

Mining Analogical Libraries in Q&A Discussions

Incorporating Relational and Categorical Knowledge into Word Embedding

**Chunyang Chen,
Sa Gao,
Zhenchang Xing**

背景

- 多くのソフトウェアプロジェクトが外部ライブラリに依存している
 - thung[1]らの調査ではGithubの1,008プロジェクト中93.3%が依存しており，平均で28のライブラリを使っている
- 1つの外部ライブラリをずっと使えるとは限らない
 - 外部ライブラリがずっと開発され続けるとは限らない
 - より多機能なライブラリが必要になったり，仕様の都合で多言語に対応させる必要がある場合も考えられる

外部ライブラリを探している例

- StackOverflowの質問
 - C++でNUnitに似たライブラリを探している
- QAサイトでは， 有益な解答を得るまで時間がかかる

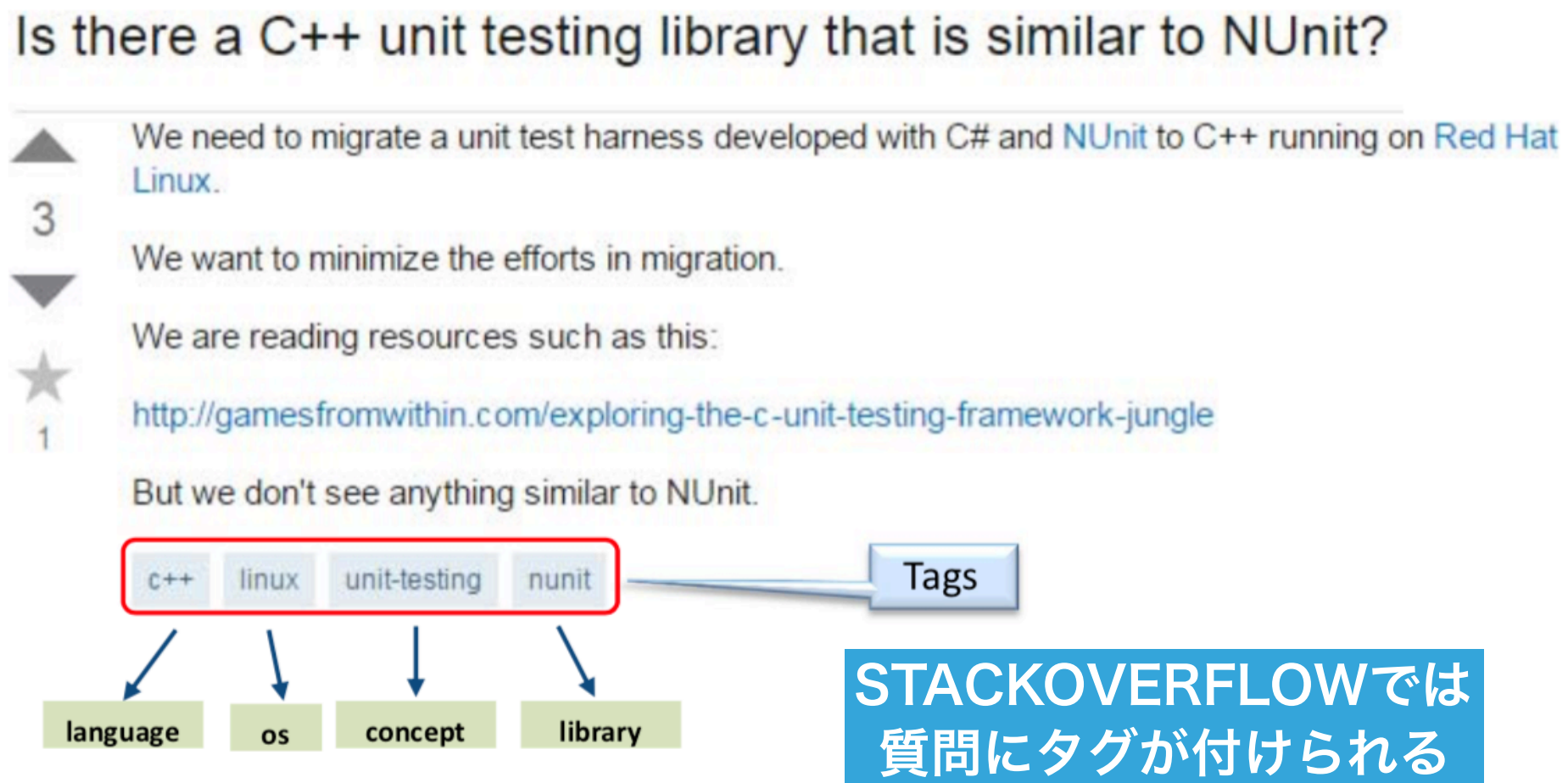


Fig. 1. An example of analogical-library question on Stack Overflow.

目的とアプローチ

- 目的

- 類似ライブラリを見つけ出す新しいアプローチの提案

- アプローチ

- 質問に付けられているtagに着目
 1. 相関ルールマイニングにより、重要な関係のタグセットを抽出
 2. Tag wikiからタグを取得し、タグの表記揺れを統一し、カテゴリー分け
 3. Word2Vecにタグを学習

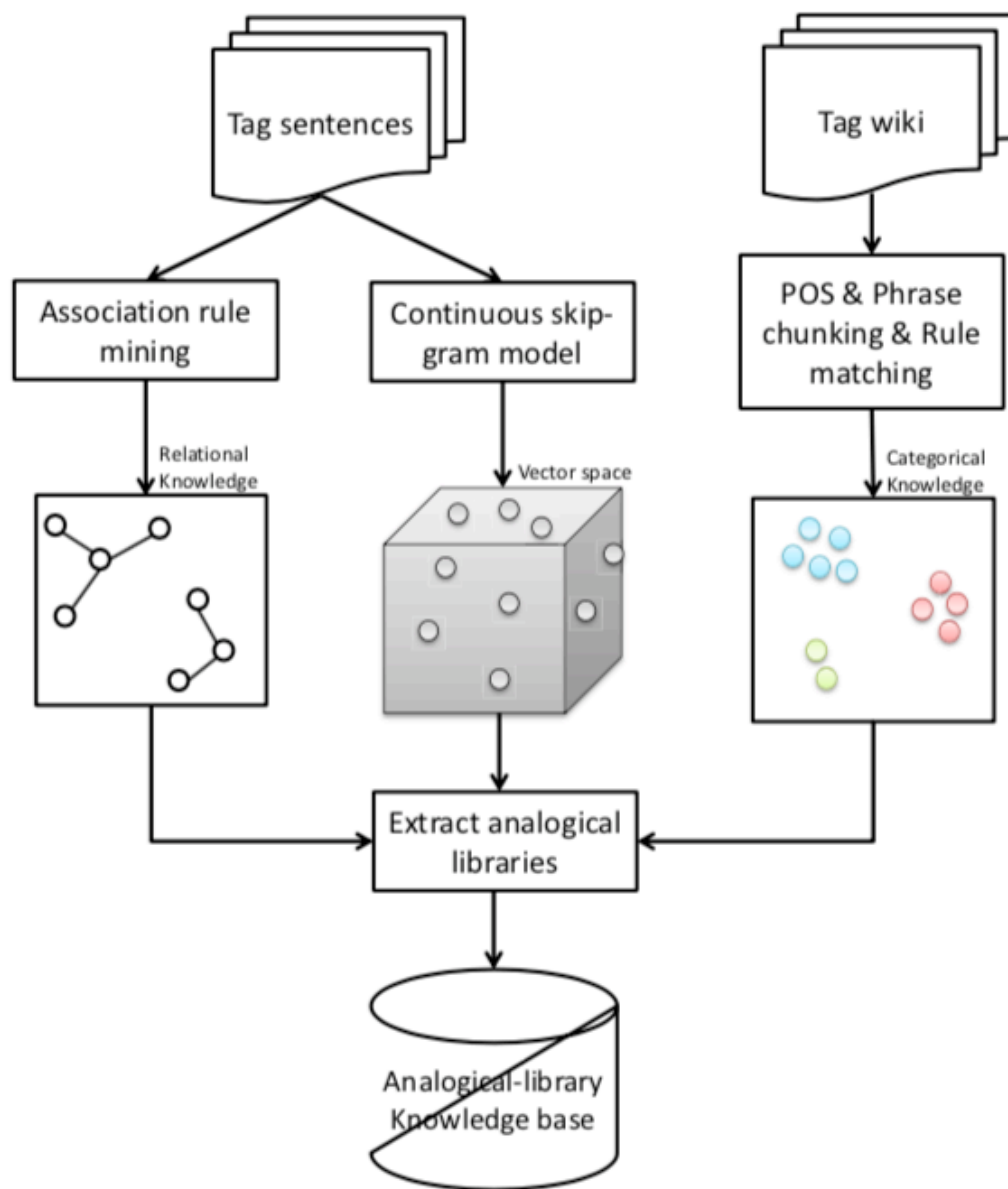


Fig. 3. The overview of our approach

1. 相関ルールマイニング

- 相関ルールマイニング
 - 頻繁に同時に起こる相関を見つけ出す
 - ▶ 同時に使われるタグを見つけ出す
 - 重要なタグの組み合わせを抽出
- タグの相関グラフ
 - 各タグの中心的なタグが把握できる
 - ▶ ライブラリータグのプログラミング言語やプラットフォームが把握できる

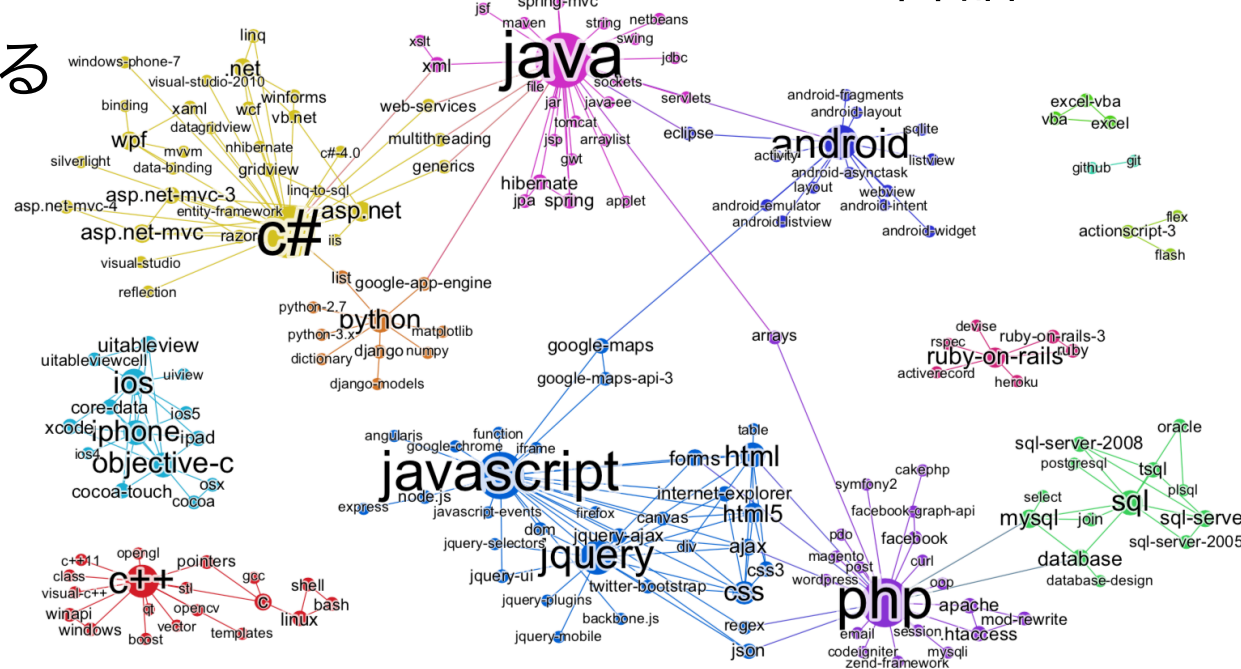


Fig. 4. An example of tag correlation graph

2.TagWikiからカテゴリー分け

- Tagwiki

- StackOverflowのタグの説明が書かれている

- カテゴリー分け

- 形態素解析によるカテゴリーの抽出
 - ▶ iosはoperating systemカテゴリーに属している

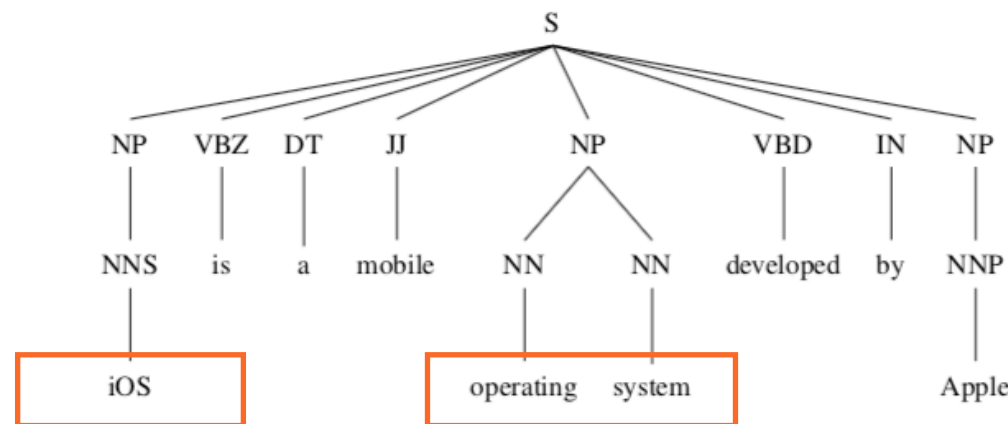


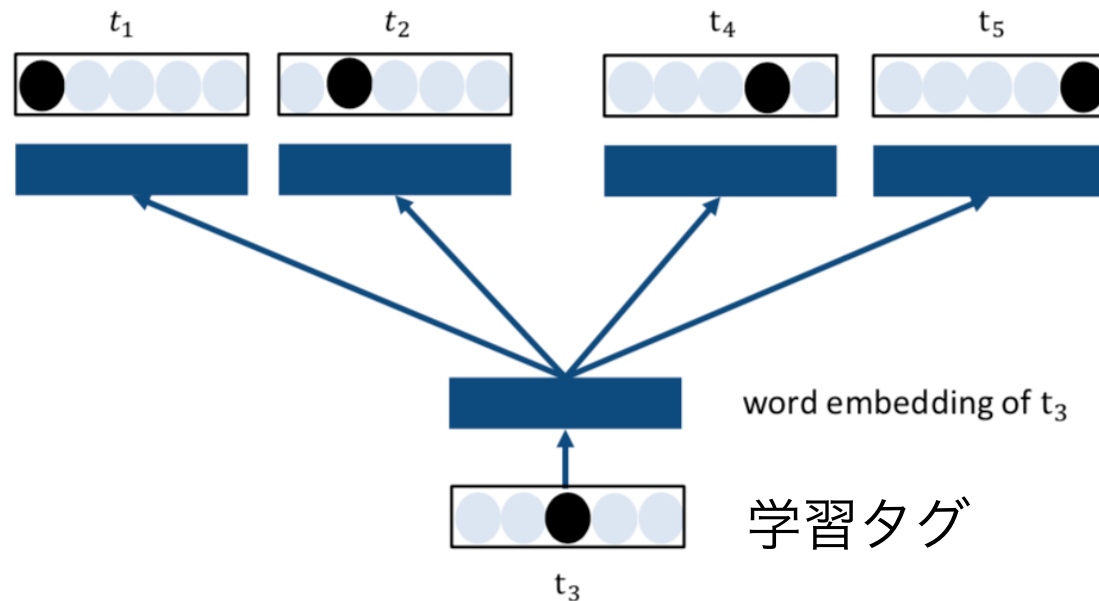
Fig. 5. POS tagging and phrase chunking results of the definition sentence of the tag *ios*

- 表記ゆれの統一 (operating system → os)
- 各カテゴリーへ手動でラベル付
 - ▶ 例：java python (言語カテゴリー)

3. Word Embedding

学習

学習タグと一緒に使われているタグ



skip-gram

単語をベクトル化

- 単語のベクトル演算が可能

▶ $\text{man} - \text{woman} + \text{queen} = \text{king}$

▶ $\text{paris} - \text{france} + \text{spain} =$

- ライブラリでもできるのでは？

▶ $\text{nltk} - \text{python} + \text{java} = \text{opennlp}$

類似ライブラリの取得

$$\operatorname{argmax}_{t_2 \in T} \cos(\operatorname{vec}(t_2), \operatorname{vec}(t_1) - \operatorname{vec}(PL_1) + \operatorname{vec}(PL_2)) \quad (2)$$

出力ライブラリ

入力ライブラリTAG

入力ライブラリの言語TAG

出力ライブラリの言語TAG

言語カテゴリーから選択

言語カテゴリーから選択

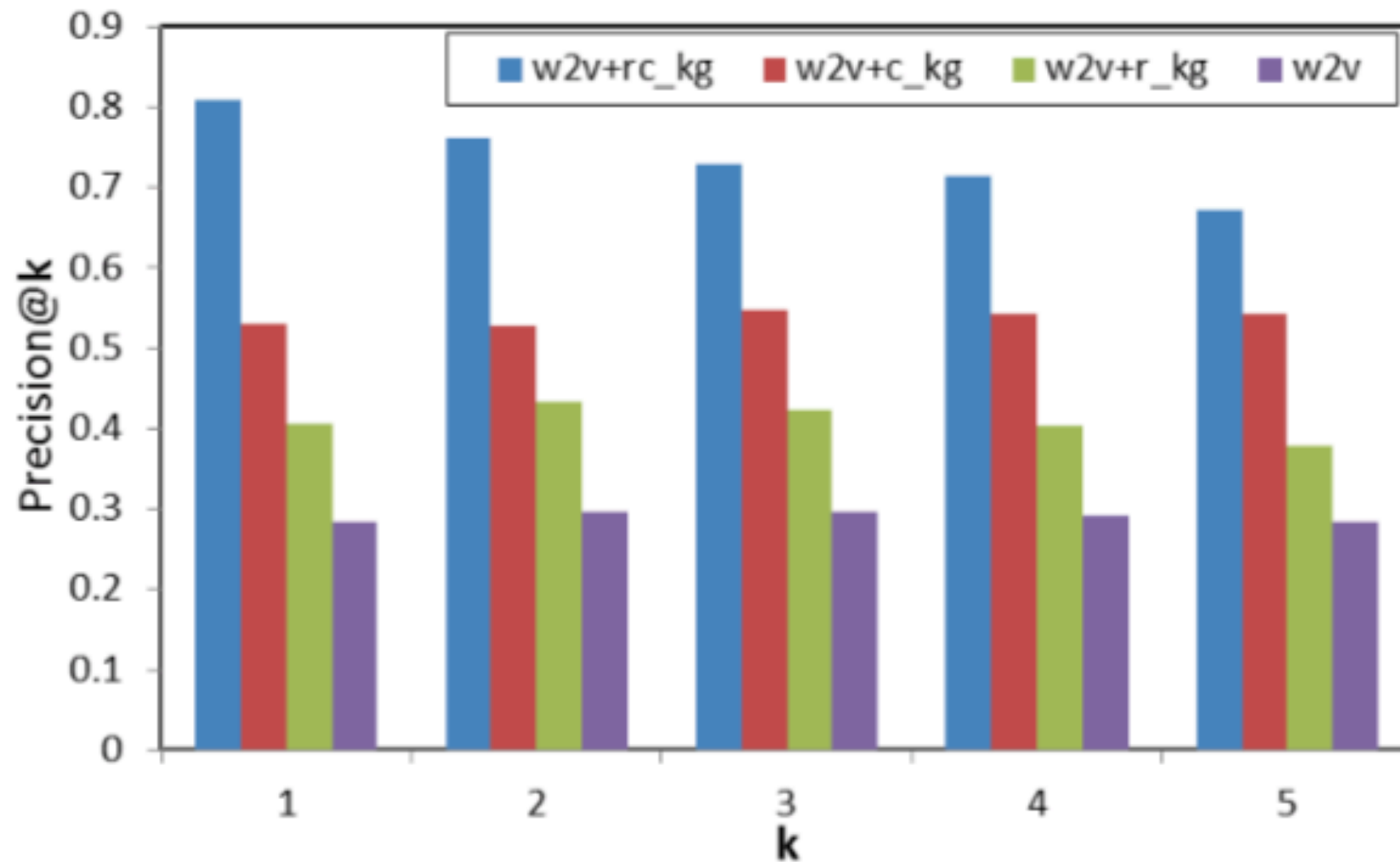
評価実験

- 実験手順
 - 100のライブラリをランダムで選択
 - 手法に適用し，適用結果を目視により評価
- 評価指標
 - Precision@K
 - 推薦された要素K（上位1~K）の正確さ

$$\sum_{i=1}^n \frac{Precision_i@k}{n}$$

結果

- word2vecのみ、各前処理を入れた場合の精度評価



W2v : word2vec

r_kg : 相関ルールマイニング

c_kg : カテゴリ分け

まとめと所感

- **まとめ**

- Q&Aサイトの質問に付けられるtagを利用した, 類似ライブラリの推薦手法の提案
- $\text{Precision@5} = 0.67$, $\text{Precision@1} = 0.81$

- **所感**

- word2vecの単語演算を利用した手法を思いついたのがすごい. (しかもタグを学習させて)
- 前処理の重要性を感じた