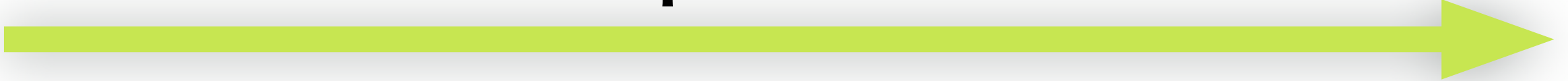


Choosing an NLP Library for Analyzing Software Documentation: A Systematic Literature Review and a Series of Experiments



Fouad Nasser A Al Omran

Christoph Treude

背景

- ◎ 自然言語で書かれた文章は大量に存在する
 - ▶ コミットメッセージ・メール・ドキュメント
- ◎ 解析するためにはNLP(自然言語処理) が用いられる
- ◎ 自然言語を扱うツール開発者、研究者のほとんどはNLPライブラリに依存している
- ◎ NLPライブラリは複数ある
 - ▶ Stanford core NLP・NLTK・spaCy

自然言語処理とは

従来の言語解析アルゴリズム

単語分解	形態素解析	大量の辞書と文法の知識を使って品詞に分解	辞書・文法的アプローチ
	N-gram解析	機械的に文をN個に分解	数学的アプローチ
構文解析	形態素解析	品詞情報から係り受け判別を行い文構造を解析	辞書・文法的アプローチ
意味解析	シソーラス解析	大量の類義語・反義語辞書（シソーラス）を使って出現語をグルーピング化	辞書・文法的アプローチ
	ベクトル空間解析	統計解析的手法を使って語の近傍度を数学的に計算	数学的アプローチ

和歌山大学システム工学部の出身です。

例：単語分割,形態素解析

.....

和歌山大学システム工学部の出身です。

NLTK
ライブラリ

和歌山大学 名詞,固有名詞,組織,*,*,*,和歌山大学,ワカヤマダイガク,ワカヤマダイガク
システム 名詞,一般,*,*,*,システム,システム,システム
工 名詞,接尾,一般,*,*,*,工,コウ,コー
の 助詞,連体化,*,*,*,の,ノ,ノ
出身 名詞,一般,*,*,*,出身,シュッシン,シュッシン
です 助動詞,*,*,*,特殊・デス,基本形,です,デス,デス

問題点

- ◎ 自然言語で書かれた文章は物によって異なる
 - ▶ 文章によっては文法が守られていない場合もある
- ◎ ライブラリによって、単語分割される部分・付けられる品詞タグが異なる
- ◎ しかし研究者はライブラリの選択には重きを置いていない？

問題点

TABLE I: Use of NLP libraries in recent software engineering conference papers

Venue	Papers	Has Keyword	Stanford	SyntaxNet	NLTK	spaCy	LangPipe	OpenNLP	FudanNLP & IctcasNLP	in-house	others
ICSE	466	68	7	0	0	0	0	0	0	0	3
ASE	250	40	4	0	0	0	0	0	0	0	2
FSE	299	48	3	0	1	0	0	1	0	1	2
MSR	148	41	1	0	0	0	1	0	1	0	2
ICSME	187	35	3	0	0	0	0	0	0	0	1
Sum	1,350	232	18	0	1	0	1	1	1	1	10

- 2010~2015年に行われた国際会議の論文 1,350本
- NLP(自然言語処理) ライブラリを使用しNLPについて言及している論文は 232本
- 使用したライブラリについて言及されている論文 33本

研究目的

- 特定のライブラリを選択することの影響を調査する
 - ▶ どういった文章に各ライブラリが適しているのか
- 研究者や開発者のライブラリの選択を助ける

比較ライブラリ

● Google's SyntaxNet

- ▶ TensorFlowに実装されているNLPライブラリ

● Stanford core NLP Suite

- ▶ ソフトウェア工学研究で最も一般的に使用されているライブラリ
- ▶ 多数の言語をサポート

● NLTK Python Library

- ▶ Pythonライブラリ
- ▶ 多くの言語をサポート

● spaCy

- ▶ PythonとCythonで書かれたNLPライブラリ
- ▶ 他のライブラリに比べて処理が高速

分析する文章

Javaについて書かれた文章

◎ Stack Over Flow

- ▶ Javaのタグがついた質問と回答文章
- ▶ 投稿者の数が多いため、多様な文体

◎ GithubのReadMeファイル

- ▶ Java言語が用いられているプロジェクト
- ▶ StackOverFlowほどではないが多様な文体

◎ JavaAPIドキュメント

- ▶ 比較的文法が整っている

アプローチ1

1. StackOverFlow , Github ReadMe , Java APIに対して各ライブラリを適用
2. 単語分割 , 形態素解析を行う
3. 文章ごとに各ライブラリの結果を比較

結果

TABLE II: Degree of overlap between the results from all libraries

Comparison		tokens for first library	tokens for second library	identical tokens	identical tokens %	identical token/POS _g	identical token/POS _g %	identical token/POS _s	identical token/POS _s %
Stack Overflow	Stanford vs. SyntaxNet	8,743	8,485	8,137	94.46	7,507	87.15	7,322	85.00
	Stanford vs. spaCy	8,743	8,589	8,312	95.92	7,743	89.35	7,548	87.10
	Stanford vs. NLTK	8,743	8,614	8,454	97.41	7,062	81.37	6,605	76.11
	SyntaxNet vs. spaCy	8,485	8,589	8,016	93.90	7,678	89.94	7,511	87.98
	SyntaxNet vs. NLTK	8,485	8,614	8,281	96.86	6,876	80.43	6,441	75.34
	spaCy vs. NLTK	8,589	8,614	8,326	96.80	7,035	81.79	6,585	76.56
GitHub ReadMe	Stanford vs. SyntaxNet	6,289	6,167	5,996	96.27	5,271	84.63	4,998	80.25
	Stanford vs. spaCy	6,289	6,278	6,034	96.03	5,332	84.86	5,100	81.16
	Stanford vs. NLTK	6,289	6,232	6,133	97.96	5,017	80.14	4,395	70.20
	SyntaxNet vs. spaCy	6,167	6,278	5,910	94.98	5,512	88.58	5,307	85.29
	SyntaxNet vs. NLTK	6,167	6,232	6,093	98.28	4,901	79.05	4,399	70.96
	spaCy vs. NLTK	6,278	6,232	6,040	96.56	4,885	78.10	4,313	68.95
Java API Doc.	Stanford vs. SyntaxNet	11,675	10,896	10,488	92.93	9,103	80.66	8,564	75.88
	Stanford vs. spaCy	11,675	11,015	10,611	93.53	9,390	82.77	8,832	77.85
	Stanford vs. NLTK	11,675	11,044	10,788	94.97	8,711	76.68	7,889	69.45
	SyntaxNet vs. spaCy	10,896	11,015	10,008	91.35	9,171	83.71	8,697	79.38
	SyntaxNet vs. NLTK	10,896	11,044	10,735	97.86	8,734	79.62	7,950	72.47
	spaCy vs. NLTK	11,015	11,044	10,311	93.49	8,602	77.99	7,713	69.93

● POS_g : 一般品詞(例えば、名詞・動詞・形容詞)

● POS_s : 特定品詞 (細かく区別)

結果

TABLE III: Summary of identical tokens, tokens/POS_g, and tokens/POS_s for all libraries

Source	average tokens	identical tokens	identical tokens %	identical token/POS _g	identical token/POS _g %	identical token/POS _s	identical token/POS _s %
Stack Overflow	8,608	7,899	91.77	6,559	76.20	6,105	70.92
GitHub	6,242	5,870	94.05	4,443	71.18	3,843	61.57
Java API	11,158	9,947	89.15	7,586	67.99	6,659	59.68

- POS_g : 一般品詞(例えば、名詞・動詞・形容詞)
- POS_s : 特定品詞 (細かく区別)

アプローチ2

1. StackOverFlow , Github ReadMe , Java APIに対して各ライブラリを適用
2. 各文章を手動で単語分割,形態素解析を行う
3. 各ライブラリと手動を比較

結果

TABLE IV: Accuracy based on comparing the manual annotation with the output of the four libraries

Comparison		tokens for manual annotation	tokens for library	identical tokens	identical tokens %	identical token/POS _g	identical token/POS _g %	identical token/POS _s	identical token/POS _s %
Stack Overflow	Manual vs. Stanford	375	385	371	97.63	339	89.21	317	83.42
	Manual vs. SyntaxNet	375	366	357	96.36	332	89.61	317	85.56
	Manual vs. spaCy	375	377	369	98.14	347	92.29	338	89.89
	Manual vs. NLTK	375	374	373	99.60	331	88.38	306	81.71
GitHub READMEs	Manual vs. Stanford	361	371	357	97.54	310	84.70	286	78.14
	Manual vs. SyntaxNet	361	358	350	97.36	308	85.67	297	82.61
	Manual vs. spaCy	361	370	344	94.12	309	84.54	291	79.62
	Manual vs. NLTK	361	360	354	98.20	312	86.55	278	77.12
Java API Docs	Manual vs. Stanford	380	398	374	96.14	328	84.32	303	77.89
	Manual vs. SyntaxNet	380	379	373	98.29	298	78.52	286	75.36
	Manual vs. spaCy	380	380	369	97.11	345	90.79	298	78.42
	Manual vs. NLTK	380	382	379	99.48	345	90.55	294	77.17

- 単語分割はNLTKが高い精度を示す
- 形態素解析

まとめ

- Stack Overflow , Github ReadMe , Java API Documentsで各ライブラリの単語分割,品詞付けを比較した
- 各ライブラリと手動で単語分割,品詞付けを比較した
 - ▶ ほとんどの研究でしようされていなかったspaCyが最も高い精度を示した
- 高い精度を示す文章とそうでない文章が存在する

所感

- 今まで使われていなかったライブラリが高い精度を持っていたためNLPを選ぶ基準になる
- 実際にライブラリの違いによる影響はどの程度あるのか？