IEEE Transactions on Software Engineering, 2019

Bridging Semantic Gaps between Natural Languages and APIs with Word Embedding

Authors: **Xiaochen Li**[1], He Jiang[1], Yasutaka Kamei[1], Xin Chen[2]
[1]Dalian University of Technology, [2]Kyushu University, [3]Hangzhou Dianzi University

BACKGROUND

Gaps between natural languages and APIs

High-level words
Low-level APIs

A simple requirement "Read a File" may call many APIs to implement.

```
public void readFile(String path) throw IOException {
    File file = new File(path);
    FileReader fr = new FileReader(file);
    BufferedReader br = new BufferedReader(fr);
    String line = "";
    while (null != (line = br.readLine()) {
        System.out.println(line);
    }
    br.close();
}
```

BACKGROUND

Semantic gaps negatively affect SE processes

Developers

Text-matching based tools

Hinder developers from comprehending APIs and bring **thousands of defects** in API documents

A tool **only returns 25.7% to 38.4%** useful code snippets in top-10 results for user queries

Y. Zhou et al. "Analyzing APIs documentation and code to detect directive defects," ICSE'17
F. Lv, et al. "Codehow: Effective code search based on API understanding and extended boolean model," ASE'15

BACKGROUND

Semantic gaps negatively affect SE processes

Developers

Text-matching based tools

Hinder developers from comprehending APIs and bring

A tool **only returns 25.7% to 38.4%** useful code snippets for

How to bridge the gaps?

Semantic Estimation

Y. Zhou et al. "Analyzing APIs documentation and code to detect directive defects," ICSE'17
F. Lv, et al. "Codehow: Effective code search based on API understanding and extended boolean model," ASE'15

BACKGROUND

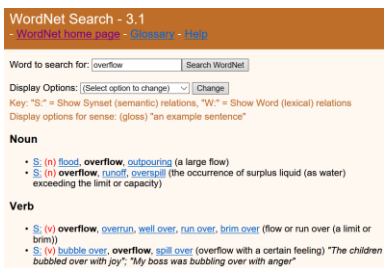
Classical algorithms to bridge the gaps

Calculating semantic relatedness / **similarity** between **a word and an API** or **a set of words and APIs**

WordNet thesaurus analysis

Latent semantic analysis (LSA or LSI)

Co-occurrence analysis



$\{X\} =$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

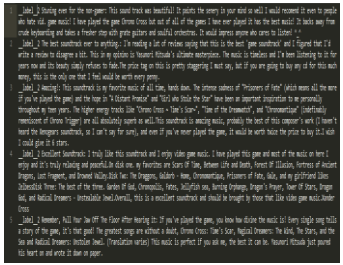
Section: Working with views and editors
Topic: Maximizing and minimizing in the eclipse presentation
Content: The minimize behavior for the Editor Area is somewhat different; minimizing the Editor Area results in a trim stack containing only a placeholder icon representing the entire editor area rather than icons for each open editor
...

MOTIVATION

Word embedding for word representing

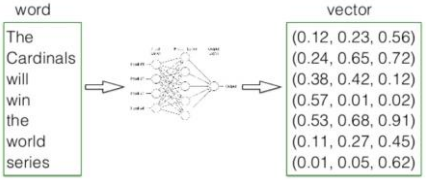
A **co-occurrence based** algorithm. Generating **low-dimensional vectors** for words or APIs.

Prepare a dataset → Prepare word2vec tool → Run CBOW or Skip-gram model



word2vec

Tool for computing continuous dis

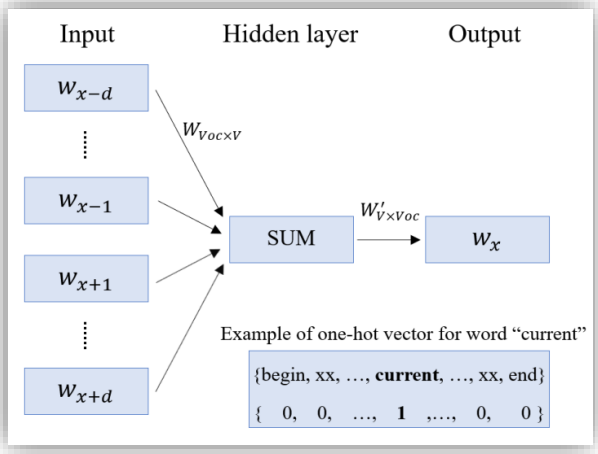


MOTIVATION

Continuous Bag-Of-Words model CBOW

Interface `IPageLayout`
Description: A page layout defines the initial layout for a perspective within a page in a workbench window... View placeholders may also have a secondary id. ... For example, the placeholder "someView:*" will match any occurrence of the view that has primary id "someView" and that also has some non-null secondary id. Note that this placeholder will not match the view if it has no secondary id ...

Generate vectors of **center words** with their **surrounding context**



$$L_M = \frac{1}{X} \sum_{x=1}^{X} \log p(w_x | W_x^d)$$

(0.12, 0.23, 0.56)
(0.24, 0.65, 0.72)
(0.38, 0.42, 0.12)
(0.57, 0.01, 0.02)
(0.53, 0.68, 0.91)
(0.11, 0.27, 0.45)
(0.01, 0.05, 0.62)

MOTIVATION

There are challenges for word-API learning

Acquisition challenge
collect large numbers of documents that contain diversity words and APIs

Alignment challenge
align words and APIs to fully mine their overall relationship in a window

Interface `IPageLayout`
Description: A page layout defines the initial layout for a perspective within a page in a workbench window... View placeholders may also have a secondary id. ... For example, the placeholder "someView:*" will match any occurrence of the view that has primary id "someView" and that also has some non-null secondary id. Note that this placeholder will not match the view if it has no secondary id ...

DocumentView#getDefaultView()?

ComponentView#new()?

Interface `IPageLayout`
Description: A page layout defines the initial layout for a perspective within a page in a workbench window... View

```
public void readFile(String path) throw IOException {
    File file = new File(path);
    FileReader fr = new FileReader(file);
    BufferedReader br = new BufferedReader(fr);
    String line = "";
    while (null != (line = br.readLine()) {
        System.out.println(line);
    }
    br.close();
}
```

No word-API collocations

FRAMEWORK

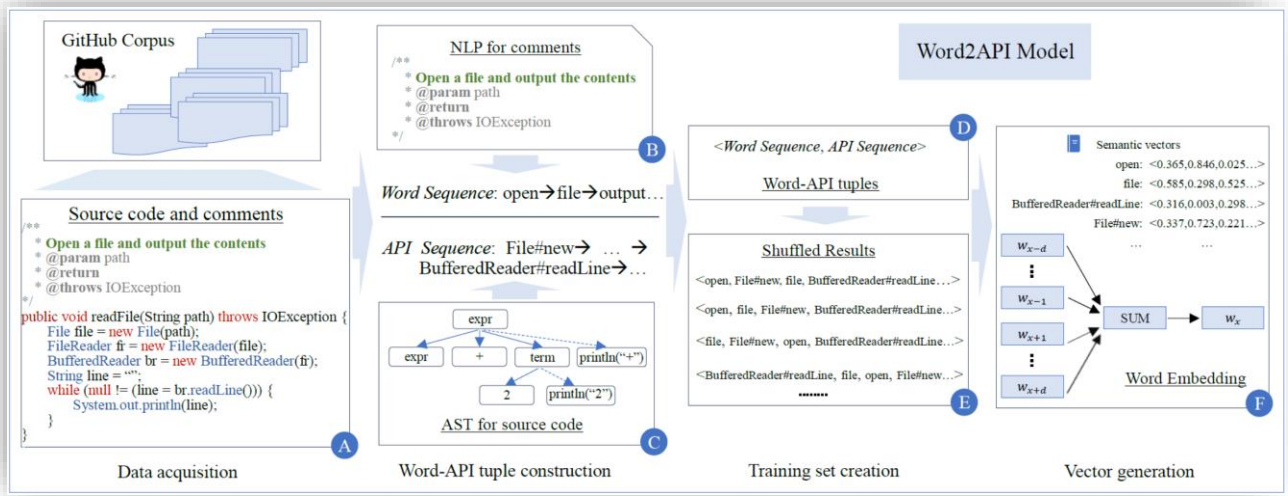
Address the challenges by our model Word2API

A Collect source code & APIs from GitHub (**Acquisition**)

B Pre-process words & APIs with NLP and AST

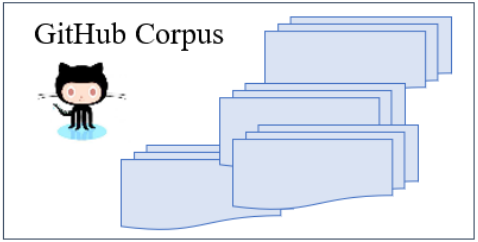
C Shuffle words & APIs in a method (**Alignment**)

D Run word embedding to generate vectors



FRAMEWORK

A. Collect source code & APIs from GitHub



Source code and comments

```
/**
  * Open a file and output the contents
  * @param path
  * @return
  * @throws IOException
*/
public void readFile(String path) throws IOException {
    File file = new File(path);
    FileReader fr = new FileReader(file);
    BufferedReader br = new BufferedReader(fr);
    String line = "";
    while (null != (line = br.readLine())) {
        System.out.println(line);
    }
}
```

A

GitHub from 2008-2016
391,690 Java projects
31,211,030 source code files

Extract words in the **method comment** and API calls in the **method body**.

These words and APIs are **widely used by developers**

FRAMEWORK

B. Pre-process words & APIs with NLP and AST

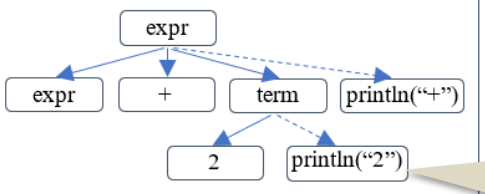
NLP for comments

```
/**
 * Open a file and output the contents
 * @param path
 * @return
 * @throws IOException
 */
```

B

Word Sequence: open→file→output…

API Sequence: File#new→ … → BufferedReader#readLine→…



AST for source code

C

Word sequence

Tokenization
Stop word removal
Stemming

< open, file, output, content >

13,883,230 **word-API tuples**

<word1, word2, ..., API1, API2...>

API sequence

AST
Find API fully qualified names

< File#new, FileReader#new, BufferedReader, String#new, BufferedReader#readLine >

FRAMEWORK

C. Shuffle words & APIs in a method

Problem of word-API tuples:
Words & APIs do not appear within each other's window

<open, file, output, contents, …, BufferedReader#readLine>

Shuffling strategy:
Words & APIs in the same word-API tuple contain valuable semantic information (relatedness) for mining

FRAMEWORK

C. Shuffle words & APIs in a method

Word-API tuples

<open, file, output, contents, …, BufferedReader#readLine>

Shuffled Results

<open, File#new, file, BufferedReader#readLine…>

<open, file, File#new, BufferedReader#readLine…>

<file, File#new, open, BufferedReader#readLine…>

<BufferedReader#readLine, file, open, File#new…>

……..

E

Increase the information interaction (collocations of words & APIs)

Help word embedding **learn** the knowledge of word-API (**the overall knowledge of each tuple**)

138,832,300 shuffled results (>30 GB)

D. Run word embedding to generate vectors

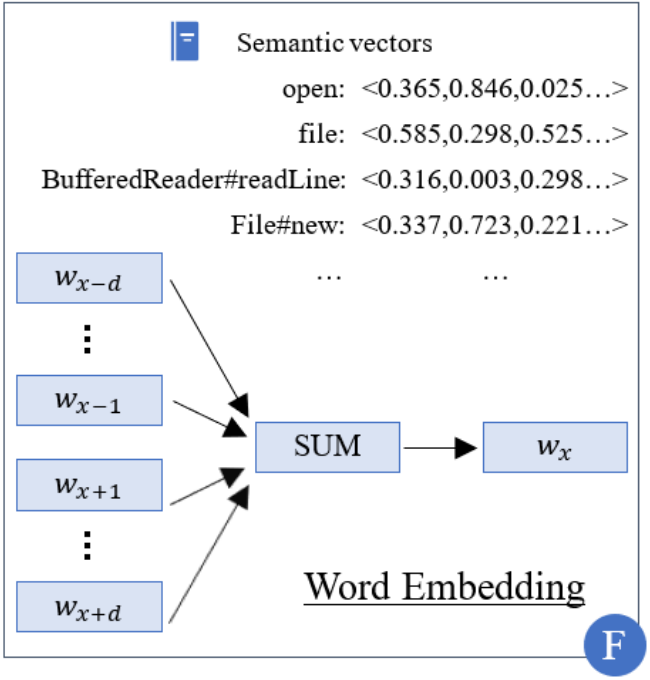
87,270 word vectors
37,431 API vectors
Semantic estimation with these vectors

- Word-API similarity

$$sim(w,a) = \frac{\overrightarrow{V_w} \cdot \overrightarrow{V_a}}{|\overrightarrow{V_w}||\overrightarrow{V_a}|}.$$

- Words-APIs similarity

$$sim(W,A) = \frac{1}{2}\left(\frac{\sum(maxSim(w,A) \times idf(w))}{\sum idf(w)} + \frac{\sum(maxSim(a,W) \times idf(a))}{\sum idf(a)}\right),$$



EVALUATION

Recommend APIs by a query word

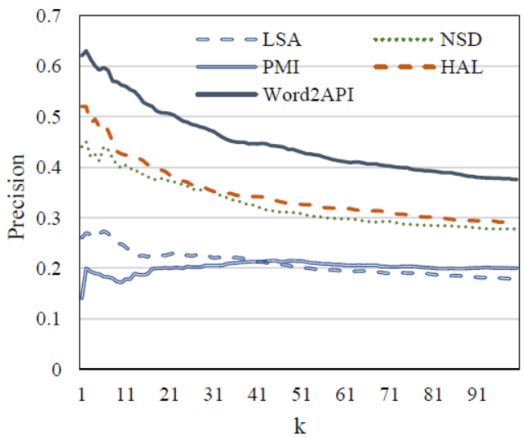
- Random selected 50 Noun. & Verb.

#	Word	#	Word	#	Word	#	Word	#	Word
1	agent	11	delete	21	key	31	random	41	tail
2	average	12	display	22	length	32	remote	42	thread
3	begin	13	environment	23	mp3	33	request	43	timeout
4	buffer	14	file	24	next	34	reserve	44	transaction
5	capital	15	filter	25	node	35	scale	45	uuid
6	check	16	graphics	26	object	36	select	46	validity
7	classname	17	http	27	open	37	session	47	word
8	client	18	input	28	parse	38	startup	48	xml
9	current	19	interrupt	29	port	39	string	49	xpath
10	day	20	iter	30	post	40	system	50	year

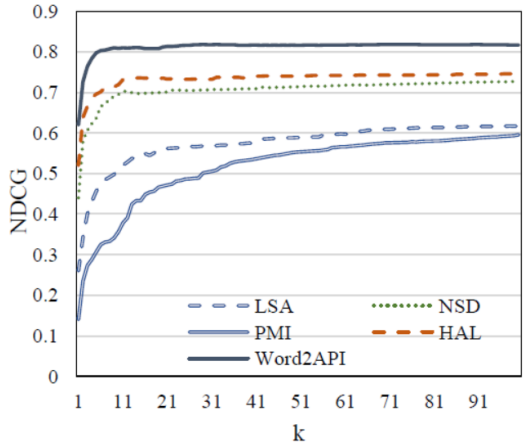
- Comparative algorithm
  - LSA (Latent Semantic Analysis)
  - PMI (Pointwise Mutual Information)
  - NSD (Normalized Software Distance)
  - HAL (Hyperspace Analogue to Language)

EVALUATION

Word2API captures the relatedness of words & APIs



(a) Evaluation on precision



(b) Evaluation on NDCG

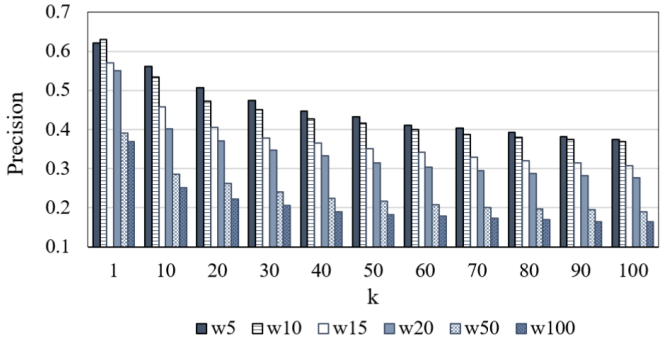
- Word2API **outperforms the baseline algorithms**;
- Volunteers judgement the relatedness between the words and the recommended APIs by different algorithms.

EVALUATION

Window size and the shuffling strategy

Increase window size, but performance drops

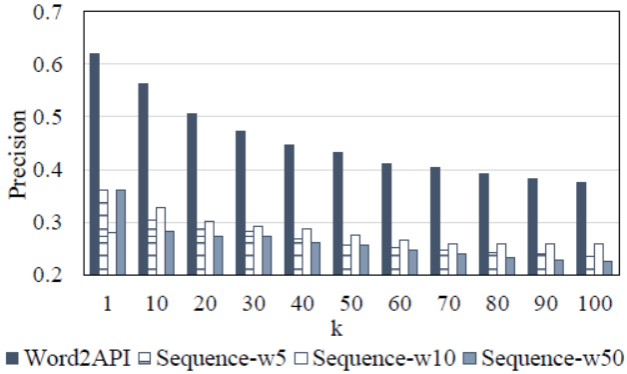
Cannot learn word embedding by simply increasing window_size (w)



(a) Precision@k

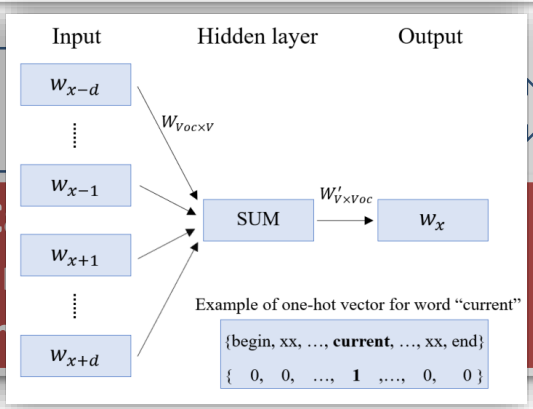
Sequence-w5: no shuffling,window_size=5

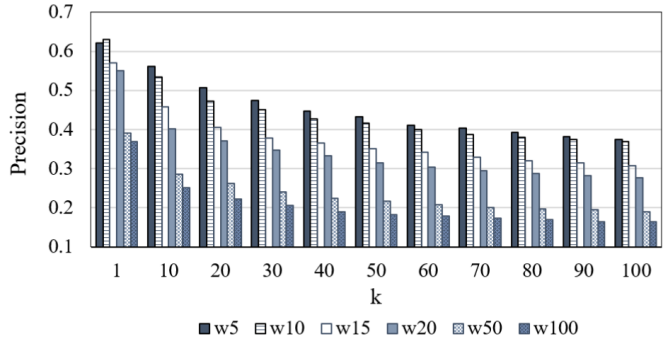
Shuffling is significantly better than no shuffling



EVALUATION

Window size and the shuffling strategy

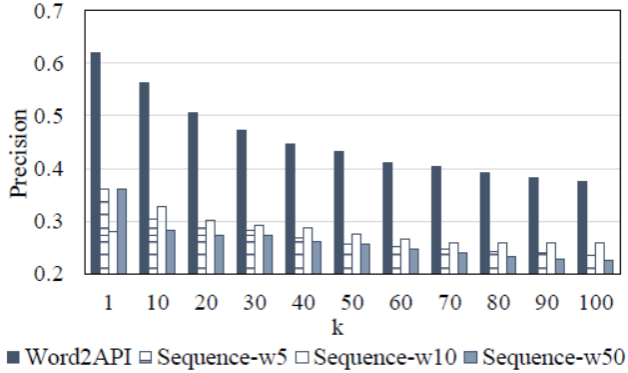




(a) Precision@k

Sequence-w5: no shuffling,window_size=5

Shuffling is significantly better than no shuffling



APPLICATION 1

Expand user query into an API vector for API sequences recommendation

Query

"read a file"

API Sequence

- java.io.File#new,
- java.io.FileReader#new,
- java.io.BufferedReader#new,
- java.io.BufferedReader#readLine

These sequences are retrieved from source code corpus, e.g. GitHub corpus.

APPLICATION 1

Expand user query into an API vector for API sequences recommendation

Query

"read a file"

API Sequence

- java.io.File#new,
- java.io.FileReader#new,
- java.io.BufferedReader#new,
- java.io.BufferedReader#readLine

These sequences are retrieved from source code corpus, e.g. GitHub corpus.

API vector

- **SWIM**: Word Alignment based Augmentation
- **CodeHow**: API Description based Augmentation
- **Word2API** based Augmentation

APPLICATION 1

TABLE III: Performance of query augmentation algorithms over 30 human written queries.

ID	Query	SWIM			CodeHow			Word2API		
		FR	RR5	RR10	FR	RR5	RR10	FR	RR5	RR10
Q1	convert int to string	11	0	0	11	0	0	3	0.2	0.1
Q2	convert string to int	1	1	0.5	11	0	0	1	0.8	0.8
Q3	append string	1	1	1	1	1	1	1	1	1
Q4	get current time	11	0	0	11	0	0	1	1	1
Q5	parse datetime from string	10	0	0.1	11	0	0	1	1	0.7
Q6	test file exists	1	1	1	1	1	1	1	0.8	0.8
Q7	open a url	1	1	1	1	1	1	1	0.8	0.8
Q8	open file dialog	11	0	0	1	0.8	0.7	1	0.4	0.7
Q9	get files in folder	11	0	0	1	0.8	0.9	1	1	0.9
Q10	match regular expressions	1	1	0.8	1	0.6	0.7	1	1	1
Q11	generate md5 hash code	11	0	0	11	0	0	1	1	1
Q12	generate random number	1	0.4	0.2	1	1	1	1	1	1
Q13	round a decimal value	11	0	0	2	0.2	0.1	1	0.8	0.8
Q14	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.5
Q15	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1
Q16	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	1
Q17	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.5
Q18	copy a file and save it to your destination path	1	1	1	2	0.2	0.5	1	0.8	0.9
Q19	delete files and folders in a directory	1	1	1	3	0.6	0.4	4	0.4	0.4
Q20	reverse a string	11	0	0	11	0	0	11	0	0
Q21	create socket	11	0	0	1	0.6	0.4	1	1	0.9
Q22	rename a file	11	0	0	11	0	0	4	0.4	0.5
Q23	download file from url	1	1	0.7	1	1	1	5	0.2	0.3
Q24	serialize an object	1	1	1	1	1	1	1	1	1
Q25	read binary file	1	1	0.6	1	1	1	1	0.8	0.8
Q26	save an image to a file	1	1	1	1	1	1	5	0.2	0.4
Q27	write an image to a file	1	1	1	1	0.8	0.6	2	0.4	0.3
Q28	parse xml	11	0	0	11	0	0	1	0.2	0.3
Q29	play audio	11	0	0	1	0.8	0.9	1	0.4	0.5
Q30	play the audio clip at the specified absolute URL	11	0	0	1	1	1	1	0.6	0.4
	Average scores over 30 queries	5.633	0.513	0.463	4.467	0.547	0.533	1.933	0.680	0.677

Position of first correct API seq. : lower is better

Ratio of correct API seq.: higher is better

APPLICATION 2

Link API doc. with Stack Overflow questions

Question: "Are there any good CachedRowSet implementations other than the proprietary Sun one?"

Answer:

13

You shouldn't be directly instantiating implementation of CachedRowSet -- use its Provider to obtain an instance: see http://docs.oracle.com/javase/7/docs/api/javax/sql/rowset/RowSetProvider.html (available since JDK7)

In fact, CachedRowSet's interface and related factory are standard/portable.

Linkage

Overview Package Class Use Tree Deprecated Index Help

Prev Class Next Class Frames No Frames All Classes

Summary: Nested | Field | Constr | Method Detail: Field | Constr | Method

javax.sql.rowset

Class RowSetProvider

java.lang.Object
javax.sql.rowset.RowSetProvider

```
public class RowSetProvider
extends Object
```

A factory API that enables applications to obtain a RowSetFactory implementation that can be used to create different types of RowSet implementations.

Example:

```
RowSetFactory aFactory = RowSetProvider.newFactory();
CachedRowSet crs = aFactory.createCachedRowSet();
....
RowSetFactory rsf = RowSetProvider.newFactory("com.sun.rowset.RowSetFactoryImpl", null);
WebRowSet wrs = rsf.createWebRowSet();
```

Tracing of this class may be enabled by setting the System property javax.sql.rowset.RowSetFactory.debug to any value but false.

Since:

1.7

APPLICATION 2

Word2API for API Documents Linking

- **Collect words in the question** "*Are there any good CachedRowSet implementations other than the proprietary Sun one?*"

1. Transform **words and APIs into vectors** with Word2API
2. Rank API documents by **words-APIs similarity**

$$sim(W, A) = \frac{1}{2}\left(\frac{\sum (maxSim(w, A) \times idf(w))}{\sum idf(w)} + \frac{\sum (maxSim(a, W) \times idf(a))}{\sum idf(a)}\right),$$

- **Collect APIs in each API document**
  - *javax.sql.rowset.RowSetProvider#newFactory*
  - *javax.sql.rowset.RowSetProvider#createCachedRowSet*
  - *……*

APPLICATION 2

Word2API can bridge the gaps of documents

- MAP: Mean Average Precision
- MRR: Mean Reciprocal Rank

- Algorithms
  - VSM: vector space model
  - Embedding: previous work
  - VSM+XXX: combined

TABLE V: MAP and MRR for API document linking.

Algorithms	MAP	MRR
VSM	0.232	0.259
Embedding	0.313	0.354
Word2API	0.402	0.433
VSM+Embedding	0.340	0.380
VSM+Word2API	0.436	0.469

Word embedding is better that VSM
We can combine Word2API with other techniques for better results

Ye et al. From Word Embeddings To Document Similarities for Improved Information Retrieval in Software Engineering (ICSE16)

CONCLUSION

- We propose Word2API to solve the problem of **constructing low-dimensional representations** for both words and APIs **simultaneously**.
- With Word2API, we generate **126,853 word and API vectors to bridge the sematic gaps** between natural language words and APIs.
- We show **two applications of Word2API**. Word2API improves the performance of two typical software engineering tasks, i.e., API sequences recommendation and API documents linking.

Thanks

Bridging Semantic Gaps between Natural Languages and APIs with Word Embedding

Reporter： Xiaochen Li
Dalian University of Technology, China

Authors: **Xiaochen Li**[1], He Jiang[1], Yasutaka Kamei[1], Xin Chen[2]
[1]Dalian University of Technology, [2]Kyushu University, [3]Hangzhou Dianzi University