロジェクトから汎用語語の削除



2. 各プロジェクトごとに単語のtf-idf値, df値を取り, それぞれ上位10単語を特徴語と頻出語として抽出

抽出したドメイン用語一覧（特徴語）

.....

	Apache Hive	Apache Pig	Apache Zookeeper	Apache Chukuwa
1	hiveopensourc	physicallay	nioservercnxn	adapter
2	srcpart	pig	zookeep	chukwa
3	metaexcept	mapreducelay	nioservercxn	hicc
4	jobconf	cocoon	todd	demux
5	thrift	gld	clientcnxn	collector
6	srcbucket	inactiveaccount	socket	seqfilewrit
7	protect	piggybank	peer	agent
8	numreducesetask	eq	antlib	row
9	processnam	acct	hunt	depart
10	libjar	foreach	elect	scienc

抽出したドメイン用語一覧（頻出語）

.....

	Apache Hive	Apache Pig	Apache Zookeeper	Apache Chukuwa
1	hiveopensource	pig	zookeep	chukuwa
2	hive	physicallay	hunt	adaptor
3	srcpart	mapreducelay	nioservercnxn	hicc
4	thrift	schema	clientcnxn	demux
5	metaexcept	foreach	todd	collector
6	jobconf	gate	elect	scienc
7	srcbucket	cocoon	socket	depart
8	protect	piggybank	nioservercxn	jiaqi
9	numreducesetask	gld	lock	agent
10	metastor	javacc	peer	seqfilewrit

結果：Apache Hiveの特徴語

	topic:100		topic:500		topic:1000	
入力語	上位 5 名の正答数	上位 1 0 名の正答数	上位 5 名の正答数	上位 1 0 名の正答数	上位 5 名の正答数	上位 1 0 名の正答数
hiveopensourc	4	8	5	7	4	6
srcpart	4	7	5	10	5	6
metaexcept	4	8	4	6	3	6
jobconf	4	8	2	3	4	8
thrift	3	8	2	5	3	5
srcbucket	5	8	2	5	5	7
protect	4	8	5	10	5	7
numreducetask	4	9	4	5	3	6
processnam	4	8	3	5	4	6
libjar	4	9	3	5	3	5
平均	4	8.1	3.5	6.1	3.9	6.2

結果：Apache Hiveの頻出語

	topic:100		topic:500		topic:1000	
入力語	上位 5 名の正答数	上位 1 0 名の正答数	上位 5 名の正答数	上位 1 0 名の正答数	上位 5 名の正答数	上位 1 0 名の正答数
hiveopensource	4	8	3	6	4	6
hive	2	7	3	7	5	7
srcpart	4	7	5	8	5	6
thrift	3	8	2	6	3	5
metaexcept	4	8	4	7	3	6
jobconf	4	8	4	6	4	8
srcbucket	5	8	5	9	5	7
protect	4	8	3	6	5	7
numreducetask	4	5	4	6	3	6
metastor	5	7	4	7	5	8
平均	3.9	7.4	3.7	6.8	4.2	6.6

結果：Apache Pigの特徴語

	topic:100		topic:500		topic:1000	
入力語	上位 5 名の正答数	上位 1 0 名の正答数	上位 5 名の正答数	上位 1 0 名の正答数	上位 5 名の正答数	上位 1 0 名の正答数
physicallay	5	10	5	10	5	10
pig	5	9	5	10	5	10
mapreduce	5	10	5	10	5	10
cocoon	5	10	5	10	5	10
gld	5	8	5	10	5	10
inactiveaccount	5	10	5	10	5	10
piggybank	4	7	5	10	4	9
eq	5	9	5	10	5	10
acct	5	10	5	10	5	10
foreach	4	8	5	10	5	10
平均	4.8	9.1	5	10	4.9	9.9

結果：Apache Pigの頻出語

	topic:100		topic:500		topic:1000	
入力語	上位 5 名の正答数	上位 1 0 名の正答数	上位 5 名の正答数	上位 1 0 名の正答数	上位 5 名の正答数	上位 1 0 名の正答数
pig	5	9	5	10	5	10
physicallay	5	10	5	10	5	10
mapreduce	5	10	5	10	5	10
schema	5	9	5	10	4	9
foreach	4	8	5	10	5	10
gate	4	6	5	9	5	10
cocoon	5	10	5	10	5	10
piggybank	5	7	5	9	4	9
gld	5	8	5	10	5	10
javacc	5	9	5	10	3	3
平均	4.8	8.6	5	9.8	4.6	9.1

結果：Apache Zookeeperの特徴語

	topic:100		topic:500		topic:1000	
入力語	上位 5 名の正答数	上位 1 0 名の正答数	上位 5 名の正答数	上位 1 0 名の正答数	上位 5 名の正答数	上位 1 0 名の正答数
nioservercnxn	3	6	5	8	5	10
zookeep	4	7	3	7	5	9
nioservercnxn	5	7	5	8	5	7
todd	5	6	5	7	4	6
clientcnxn	4	7	4	8	4	8
socket	2	5	3	6	2	5
peer	3	5	4	7	4	7
antlib	2	4	3	4	3	4
hunt	3	6	4	7	5	8
elect	5	8	5	8	5	9
平均	3.6	6.1	4.1	7	4.2	7.3

結果：Apache Zookeeperの頻出語

	topic:100		topic:500		topic:1000	
入力語	上位 5 名の正答数	上位 1 0 名の正答数	上位 5 名の正答数	上位 1 0 名の正答数	上位 5 名の正答数	上位 1 0 名の正答数
zookeep	4	7	4	7	5	9
hunt	3	6	3	6	5	8
nioservercnxn	3	6	3	6	5	8
clientcnxn	4	7	4	7	4	8
todd	5	6	5	6	4	6
elect	5	8	5	8	5	9
socket	2	5	2	5	2	5
nioservercxn	5	7	5	7	5	7
lock	5	9	5	9	3	6
peer	3	5	5	6	4	7
平均	3.9	6.6	4.1	6.7	4.2	7.3

結果：Apache chukuwaの特徴語

	topic:100		topic:500		topic:1000	
入力語	上位 5 名の正答数	上位 1 0 名の正答数	上位 5 名の正答数	上位 1 0 名の正答数	上位 5 名の正答数	上位 1 0 名の正答数
adapter	5	9	5	8	5	9
chukwa	5	8	5	7	5	10
hicc	5	7	3	5	5	9
demux	5	10	5	7	5	8
collector	4	6	4	7	5	7
seqfilewrit	4	7	5	7	4	5
agent	4	6	5	8	5	8
row	4	7	2	4	2	3
depart	3	6	5	8	5	10
scienc	3	5	5	8	5	9
平均	4.2	7.1	4.4	6.9	4.6	7.8

結果：Apache chukuwaの頻出語

	topic:100		topic:500		topic:1000	
入力語	上位 5 名の正答数	上位 1 0 名の正答数	上位 5 名の正答数	上位 1 0 名の正答数	上位 5 名の正答数	上位 1 0 名の正答数
chukuwa	5	8	5	8	5	10
adaptor	5	9	5	7	5	9
hicc	5	7	5	8	5	9
demux	5	10	5	8	5	8
collector	4	6	5	7	5	7
scienc	3	5	5	8	5	9
depart	3	6	5	7	5	10
jiaqi	4	7	3	6	5	7
agent	4	6	5	7	5	8
seqfilewrit	4	7	4	5	4	5
平均	4.2	7.1	4.7	7.1	4.9	8.2

結果：まとめ

- 正答数はプロジェクトとトピック数によるが、 6割～10割
- Apache Pigの特徴語を用いたトピック数1000に設定した際の正答数が10割で最も高い
- Apache Hiveの特徴語を用いたトピック数500に設定した際の正答数が上位5名が約7割、上位10名が約6割が最も低い

考察

- プロジェクトごとの正答数の考察
 - ▶ Apache Pigの正答数が高いのは、他のプロジェクトと比較して開発者が2倍いたためだと考えられる
- トピック数の考察
 - ▶ トピック数による正答数の変化は、多少はあるがそこまで大きくない
 - ▶ 手法の入力語の生起確率の高いトピックを特定している部分で、不要なトピックは削っているためと考えられる
- 検索語の考察
 - ▶ 検索語ごとに使用した開発者数を出してみたら何か分かるかもしれない