

SES2018参加報告

概要

- 場所：東洋大学白山キャンパス
- 日程：9月5日～9月7日（5日：ワークショップ）
 - 台風のためワークショップには参加できず...
- 参加者：計156名
- 発表者の割合
 - 学生60%，企業の方，教員の方20%ずつ
- 今年からジャーナルの発表も追加

ソースコード解析 座長：今井 健男（LeapMind）	ソフトウェア品質・再利用 座長：遠藤 佑介（クックパッド）	招待論文(1) アプリケーション開発 座長：角田 雅照（近畿大学）
字句列の共通性に基づく例外処理条件文の抽出手法の提案 吉田 敦（南山大学），加藤 大貴（バッファロー），蜂巢 吉成，桑原 寛明（南山大学），阿草 清滋（京都高度技術研究所）（研究論文/ロング発表）	SMTソルバーを使ったカスタマイズ可能なブロックチェーン合意ルールの検証 河原 亮（IBM Research）（研究論文/ロング発表）	物語記述によるホームネットワークアプリケーション開発手法の提案 大村 廉（豊橋技術科学大学）（招待論文/ロング発表）
順伝播型ニューラルネットワークを用いた類似コードブロック検索の試み 藤原 裕士（大阪大学），崔 恩潯（奈良先端科学技術大学院大学），吉田 則裕（名古屋大学），井上	Web API品質モデルの提案とその定量評価 山本 里枝子，大橋 恭子，福寄 雅洋，木村 功作，関口 敦二，梅川 竜一，上原 忠弘（富士通研究所），青山 幹雄（南山大学）（実践論文/ロング発表）	ソーシャルコーディングにおける有益提案の抽出 江見 圭祐（岡山大学 / 現在 サイボウズ株式会社），乃村 能成，谷口 秀夫（岡山大学）（招待論文/ロング発表）

発表

- ポスター（90分）

- 非機能要件の詳細な自動分類手法の構築に向けて：セキュリティ要件の分類
- Politeness分析に基づくOSS開発者の活動継続性の理解
- ソフトウェアテストにおけるSilent Horrorsの分析へ向けて

- 論文発表（発表20分質疑10分）

- OSSコミュニティにおける開発者の活動継続性を理解するためのPoliteness分析

フィードバック

- 非機能要件の詳細な自動分類手法の構築に向けて：セキュリティ要件の分類
 - 企業の方の声
 - 要件定義書に書いた書いてないで対立することがよくある
 - 人によって書き方が違うのをなんとかしようとしている
 - 英語を対応としているが，日本語にも使えるの？

フィードバック

- ソフトウェアテストにおけるSilent Horrorsの分析へ向けて
 - この研究は大変. . . （ほぼほぼ全員の方から）
 - 非常に多くの方から興味は持たれていた
 - 今回対象としてるREOPENED欠陥は先行研究のSilent Horrorsの原因に含まれてる？

フィードバック

- Politeness

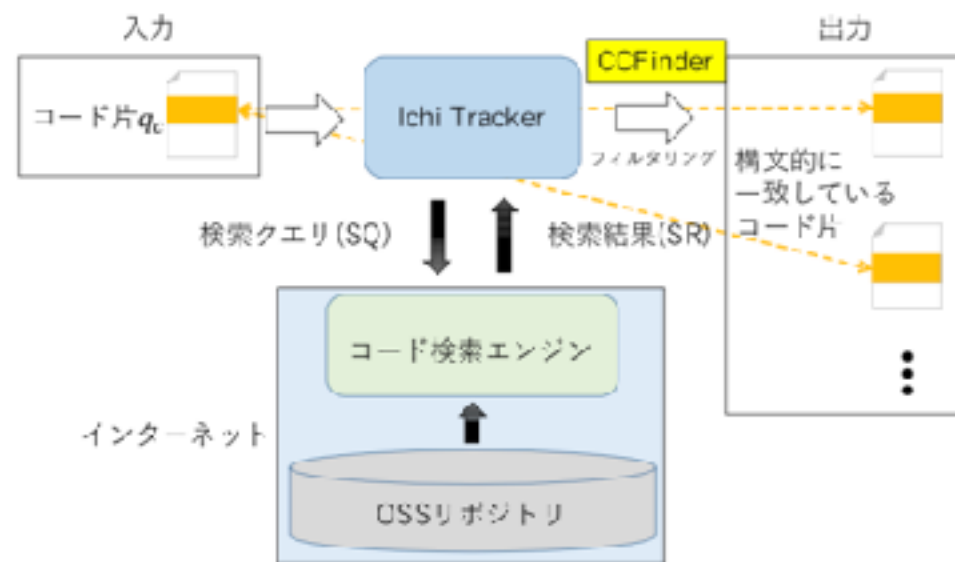
- バグ報告とかレビューのコメントを対象にしてみては？
 - 話が長くなりすぎて放置状態になってたりする
- コミュニケーションが影響で離脱しているのはどれくらいの割合？
- 開発の効率性とPolitenessはトレードオフじゃない？
- ネイティブでない人だと「Please～」や「So～」といったPolitenessを下げる表現をしがち（Rubyのコミッターの方から）

論文紹介

順伝播型ニューラルネットワークを用いた類似コードブロック検索の試み

藤原 裕士* 崔 恩瀨** 吉田 則裕*** 井上 克郎*

*大阪大学 **奈良先端科学技術大学院大学 ***名古屋大学



$$Similarity = \frac{2 * |t1 \cap t2|}{|t1| + |t2|}$$

2つのコード片中に重複している行があれば類似コード片として定義

図 3 Ichi Tracker の概要

- コードクローン検出ツールを用いた手法ではそれぞれの類似の定義によって特定のコード片を特定できない点を解消するのが目的
- 特徴ベクトルにはBOWとDoc2Vecを利用
 - BOWの方が精度はわずかに良い

順伝播型ニューラルネットワークを用いた類似コードブロック検索の試み

.....

• 出力例

```
{
  int num,i;
  char *p;
  (中略)
  for (;;)
  {
    (中略)
    if (num >= arg->count)
    {
      char **tmp_p;
      int tlen = arg->count + 20;
      tmp_p = (char **)OPENSSL_realloc(arg->data,
        sizeof(char *)*tlen);
      if (tmp_p == NULL)
        return 0;
      arg->data = tmp_p;
      arg->count = tlen;
      for (i = num; i < arg->count; i++)
        arg->data[i] = NULL;
    }
    arg->data[num++] = p;
    (中略)
  }
  *argc=num;
  *argv=arg->data;
  return(1);
}
```

(a) 入力コード片

```
{
  int num,len,i;
  char *p;
  (中略)
  for (;;)
  {
    (中略)
    if (num >= arg->count)
    {
      arg->count+=20;
      arg->data=(char **)OPENSSL_realloc(arg->data,
        sizeof(char *)*arg->count);
      if (argc == 0) return(0);
    }
    arg->data[num++] = p;
    (中略)
  }
  *argc=num;
  *argv=arg->data;
  return(1);
}
```

(b) 検索結果

ソフトウェア変更時のコーディング規約違反と不具合の共起傾向の調査

田口 健介* 名倉 正剛** 高田 眞吾*

*慶應義塾大学 **南山大学

- これまで使用されてきたメトリクスはプロジェクトに依存するため、プロジェクトに依存しないようコーディング規約違反をメトリクスに利用
 - 分類器にはDeep Embedded Clusteringを利用

表 4 コーディング規約違反の有意差による順位付け

コーディング規約名	ndk	hadoop	presto	netty	Google	Sun
FinalParameters	9	1	1	2	×	○
EmptyLineSeparator	4	4	2	8	○	×
JavadocType	3	6	6	3	×	○
LineLength	10	2	9	4	○	○
MissingCtor	1	10	3	7	×	×
JavadocMethod	7	3	8	5	○	○
OuterTypeFilename	2	8	4	9	○	×
WriteTag	5	5	7	6	×	×
AnnotationOnSameLine	8	9	10	1	×	×
ImportControl	6	7	5	10	×	×
MissingSwitchDefault	11	11	11	11	○	○

表 5 コーディング規約ごとの不具合予測正解率

コーディング規約名	camel	fabric8	log4j	tomcat
FinalParameters	72.5%	70.1%	55.4%	60.7%
LineLength	66.1%	68.3%	50.4%	53.3%
MissingSwitchDefault	70.8%	69.1%	52.2%	56.2%
すべて利用した場合	77.8%	76.3%	60.5%	68.7%
(適合率)	60.1%	53.8%	64.1%	40.3%
(再現率)	51.2%	49.0%	30.8%	28.4%
(F 値)	0.553	0.513	0.605	0.333

不具合有無とコーディング規約の関係

コード片のベクトル表現に基づく大規模コードクローン集合の特徴調査

横井 一輝* 崔 恩瀨** 吉田 則裕*** 井上 克郎*

*大阪大学 **奈良先端科学技術大学院大学 ***名古屋大学

- 情報検索技術を用いることで構文が類似していないクローンでも検出可能だが、どのベクトル表現がクローン検出に適しているかわからないため調査

表 1 ベクトル表現ごとの再現率

	BoW	TF-IDF	LSI	LDA	Doc2Vec	WV-avg	FT-avg
T1	0.99	0.99	0.99	0.99	0.99	0.99	0.99
T2	0.84	0.82	0.92	0.85	0.91	0.95	0.94
VST3	0.90	0.82	0.91	0.95	0.83	0.97	0.93
ST3	0.45	0.37	0.61	0.61	0.46	0.84	0.79
MT3	0.06	0.03	0.09	0.23	0.04	0.55	0.43
WT3/T4	0.00	0.00	0.00	0.02	0.00	0.08	0.05

- 分散表現再現率向上，ただし適合率はおそらく減
- WMD(Word Mover's Distance)を使うとFalse Positive を下げれる
- BOWで分類した後に， Word2Vec + WMDで分類する手法を最後に提案

Eclipse寄付バッジによるバグレポート応答時間の影響分析

.....

中才 恵太郎* 畑 秀明* Raula Gaikovina Kula* 松本 健一*

*奈良先端科学技術大学院大学

- 寄付バッチ：35ドル以上寄付した場合1年間つけれるバッチ, プロジェクトへの貢献の証
- 寄付バッジ有り開発者と寄付バッジ無し開発者で報告したバグレポートの応答時間を比較
 - 差分の差分法を用いて分析を実施
 - 寄付バッジ有りの方が約2時間応答時間が短縮
 - アンケートをとった結果, ほとんどの開発者は寄付バッチの有無は関係ないと返答
- IEEE Softwareに掲載予定

所感・その他

- ポスター発表3つはやっぱりきびしい（体力面&得られるフィードバック面）
 - 一つ説明中に他のポスターを見てくれてる集団ができて、説明が終わった頃にはその集団も移動...
 - 質問・フィードバックを受けきれないので残念
- 情報交換会には積極的に参加を
 - 他大学の学生と喋ったり、企業の方の困っていること、発表中では聞けなかったフィードバックが聞けたりします
- 伊原研の学生に会いました
 - コミッターに関する研究をする予定の学生がいた
 - やる気がすごかったので頑張りましょう