

Evaluating the Use of Semantics for Identifying Task-relevant Textual Information

Arthur Marques

Department of Computer Science
University of British Columbia, Canada
msarthur@cs.ubc.ca

Gail C. Murphy

Department of Computer Science
University of British Columbia, Canada
murphy@cs.ubc.ca

背景

▶ バグ修正や機能追加などのタスクでは

APIドキュメント

バグレポート

コミュニティフォーラム

などの多くの**artifact**を参照する

▶ 開発者はこれらの**artifact**から自分のタスクに関連するごく一部のテキストを特定する必要がある

→ 不正解・不完全な解を導いてしまう可能性

先行研究

▶ AnswerBot[4]

テキストの内容

タスクの内容

Stack Overflowの投稿にある回答の各段落で利用できる構造化データ



MMRアルゴリズム

関連するテキストを識別

クエリに関連する答えを推薦

[4] B. Xu, Z. Xing, X. Xia, and D. Lo, "AnswerBot: Automated generation of answer summary to developers' technical questions," in *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2017, pp. 706–716.

目的

- ▶ 複数の種類のソフトウェア **artifact** と自分のタスクに関連するテキストを特定する技術を開発
- ▶ **artifact** の種類によらず、1つの技術に依存できるようにする
 - 特定の種類の **artifact** に特化した手法と同様の性能が必要

提案手法

▶ タスクに関連するartifactの文を特定

- ・ 入力：タスクの記述とartifactのテキスト
- ・ 出力：タスクの解決に最も役立ちそうな文の順位

▶ 評価方法

TABLE I: Summary of semantic-based approaches for automatically identifying task-relevant text

Identifier	Sentence filters	Description
word2vec	<i>no filter</i>	Uses the Skip-gram model to identify relevant sentences as the sentences most semantically similar to a task
	<i>w/ frame-elements</i>	Modifies the output of the word2vec base approach according to whether sentences contain meaningful frame elements
	<i>w/ frame-associations</i>	Modifies the output of the word2vec base approach according to whether sentences contain meaningful task-artifact frame pairs
BERT	<i>no filter</i>	Fine-tunes BERT to predict the sentences that are likely relevant to an input task
	<i>w/ frame-elements</i>	Modifies the output of the BERT base approach according to whether sentences contain meaningful frame elements
	<i>w/ frame-associations</i>	Modifies the output of the BERT base approach according to whether the sentences contain meaningful task-artifact frame pairs

▶ Word2vecとBERTでタスクに関連するartifactのテキストを順位付け

▶ フレームセマンティクスを用いてタスク関連テキストのフィルタリング

提案手法

▶ フレームセマンティクス

- ▶ ある事象に対して、文中の単語の役割をどのように理解するかに基づいた推論

▶ フィルタリング

- ▶ Wrod2VecとBERTの後処理段階としてフィルタをセット
- ▶ 文がフィルタの関連性基準に一致するかを判定
- ▶ ソフトウェア工学のテキストに関連するフレームを割り当てるためにSEFrame[35]を使用

[35] A. Marques, G. Viviani, and G. C. Murphy, “Assessing semantic frames to support program comprehension activities,” in *2021 IEEE/ACM 29th International Conference on Program Comprehension (ICPC)*, 2021, pp. 13–24.

提案手法

フレーム要素

フレームで識別した文が、 関心のある情報を含む可能性がある と判定するフィルタ

$$F = \{\text{being obligated, causation, } \dots \text{required event, using}\} \quad (2)$$

$$\text{frame-elements}(a_i) = \begin{cases} 1, & \text{if } \text{frames}(a_i) \cap F \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

- 文 $a_i \in A$ を取り,
- フレーム要素のいずれかが
フレーム $f \in F$ に一致するか判定

フレームアソシエーション

文の関連性がタスクの意図に依存することを考慮したフィルタ

入力で得られたフレームのペアが、 予め設定されたペアの集合に含まれるか判定

相関ルールマイニングを使用してペア $p \in P$ を識別

$$P = \{(\text{execution, being obligated}), \dots (\text{questioning, using})\} \quad (4)$$

$$\text{frame-associations}(a_i, T) = \begin{cases} 1, & \text{if } \text{pair}(a_i, T) \cap P \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

データセット



Fig. 2: Summary of procedures for dataset creation

- ① GitHubとStack Overflowから50のAndroid開発タスクをランダムに抽出
- ② Google検索エンジンでそのタスクに関連する情報を含む可能性のあるartifactを見つける
- ③ ②で見つけたartifactからタスクに関連する文を経験豊富な開発者から収集

TABLE II: Android dataset summary

	# of sentences			# of sentences marked		
	total	mean	stdv	total	mean	stdv
API documentation	4,915	109.22	96.77	327	9.62	7.88
GitHub issues	1,890	43.95	33.69	146	4.87	3.37
Stack Overflow answers	1,420	28.40	28.48	330	7.33	5.08
Miscellaneous pages	4,176	70.78	56.64	590	12.55	9.88
Overall	12,401	62.95	66.25	1393	8.93	7.78

評価

▶ AnswerBotと比較

- ・タスクとStack Overflowのartifactのみで比較
- ・プラットフォーム上で得た回答の投票数や回答が受け入れられたかも評価

▶ 結果

TABLE III: Comparison to AnswerBot

technique	<i>no filter</i>		<i>w/ frame-elements</i>		<i>w/ frame-associations</i>	
	precision	recall	precision	recall	precision	recall
AnsBot	0.59	0.63	-	-	-	-
word2vec	0.43*	0.48*	0.49*	0.45*	0.45*	0.50*
BERT	0.58	0.63	0.53*	0.58*	0.58	0.62

* AnsBot performs better with a large effect size ($\alpha = 0.1$, Cohen's $D \geq 0.8$);

- ・ AnswerBotと提案手法のprecisionとrecallで評価
- ・ BERTとフレームアソシエーションの組み合わせがAnswerBotと同等の性能

評価

▶ 全artifactタイプでの評価

- ▶ タスクに関連するartifact内のテキストをどれだけ識別できるかを測定
- ▶ テストデータ中の全てのartifactについてprecisionとrecallを計算

▶ 結果

- ▶ Word2vecではprecisionとrecallどちらもあまり高くない
- ▶ BERTは多くのartifactに対して良好な結果

TABLE V: Evaluation Metrics Over All Artifacts and Artifact-type Wise

	artifact	<i>no filter</i>		<i>w/ frame-elements</i>		<i>w/ frame-associations</i>	
		precision	recall	precision	recall	precision	recall
word2vec	API documentation	0.47	0.39	0.47	0.40	0.53	0.52
	GitHub issues	0.40	0.36	0.60	0.54	0.44	0.36
	Stack Overflow answers	0.43	0.48	0.49	0.45	0.45	0.50
	Miscellaneous pages	0.49	0.49	0.50	0.45	0.50	0.47
	overall	0.45	0.43	0.51	0.46	0.48	0.46
	standard deviation	0.03	0.05	0.05	0.05	0.03	0.06
BERT	API documentation	0.52	0.55	0.51	0.57	0.52	0.55
	Github issues	0.52	0.56	0.53	0.55	0.52	0.56
	Stack Overflow answers	0.58	0.63	0.53	0.58	0.58	0.62
	Miscellaneous pages	0.52	0.56	0.51	0.54	0.53	0.57
	overall	0.54	0.58	0.52	0.56	0.54	0.58
	standard deviation	0.02	0.03	0.01	0.01	0.02	0.02

まとめ

▶ 目的

- ・ 複数の種類のソフトウェア**artifact**に適用できる技術を開発
- ・ **artifact**の種類ごとに 1 つの技術に依存できるようにする

▶ 結果

- ・ BERTでは先行研究と同等, もしくはそれ以上の結果が得られた
- ・ 一方, Word2Vecではprecisionとrecallどちらも向上していない
- ・ Artifactによるprecisionとrecallの差はあまりない

所感

- ▶ 正確な情報が得られるのはいいが、結局自分で検索しないといけないのはめんどくさい. .
- ▶ Precisionが0.6くらいでは実用的ではない？