深度学习与自然语言处理

以软件仓库挖掘为例

王秋里

智能服務與軟件工程中心
Center for Intelligent Services and Software Engineering

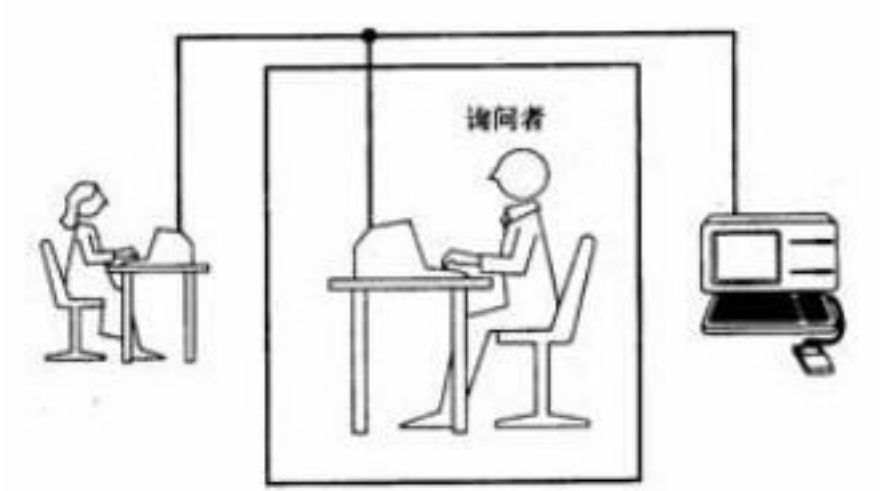
A.自然语言处理发展历程

B.深度学习与自然语言处理

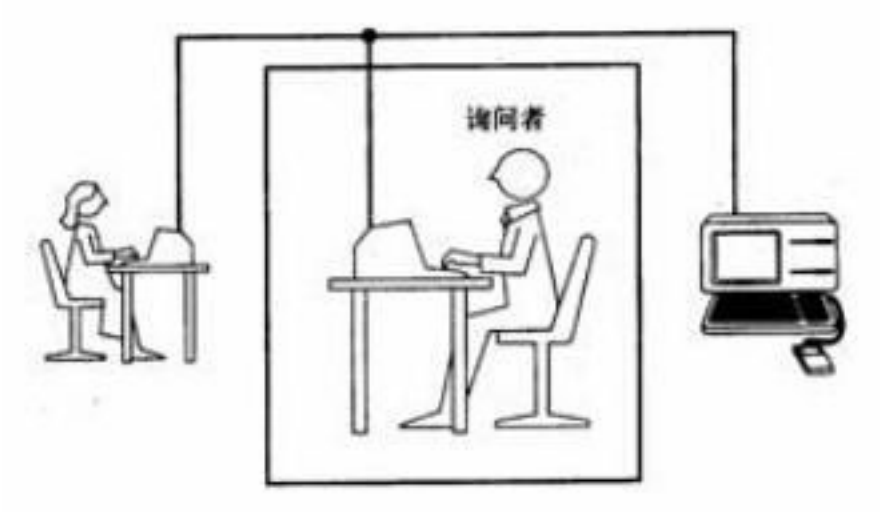
C.自然语言处理与软件仓库挖掘

自然语言处理

A 自然语言处理



A 自然语言处理



图灵测试

A

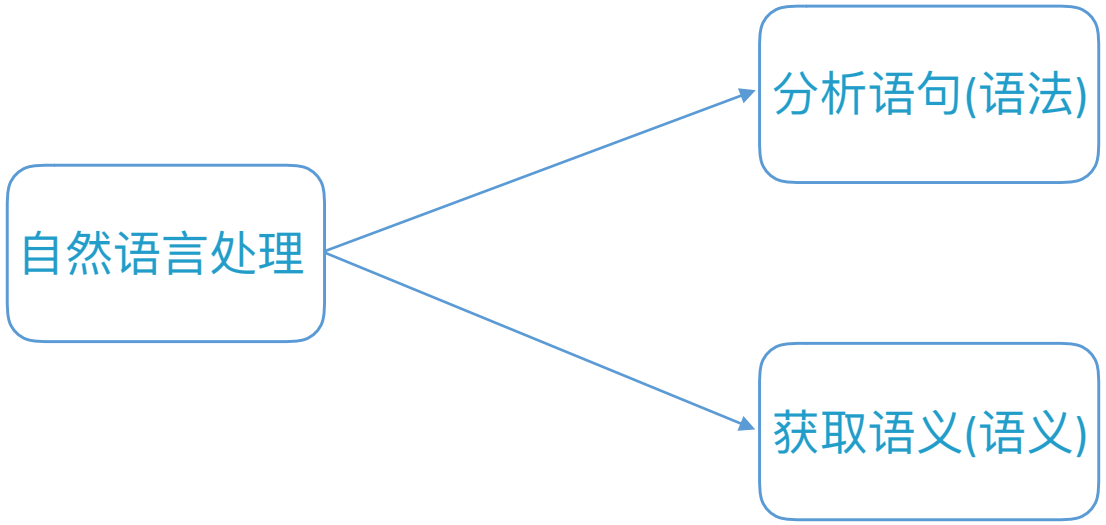
自然语言处理

第一阶段：20世纪50年代到70年代

要让机器完成翻译或者语音识别等只有人来才能做的事情
1.就必须先让计算机理解自然语言
2.而做到这一点就必须让计算机拥有类似我们人类这样的智能

A 自然语言处理

第一阶段：20世纪50年代到70年代



A

自然语言处理

第一阶段：20世纪50年代到70年代

学习一门语言(西方语言)

1 语法规则(Grammar Rules)

2 词性(Part of Speech)

3 构词法(Morphologic)

A

自然语言处理

第一阶段：20世纪50年代到70年代

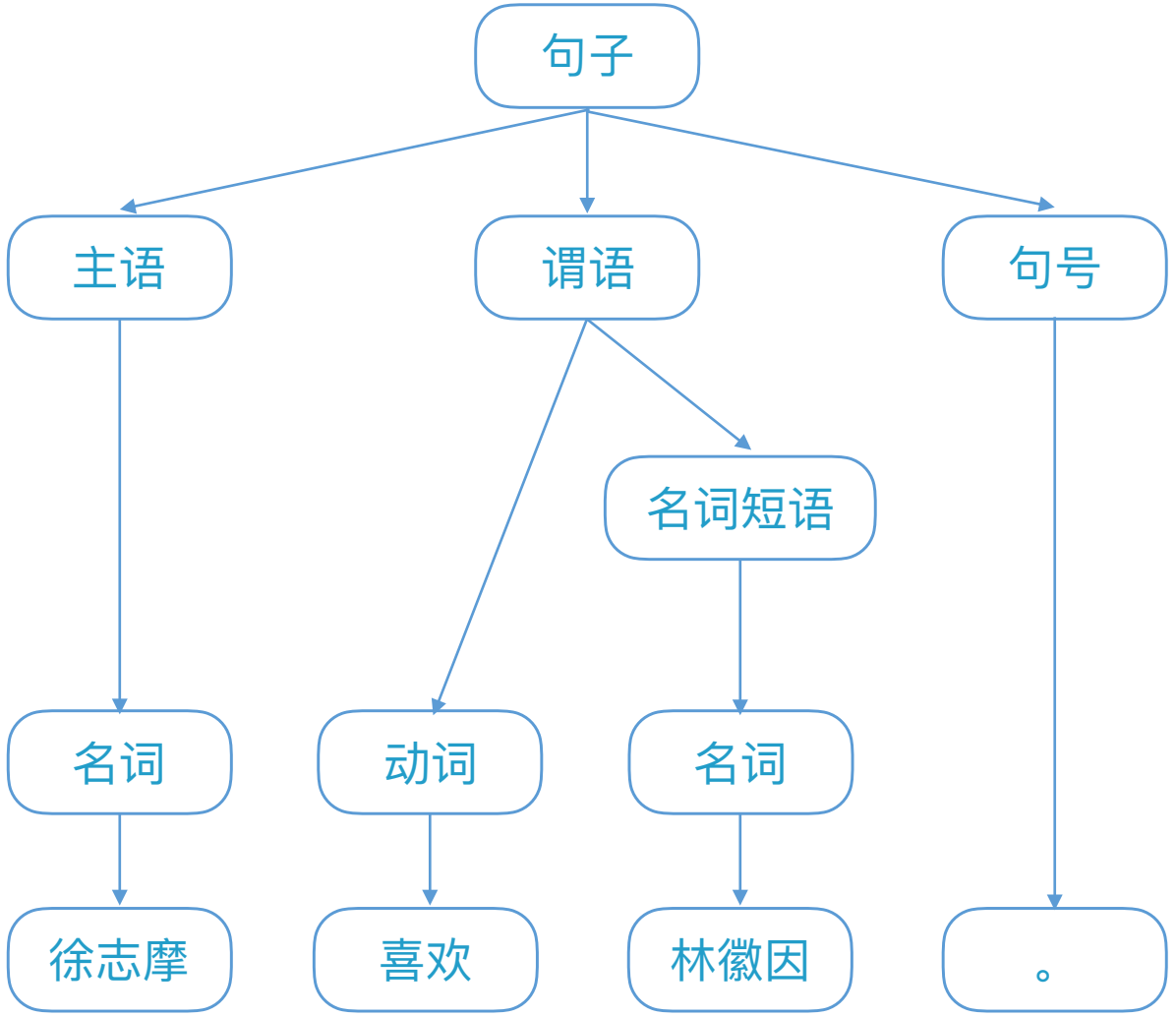
徐志摩喜欢林徽因

A

自然语言处理

第一阶段：20世纪50年代到70年代

徐志摩喜欢林徽因



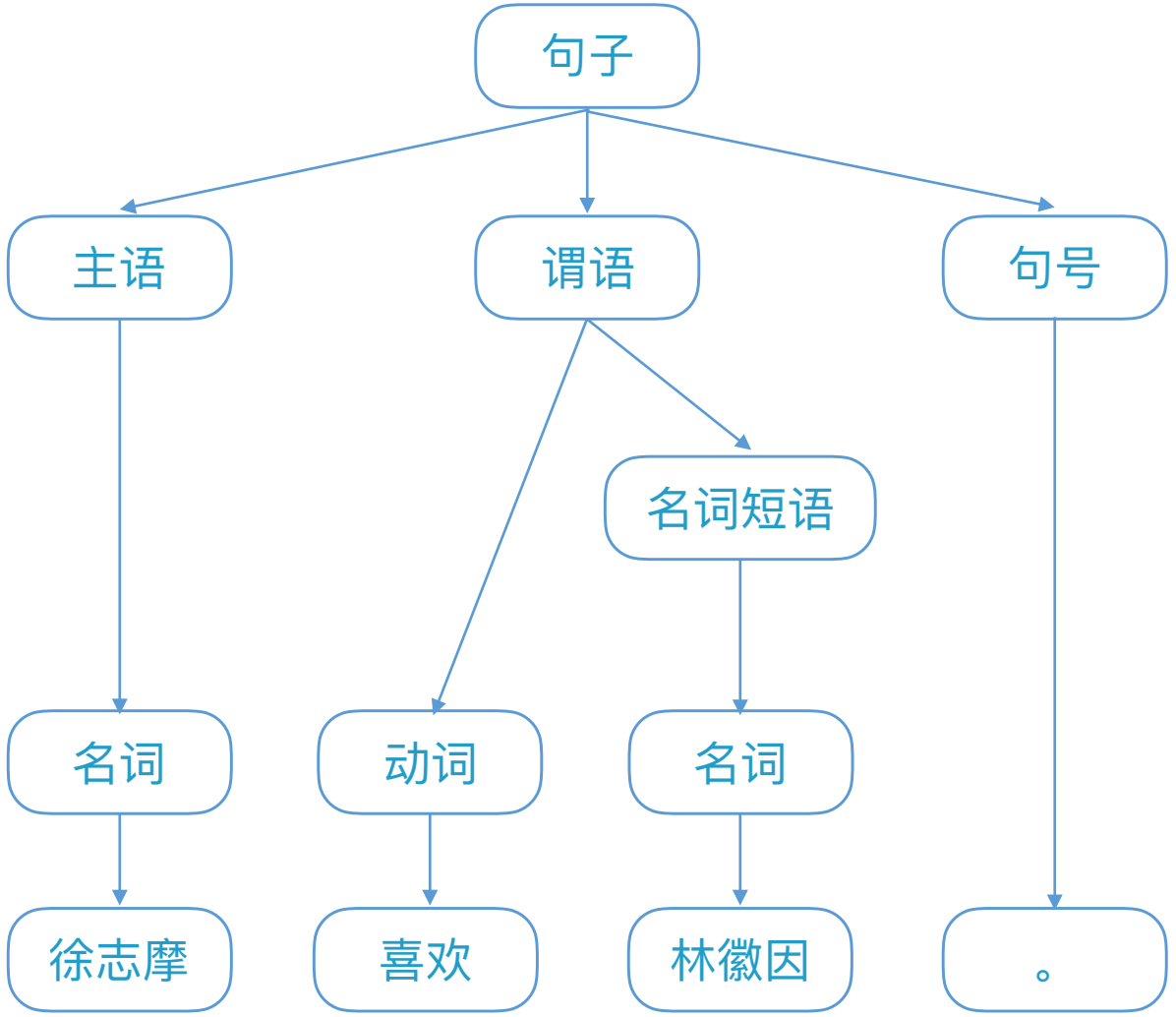
A

自然语言处理

第一阶段：20世纪50年代到70年代

徐志摩喜欢林徽因

美联储主席本-伯南克昨天告诉媒体7000亿美元的救助资金将借给上百家银行、保险公司和汽车公司。 ？



A

自然语言处理

第一阶段：20世纪50年代到70年代

The pen is in the box ——→ 笔在盒子里

The box is in the pen ——→ 盒子在笔里 ？

第一阶段：20世纪50年代到70年代

The pen is in the box ⟶ 笔在盒子里

The box is in the pen ⟶ 盒子在笔里 ？

盒子在围栏里 ！

A

自然语言处理

第二阶段：20世纪70年代以后
从规则到统计

弗里德里克.贾里尼克

IBM华生实验室

Frederick Jelinek (18 November 1932 – 14 September 2010)

第二阶段：20世纪70年代以后
从规则到统计

1.美联储主席本-伯南克昨天告诉媒体7000亿美元的救助资金将借给上百家银行、保险公司和汽车公司。

2.本-伯南克美联储主席昨天7000亿美元的救助资金告诉媒体将借给银行、保险公司和汽车公司上百家。

3.联储美主席南克告助资金将借本-伯给上司和汽车公司昨天诉媒体7000亿百家美元的救银行、保险公。

A

自然语言处理

第二阶段：20世纪70年代以后
从规则到统计

一个句子是否合理，就看它的可能性大小如何

1.美联储主席本-伯南克昨天告诉媒体7000亿美元的救助资金将借给上百家银行、保险公司和汽车公司。 10^{-20}

2.本-伯南克美联储主席昨天7000亿美元的救助资金告诉媒体将借给银行、保险公司和汽车公司上百家。 10^{-25}

3.联储美主席南克告助资金将借本-伯给上司和汽车公司昨天诉媒体7000亿百家美元的救银行、保险公。 10^{-70}

第二阶段：20世纪70年代以后
从规则到统计

一个句子是否合理，就看它的可能性大小如何

假定S表示一个有意义的句子

$S = (w_1, w_2, w_3, w_4, w_5, w_6 \ldots\ldots w_n)$

n为句子长度

$P(S) = P(w_1, w_2, w_3, w_4, w_5, w_6 \ldots\ldots w_n)$

$P(S) = P(w_1)P(w_2|w_1)P(w_3|w_1,w_2)\ldots\ldots$
$P(w_n,|w_1,w_2,w_3,\ldots w_{n-1})$

第二阶段：20世纪70年代以后
从规则到统计

一个句子是否合理，就看它的可能性大小如何

俄国数学家马尔可夫(Andrey Markov)

马尔可夫假设:

假设任意一个词w_i出现的概率只同它前面的词w_{i-1}有关

$$P(S)=P(w_1)P(w_1|w_2)P(w_3|w_2)\ldots P(w_n|w_{n-1})$$

深度学习与自然语言处理

B

深度学习与自然语言处理

深度学习，从基本的层面来说是表征学习

我们要将每一个单词都表征为一个d维向量

```
Uninterested = [_ _ _ _ _ _]
```

我们希望通过填写值的方式可以让向量表征词，以及词的语境、意思或者语音

建立一个共生矩阵(concurrence matrix)

I love NLP and I like dogs

建立一个共生矩阵(concurrence matrix)

I love NLP and I like dogs

```
I = [0 1 0 1 1 0]
Love = [1 0 1 0 0 0]
NLP = [0 1 0 1 0 0]
And = [1 0 1 0 0 0]
Like = [1 0 0 0 0 1]
Dogs = [0 0 0 0 1 0]
```

建立一个共生矩阵(concurrence matrix)

I love NLP and I like dogs

	I	Love	NLP	And	Like	Dogs
I	0	1	0	1	1	0
Love	1	0	1	0	0	0
NLP	0	1	0	1	0	0
And	1	0	1	0	0	0
Like	1	0	0	0	0	1
Dogs	0	0	0	0	1	0

建立一个共生矩阵(concurrence matrix)

I love NLP and I like dogs

建立一个共生矩阵(concurrence matrix)

I love NLP and I like dogs

$$J(\theta) = \frac{1}{T}\sum_{t=1}^{T}\sum_{-m\leq j\leq m, j\neq 0} \log p(w_{t+j}|w_t)$$

Word2Vec

建立一个共生矩阵(concurrence matrix)

I love NLP and I like dogs

$$X_{shirt} - X_{clothing} \approx X_{chair} - X_{furniture}$$

$$X_{king} - X_{man} \approx X_{queen} - X_{woman}$$

B

深度学习与自然语言处理

循环神经网络(RNN)

B

深度学习与自然语言处理

循环神经网络(RNN) Why?

B

深度学习与自然语言处理

循环神经网络(RNN) Why?

Finding Structure in Time

循环神经网络(RNN) Why?

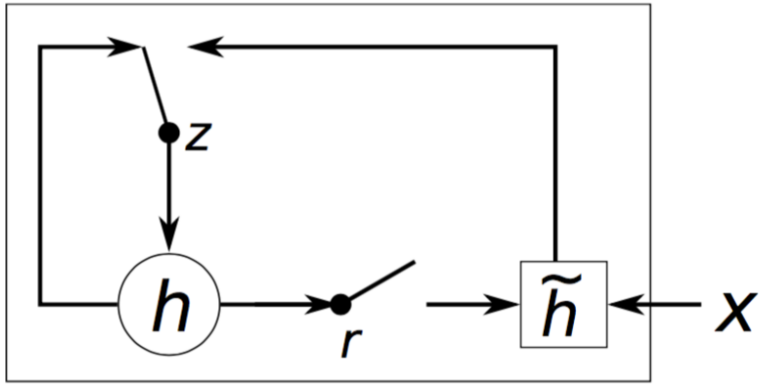
Finding Structure in Time

Other NNs’ drawbacks

The input should be presented all at once

The input layer must provide for the longest possible pattern

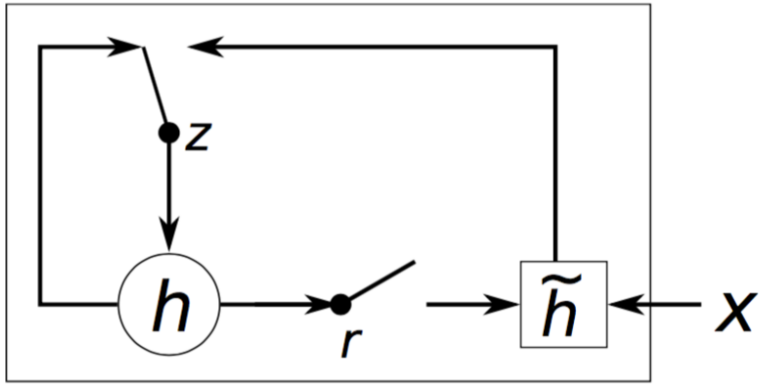
循环神经网络(RNN)
门控循环单元(gated recurrent unit / GRU)



传统RNN中
隐藏状态向量是通过该公式计算的

$$h_t = f\left(W^{(hh)}h_{t-1} + W^{(hx)}x_t\right)$$

循环神经网络(RNN)
门控循环单元(gated recurrent unit / GRU)



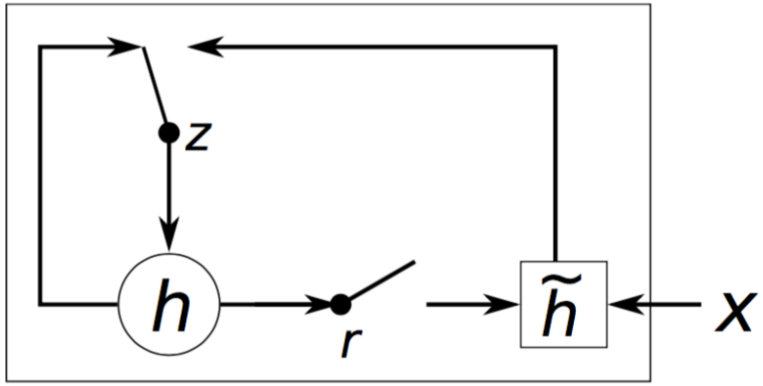
GRU提供

一个更新门(update gate) — z

一个重置门(reset gate) — r

一个新的记忆存储器(memory container) — h’

循环神经网络(RNN)
门控循环单元(gated recurrent unit / GRU)

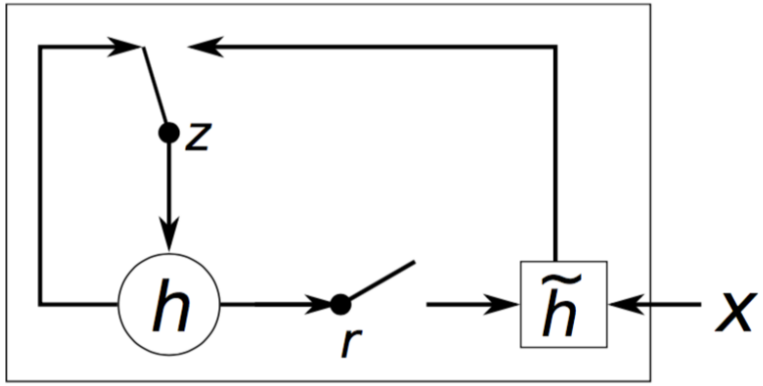


$$z_t = \sigma\left(W^{(z)}x_t + U^{(z)}h_{t-1}\right)$$

$$r_t = \sigma\left(W^{(r)}x_t + U^{(r)}h_{t-1}\right)$$

$$\tilde{h}_t = \tanh\left(Wx_t + r_t \circ Uh_{t-1}\right)$$

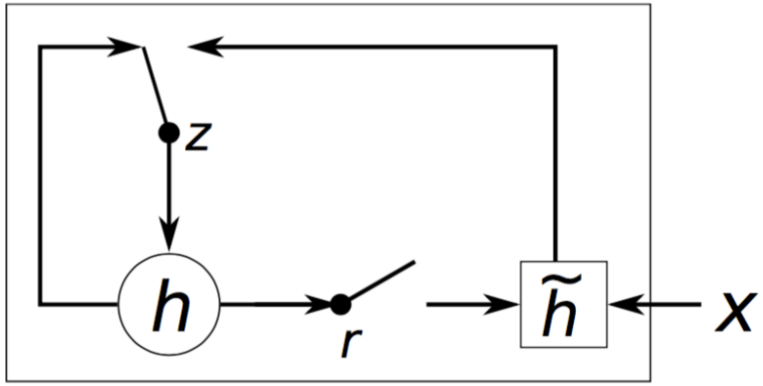
循环神经网络(RNN)
门控循环单元(gated recurrent unit / GRU)



$$h_t = f\left(W^{(hh)}h_{t-1} + W^{(hx)}x_t\right)$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

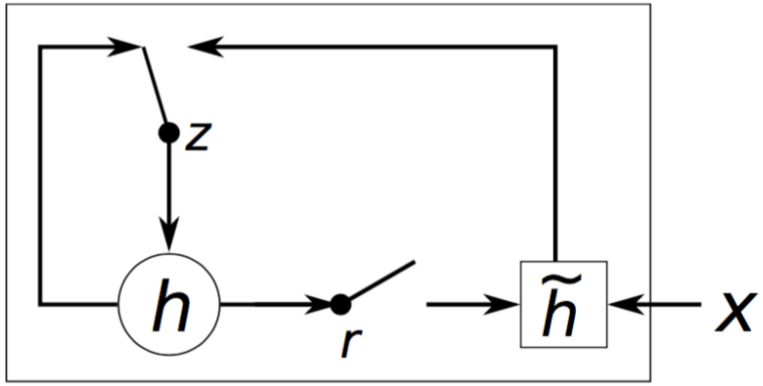
循环神经网络(RNN)
门控循环单元(gated recurrent unit / GRU)



$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

更新门
如果zt趋向于1，ht就完全忽略现在的词向量，仅仅是复制前隐藏状态。
如果zt趋向于0，ht就完全忽略前一时间步骤的隐藏状态，仅仅只依赖于新的记忆存储器。

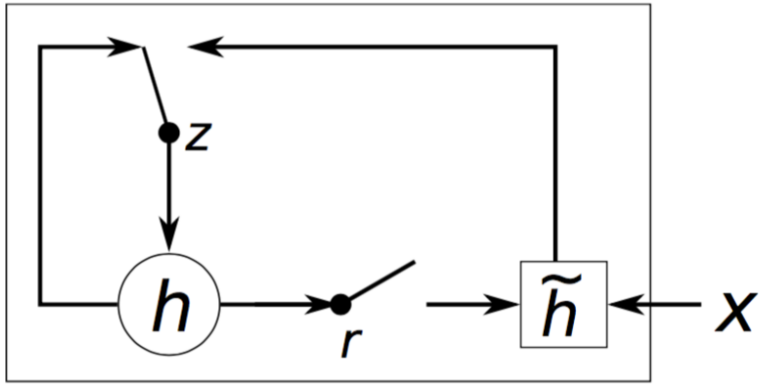
循环神经网络(RNN)
门控循环单元(gated recurrent unit / GRU)



$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

重置门
如果rt趋向于1，记忆存储器将保持前一隐藏状态的信息。
如果rt趋向于0，记忆存储器将忽略前一隐藏状态的信息。
此门控能允许模型丢弃一些对未来不相干的信息。

循环神经网络(RNN)
RNN Encoder-Decoder



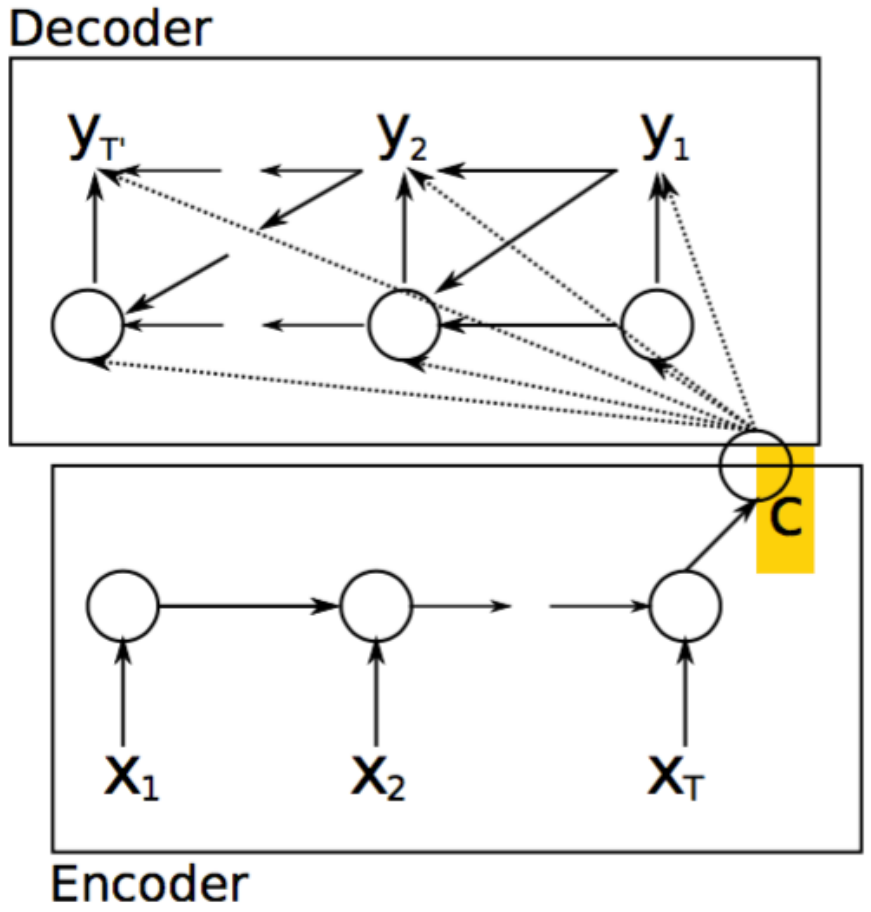
$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

Those units that learn to capture ***short-term*** dependencies will tend to have ***reset gates*** that are frequently active.

But those that capture ***longer-term*** dependencies will have ***update gates*** that are mostly active.

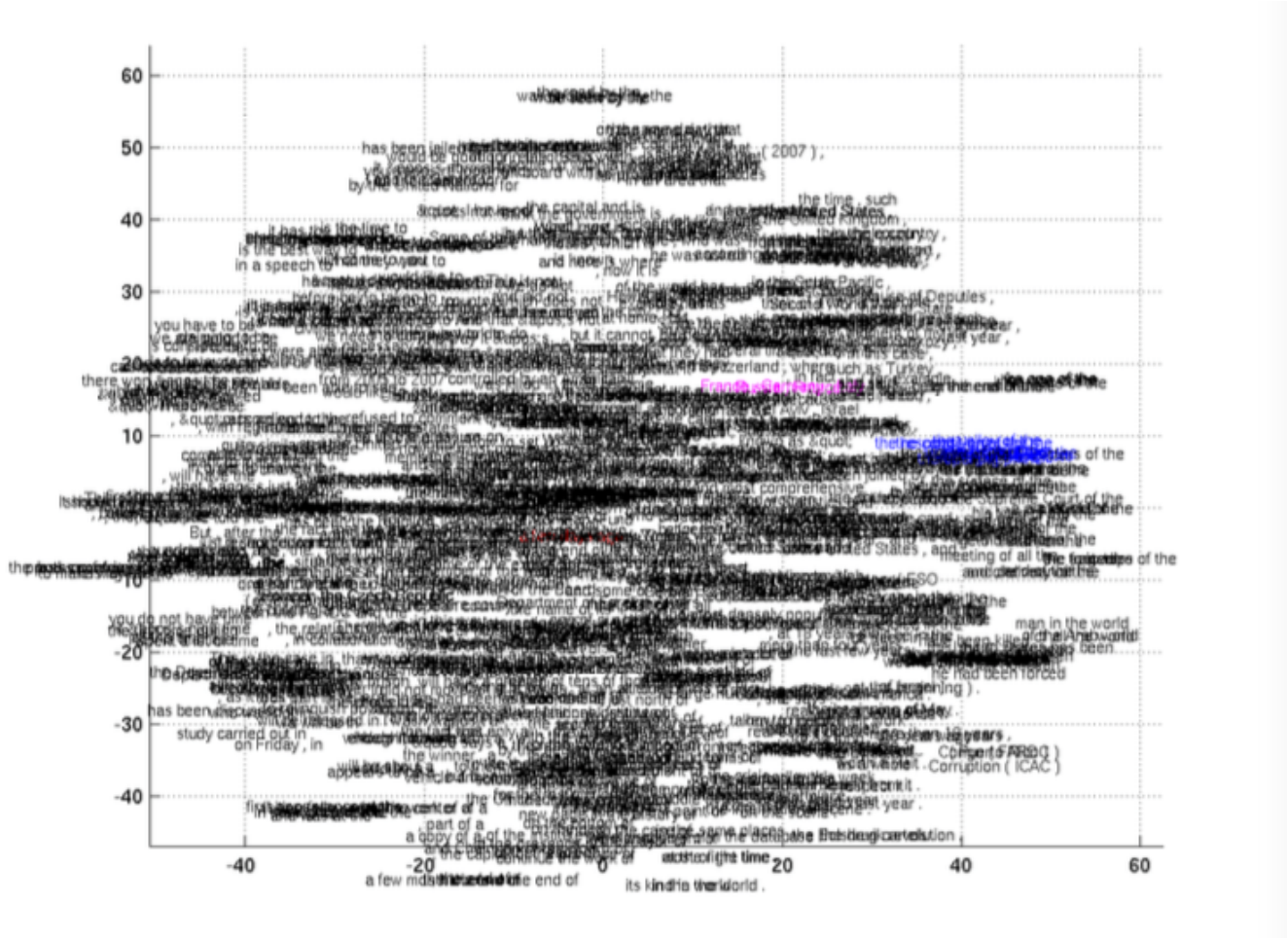
循环神经网络(RNN)
RNN Encoder-Decoder

It learns to encode a variable-length sequence into a fixed-length vector representation and to decode a given fixed-length vector representation back into a variable-length sequence.



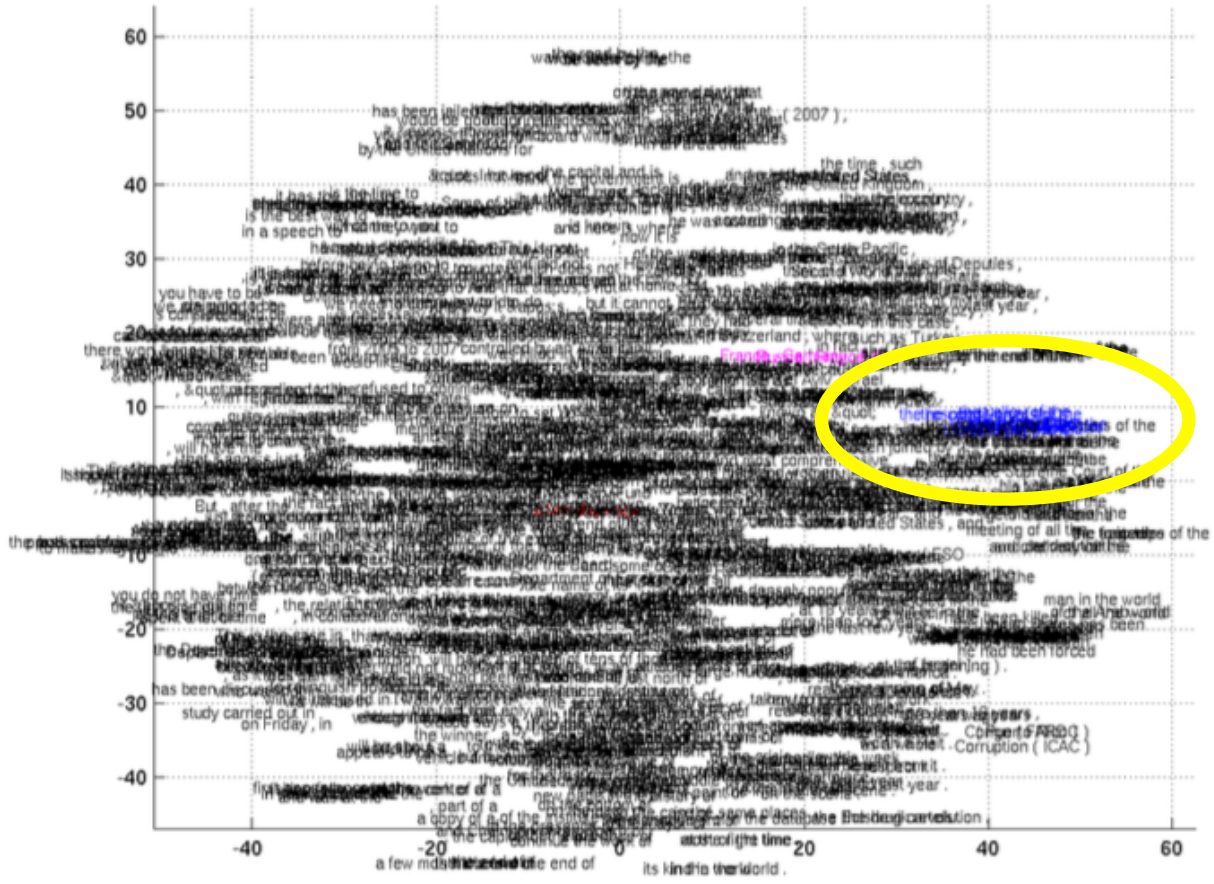
B

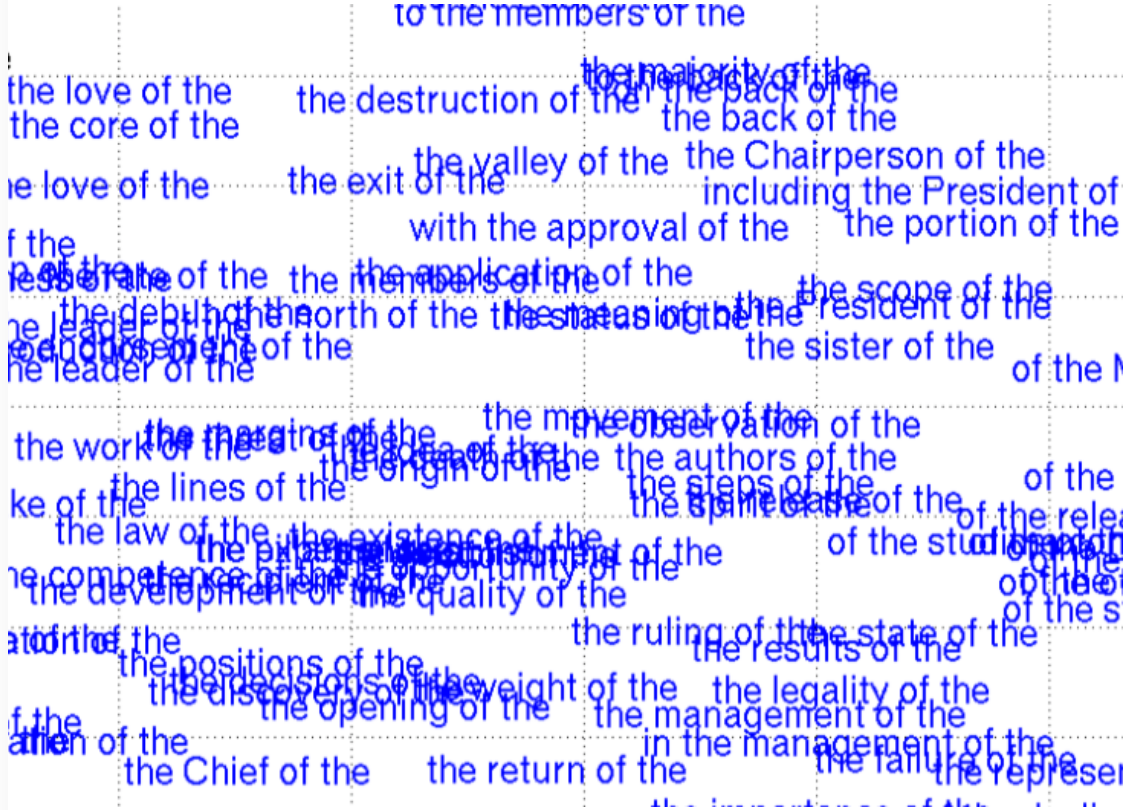
深度学习与自然语言处理



B

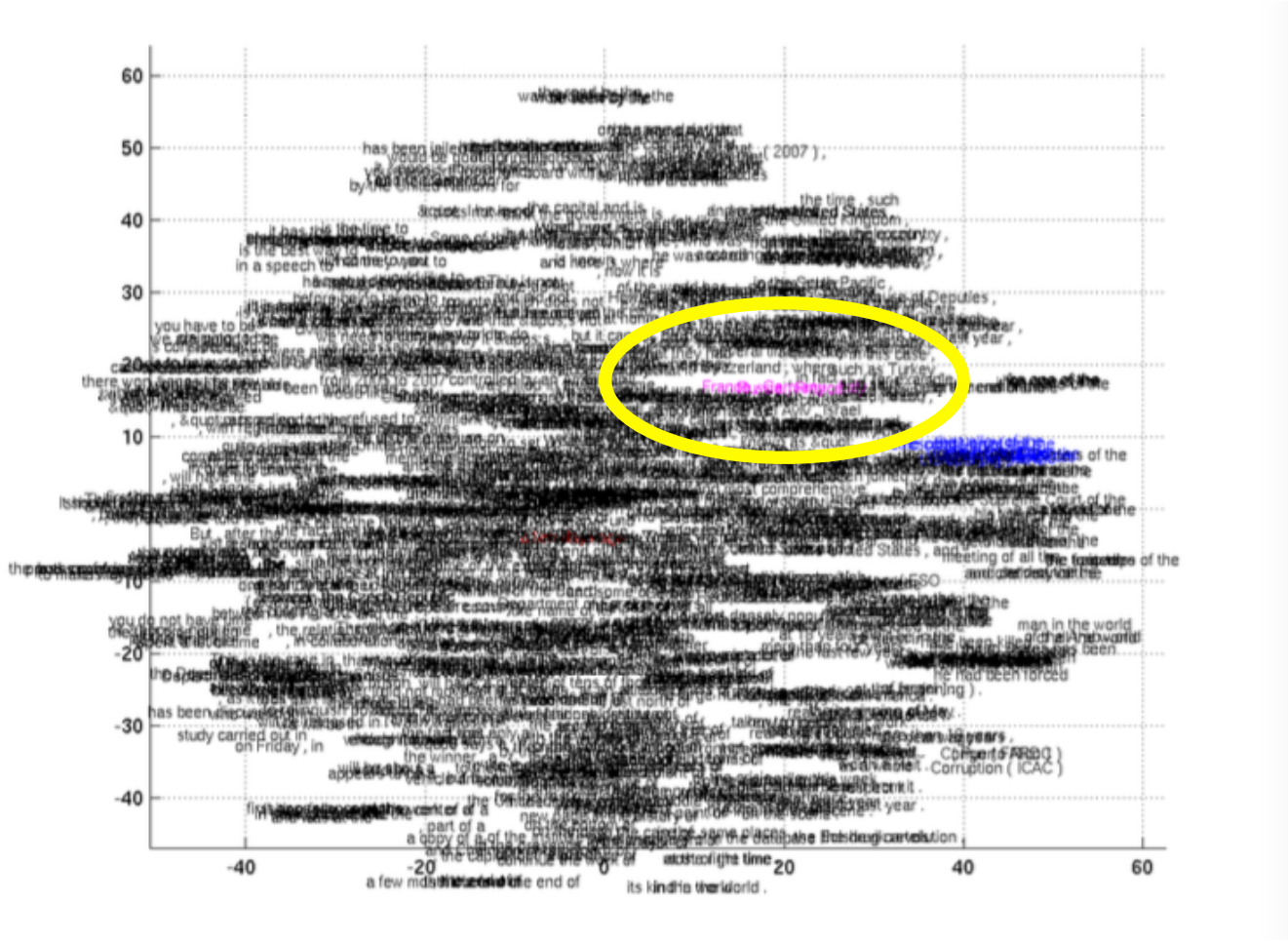
深度学习与自然语言处理

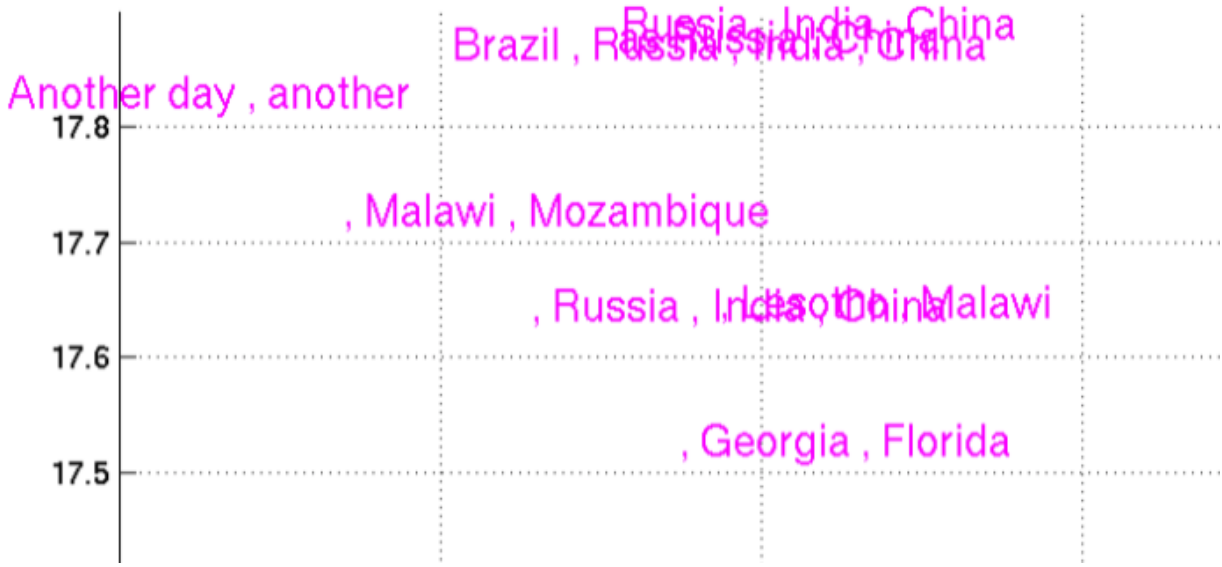




B

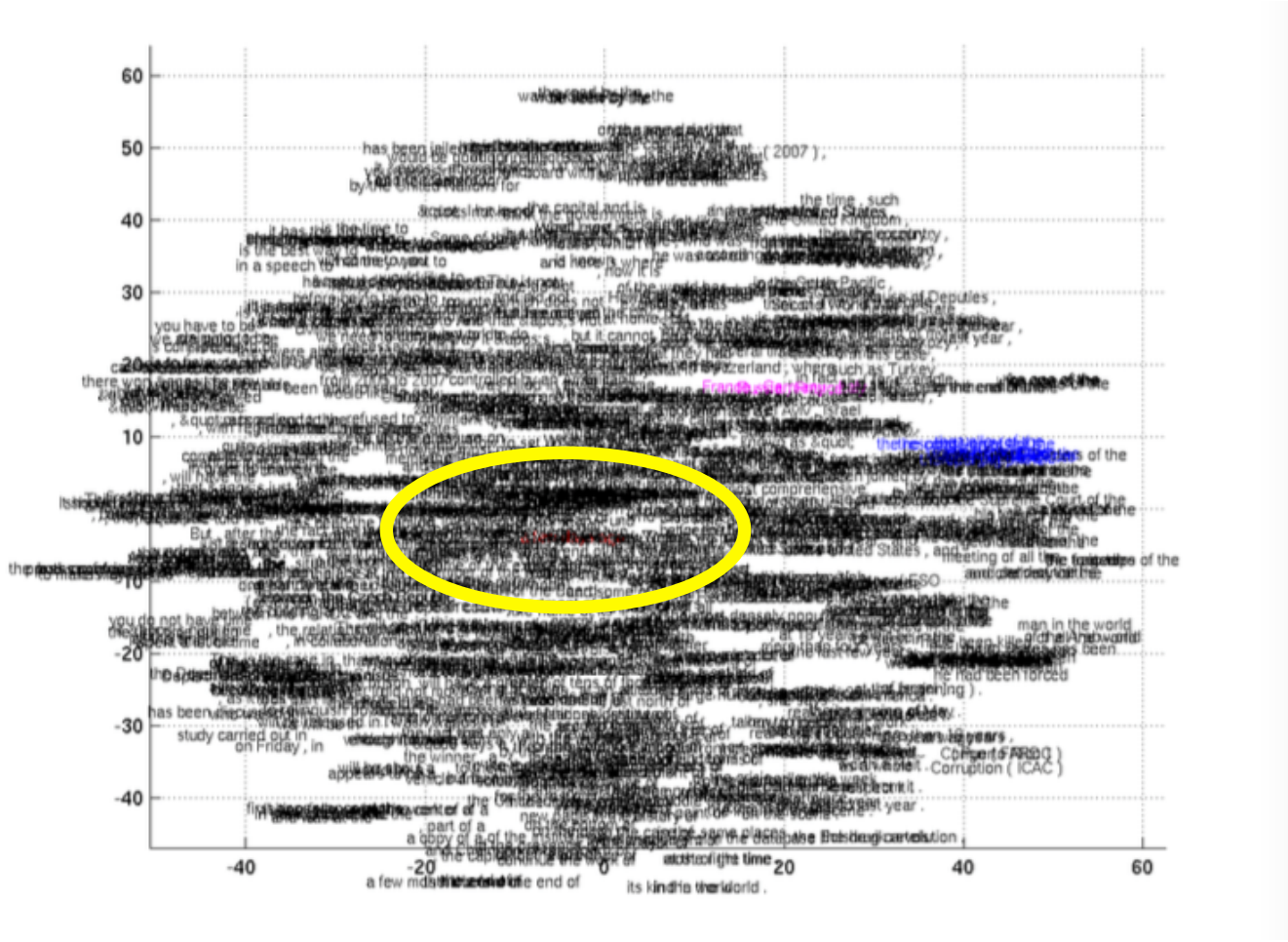
深度学习与自然语言处理





B

深度学习与自然语言处理



软件仓库挖掘

C

软件仓库挖掘

关键词搜索

优点

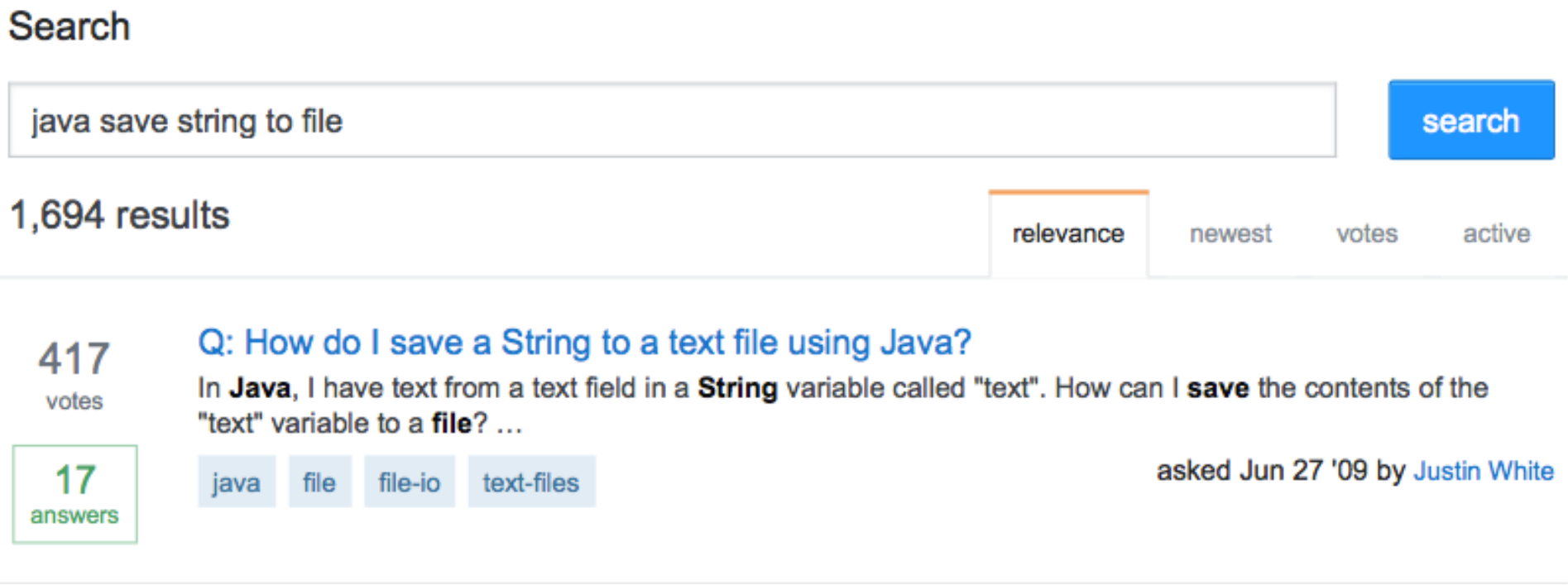
能很快找到相关内容

缺点

搜索到的内容混杂，需要甄别

C

软件仓库挖掘


Search
java save string to file
search
1,694 results
relevance
newest
votes
active
417
votes
17
answers
Q: How do I save a String to a text file using Java?
In Java, I have text from a text field in a String variable called "text". How can I save the contents of the "text" variable to a file? …
java
file
file-io
text-files
asked Jun 27 '09 by Justin White

软件仓库挖掘

Question Id 19477465
Title: Get python program to end by pressing anykey and not enter **Body:** How can I get my Python program to end by pressing any key without pressing enter. So if the user types ”c”, the program should automatically end without pressing enter. My code so far: print(”Hi everyone! This is just a quick sample code I made”) print(”Press anykey to end the program.”)

Question Id 510357 (marked as *duplicate* to 19477465)
Title: Python read a single character from the user **Body**: Is there a way of reading one single character from the user input? For instance, they press one key at the terminal and it is returned (sort of like getch()). I know there’s a function in Windows for it, but I’d like something that is cross-platform.

C

软件仓库挖掘

Search engines based on keyword matching

common places to discover APIS and their usage sequences:

Google, Bing, Baidu

Stack Overflow, GitHub

C

软件仓库挖掘

Search engines based on keyword matching

common places to discover APIS and their usage sequences:

Google, Bing, Baidu　　Stack Overflow, GitHub

Drawbacks:　Inefficient and inaccurate for programming tasks

Developers need to manually examine many web pages

Search engines based on keyword matching ignore the semantics of natural language queries.

Deep API Learning

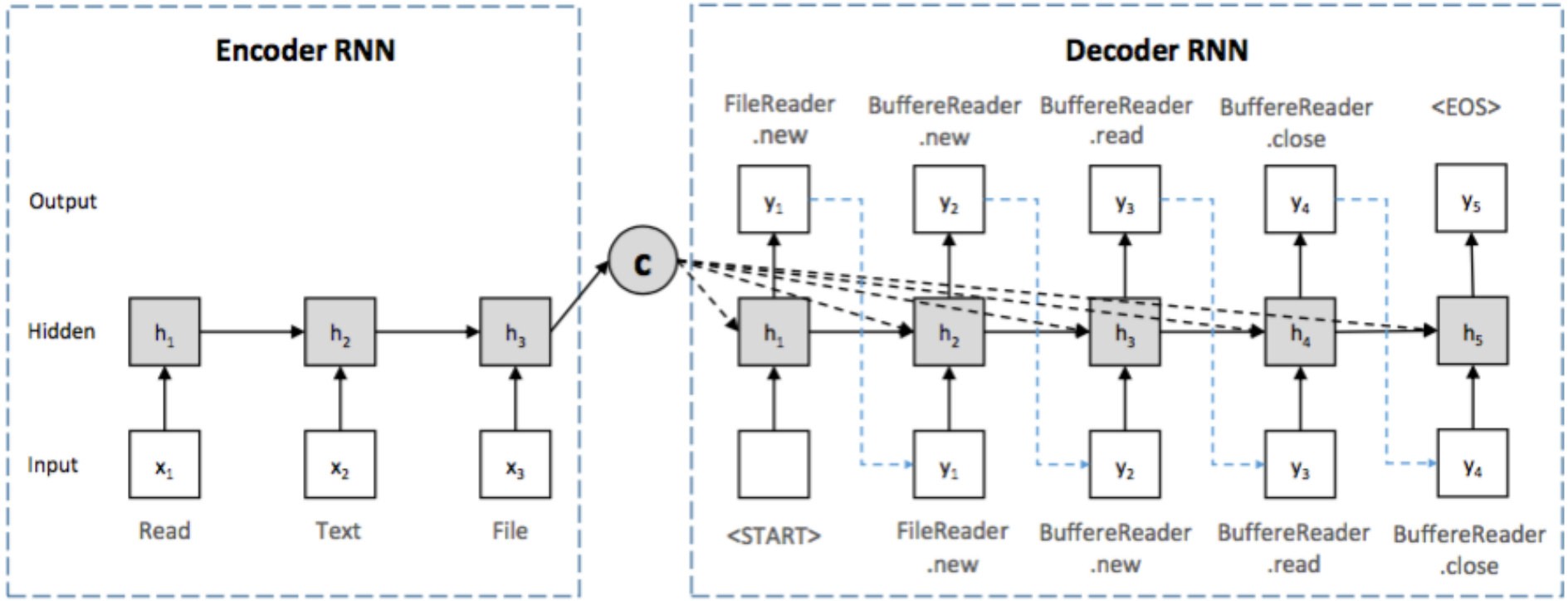
First, instead of matching keywords, DeepAPI learns the semantics of words by embedding them into a vector representation of content.

Deep API Learning

First, instead of matching keywords, DeepAPI learns the semantics of words by embedding them into a vector representation of content.

Second, DeepAPI learns the sequences of words in the natural language query and the sequence of associated APIs.

Deep API Learning



Deep API Learning

INPUT → Natural Language

OUTPUT → Natural Language

Deep API Learning

INPUT → API sequences with corresponding NL annotation

OUTPUT → API sequences

C

软件仓库挖掘

Deep API Learning　Extracting API Usage Sequences

new C()

o.m()

o1.m1(o2.m2(), o3.m3())

stmt1; stmt2; …; stmt

if(stmt1) {stmt2;} else {stmt3;}

while(stmt1) {stmt2; }

C

软件仓库挖掘

Deep API Learning	Extracting API Usage Sequences
new C()	C.new
o.m()	c.m
o1.m1(o2.m2(), o3.m3())	C2.m2-C3.m3-C1.m1
stmt1; stmt2; …; stmt	s1-s2-…-st
if(stmt1) {stmt2;} else {stmt3;}	s1-s2-s3
while(stmt1) {stmt2; }	s1-s2

软件仓库挖掘

Deep API Learning　Extracting Annotations

```
/***
 * Copies bytes from a large (over 2GB) InputStream to an OutputStream.
 * This method uses the provided buffer, so there is no need to use a
 * BufferedInputStream.
 * @param input the InputStream to read from
 *  . . .
 * @since 2.2
 */
```

C

软件仓库挖掘

Deep API Learning　Extracting Annotations

```
/***
 * Copies bytes from a large (over 2GB) InputStream to an OutputStream.
 * This method uses the provided buffer, so there is no need to use a
 * BufferedInputStream.
 * @param input the InputStream to read from
 *  . . .
 * @since 2.2
 */
```

Copies bytes from a large (over 2GB) InputStream to an OutputStream

Deep API Learning

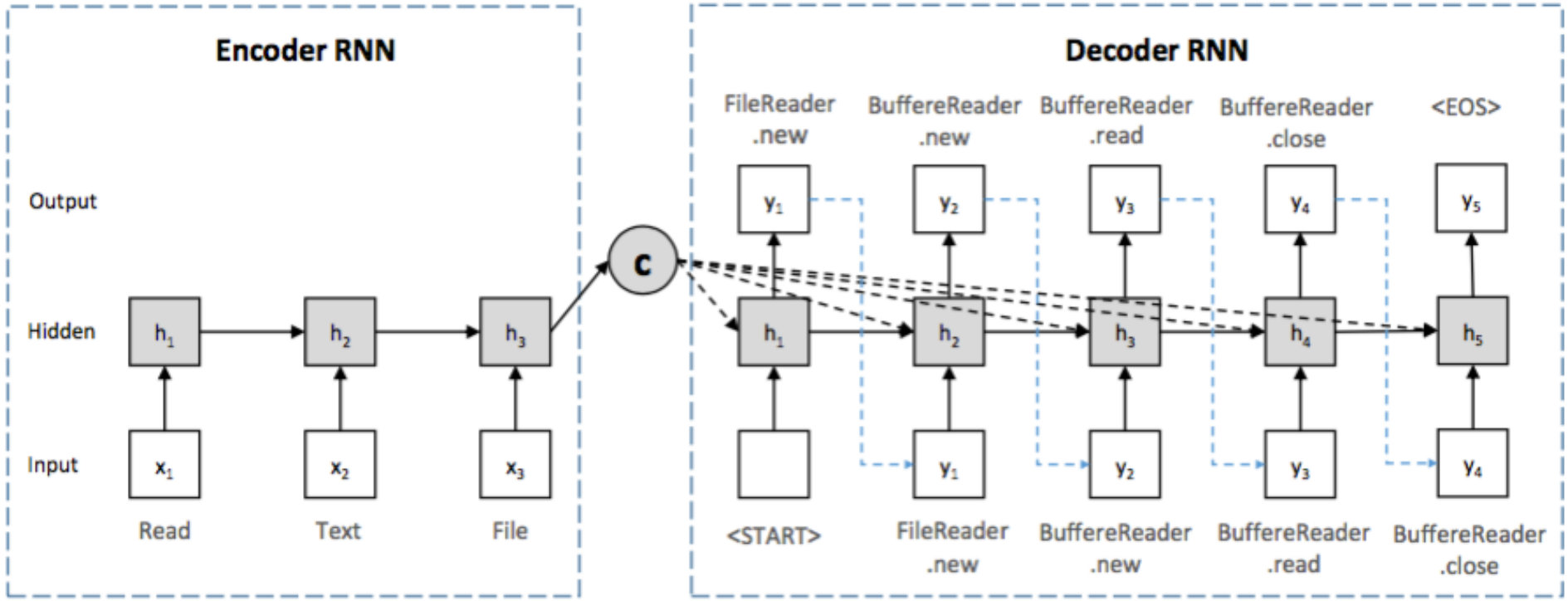
```
/***
 * Copies bytes from a large (over 2GB) InputStream to an OutputStream.
 * This method uses the provided buffer, so there is no need to use a
 * BufferedInputStream.
 * @param input the InputStream to read from
 *  . . .
 * @since 2.2
 */
public static long copyLarge(final InputStream input,
   final OutputStream  output, final byte[] buffer) throws IOException {
   long count = 0;
   int n;
   while (EOF != (n = input.read(buffer))) {
      output.write(buffer, 0, n);
      count += n;
   }
   return count;
}
```

Deep API Learning

```
/***
 * Copies bytes from a large (over 2GB) InputStream to an OutputStream.
 * This method uses the provided buffer, so there is no need to use a
 * BufferedInputStream.
 * @param input the InputStream to read from
 *  . . .
 * @since 2.2
 */
public static long copyLarge(final InputStream input,
   final OutputStream  output, final byte[] buffer) throws IOException {
   long count = 0;
   int n;
   while (EOF != (n = input.read(buffer))) {
      output.write(buffer, 0, n);
      count += n;
   }
   return count;
}
```

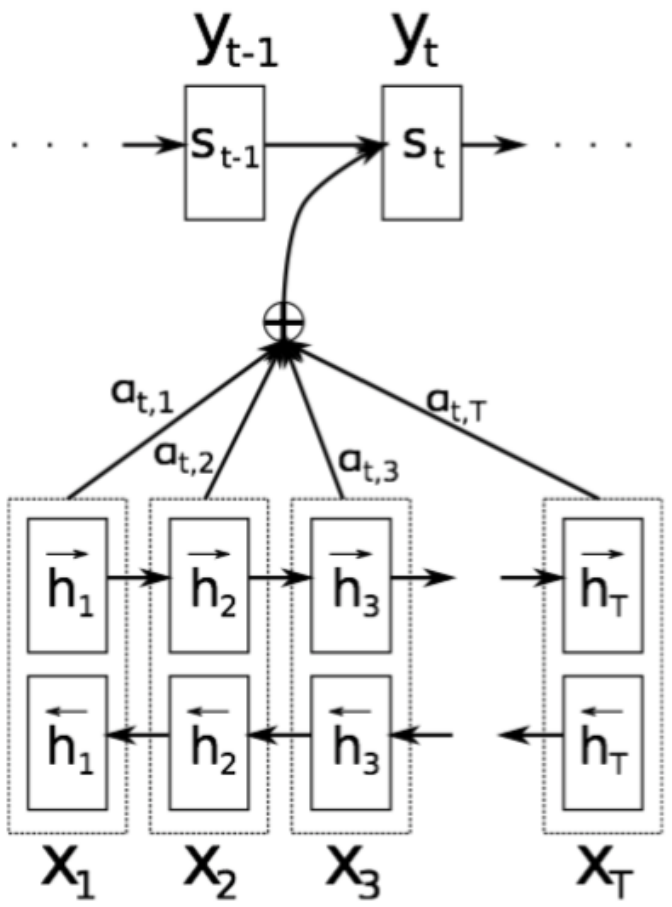
API sequence: InputStream.read ⟶ OutputStream.write
Annotation: copies bytes from a large inputstream to an outputstream.

Deep API Learning



Deep API Learning

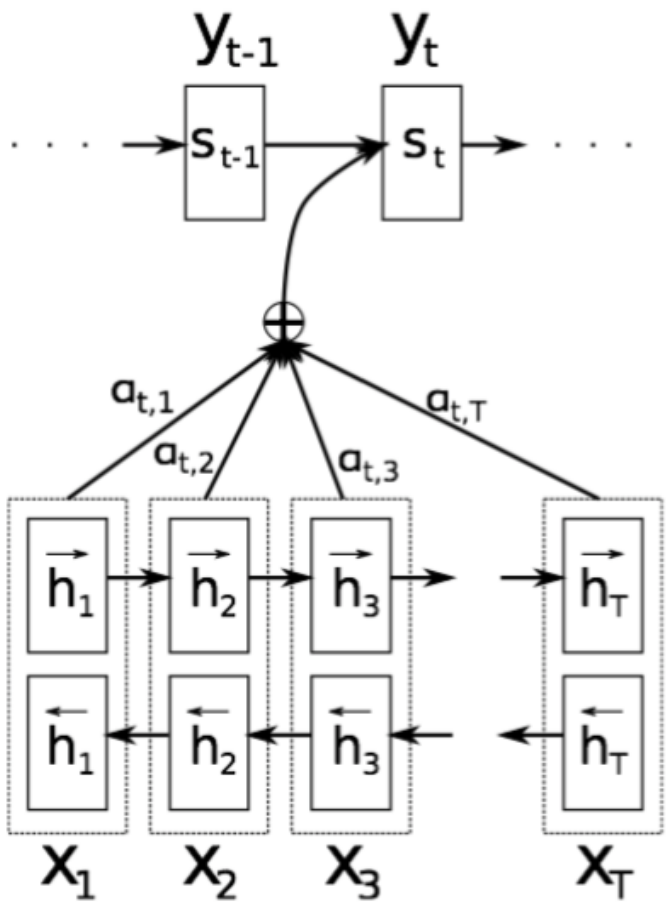
Two RNNs for encoder



Deep API Learning

Two RNNs for encoder

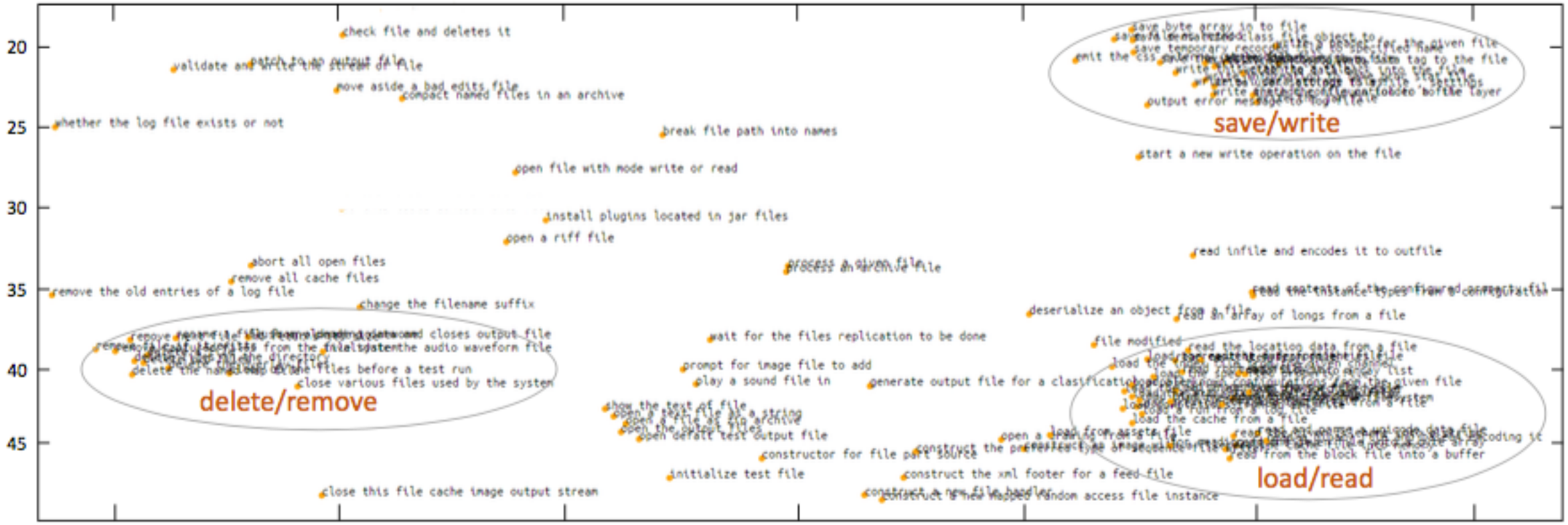
a forward RNN that directly encodes the sources sentences
a backward RNN that encodes the reversed source sentences



C

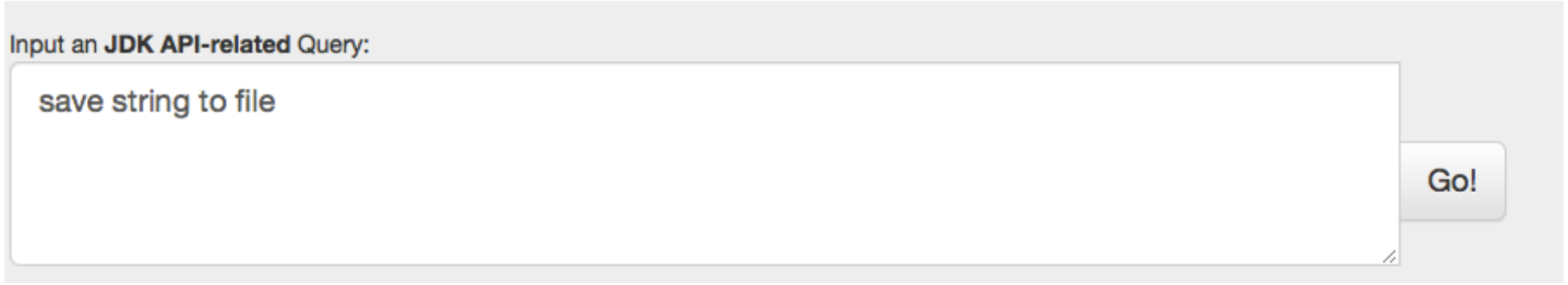
软件仓库挖掘

Deep API Learning



软件仓库挖掘

Deep API Learning



C

软件仓库挖掘

API Sequence [Note: your query may not be supported by Java SDK library]

FileWriter.new→FileWriter.write→FileWriter.close 0.0026137046515941 62

FileWriter.new→BufferedWriter.new→BufferedWriter.write→BufferedWriter.flush→BufferedWriter.close 0.10887346928939223

File.new→FileOutputStream.new→String.getBytes→FileOutputStream.write→FileOutputStream.flush→FileOutputStream.close 0.11973753806791808

String.getBytes→FileOutputStream.write→FileOutputStream.flush→FileOutputStream.getFD→FileOutputStream.flush→FileOutputStream.close 0.16190701350569725

FileWriter.new→BufferedWriter.new→BufferedWriter.write→BufferedWriter.flush→BufferedWriter.close→File.getPath 0.17773608275149999

FileWriter.new→BufferedWriter.new→BufferedWriter.write→BufferedWriter.flush→BufferedWriter.close→FileWriter.close 0.18629461017094159

File.new→FileOutputStream.new→OutputStreamWriter.new→OutputStreamWriter.write→OutputStreamWriter.close→FileOutputStream.close 0.18909082836226412

PrintWriter.new→PrintWriter.println→PrintWriter.close 0.21147412657737732

FileWriter.new→FileWriter.write→FileWriter.close→FileWriter.new→FileWriter.write→FileWriter.close 0.21948248344032387

String.getBytes→FileOutputStream.write→FileOutputStream.flush→FileOutputStream.close 0.22000371062984833

结论

自然语言处理　　语法语义 ⟶ 概率统计

结论

自然语言处理　　语法语义 ⟶ 概率统计

深度学习网络　　表征学习、RNN、Encoder-Decoder

结论

自然语言处理　　语法语义 ⟶ 概率统计

深度学习网络　　表征学习、RNN、Encoder-Decoder

软件仓库挖掘　　RNN Encoder-Decoder、API sequences

END

THANKS