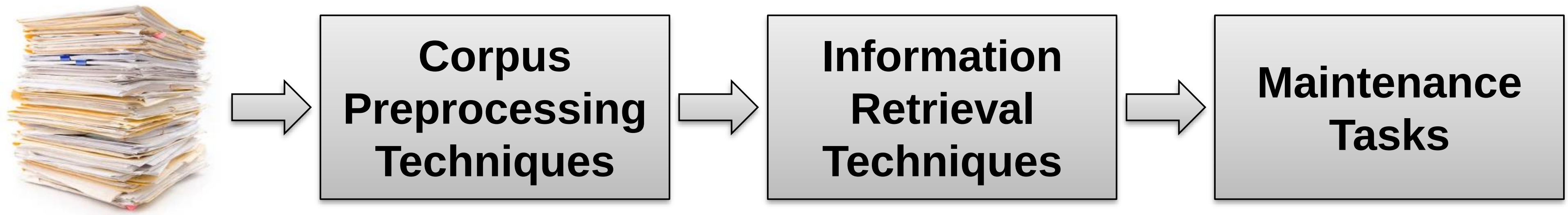
Brief Information Retrieval Tutorial

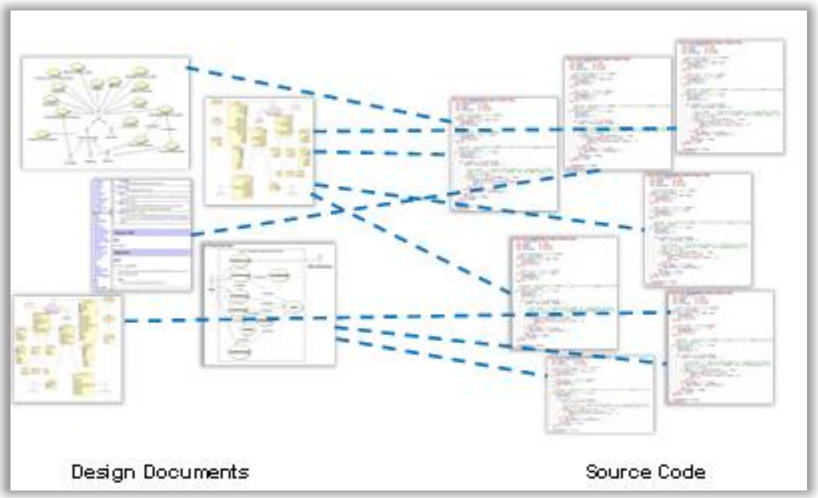
(with application in Software Engineering)

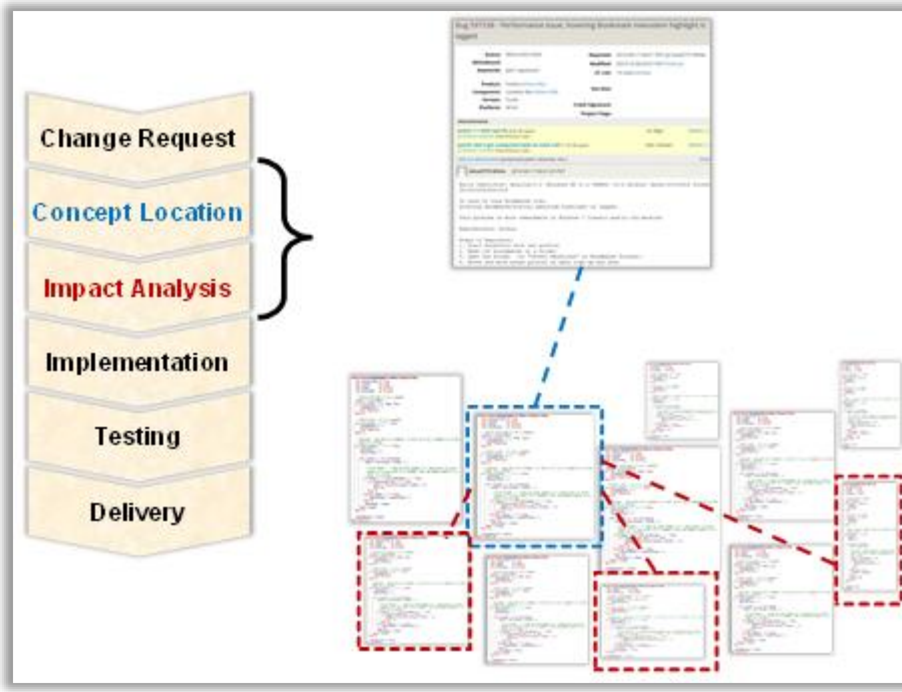

Corpus Preprocessing Techniques
Information Retrieval Techniques
Maintenance Tasks

Corpus Preprocessing Techniques → Information Retrieval Techniques → Maintenance Tasks

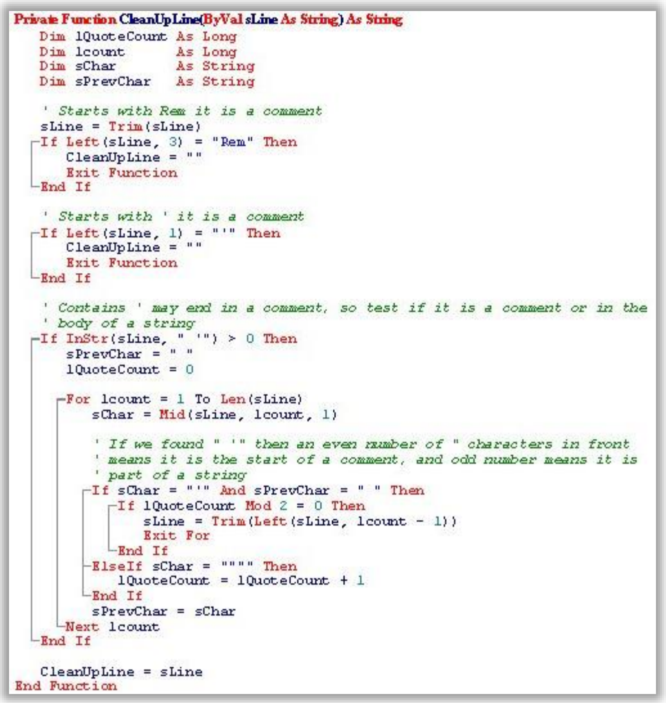
Concept Location
Impact Analysis

Traceability Link Recovery





Corpus Preprocessing Techniques → Information Retrieval Techniques → Maintenance Tasks



```vb
Private Function CleanUpLine(ByVal sLine As String) As String
    Dim lQuoteCount As Long
    Dim lcount      As Long
    Dim sChar       As String
    Dim sPrevChar   As String

    ' Starts with Rem it is a comment
    sLine = Trim(sLine)
    If Left(sLine, 3) = "Rem" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Starts with ' it is a comment
    If Left(sLine, 1) = "'" Then
        CleanUpLine = ""
        Exit Function
    End If

    ' Contains ' may end in a comment, so test if it is a comment or in the
    ' body of a string
    If InStr(sLine, " '") > 0 Then
        sPrevChar = " "
        lQuoteCount = 0

        For lcount = 1 To Len(sLine)
            sChar = Mid(sLine, lcount, 1)

            ' If we found " '" then an even number of " characters in front
            ' means it is the start of a comment, and odd number means it is
            ' part of a string
            If sChar = "'" And sPrevChar = " " Then
                If lQuoteCount Mod 2 = 0 Then
                    sLine = Trim(Left(sLine, lcount - 1))
                    Exit For
                End If
            ElseIf sChar = """" Then
                lQuoteCount = lQuoteCount + 1
            End If
            sPrevChar = sChar
        Next lcount
    End If

    CleanUpLine = sLine
End Function
```

(AST) Parser →

```java
synchronized void print(TestResult result,
            long runTime) throws IOException
{
    //standard header format
    printHeader(runTime);

    printErrors(result);
    printFailures(result);
    printFooter(result);
}

synchronized void printHeader(long runTime)
                 throws IOException
{
    …
}
```

Corpus Preprocessing Techniques → Information Retrieval Techniques → Maintenance Tasks

Remove Special Characters

```
synchronized void print(TestResult
result, long runTime) throws
IOException
{
    //standard header format
    printHeader(runTime);

    printErrors(result);
    printFailures(result);
    printFooter(result);
}

synchronized void printHeader(long
runTime) throws IOException
{
    …
}
```

```
synchronized void print TestResult
result long runTime  throws
IOException

standard header format
printHeader runTime

printErrors result
printFailures result
printFooter result


synchronized void printHeader long
runTime
throws IOException
```

Corpus Preprocessing Techniques → **Information Retrieval Techniques** → **Maintenance Tasks**

Remove Special Characters → **Split Identifiers**

```
synchronized void print TestResult
result long runTime  throws
IOException

standard header format
printHeader runTime

printErrors result
printFailures result
printFooter result


synchronized void printHeader long
runTime
throws IOException
```

→

```
synchronized void print test result
result long run time  throws
io exception

standard header format
print header run time

print errors result
print failures result
print footer result


synchronized void print header long
run time
throws io exception
```

Identifier Splitting Algorithms

Original Identifier
userId
setGID
print_file2device
SSLCertificate
MINstring
USERID
currentsize
tolocale
imitating
DEFMASKBit

Identifier Splitting Algorithms

Original Identifier	Camel Case
userId	user Id
setGID	set GID
print_file2device	print file 2 device
SSLCertificate	SSL Certificate
MINstring	MI Nstring
USERID	USERID
currentsize	currentsize
tolocale	tolocale
imitating	imitating
DEFMASKBit	DEFMASK Bit

Identifier Splitting Algorithms

Original Identifier	Camel Case
userId	user Id
setGID	set GID
print_file2device	print file 2 device
SSLCertificate	SSL Certificate
MINstring	MI Nstring
USERID	USERID
currentsize	currentsize
tolocale	tolocale
imitating	imitating
DEFMASKBit	DEFMASK Bit

Handles underscore and digits

Fails at mixed cases

Fails at same case identifiers

Identifier Splitting Algorithms

Original Identifier	Camel Case	Samurai*
userId	user Id	user Id
setGID	set GID	set GID
print_file2device	print file 2 device	print file 2 device
SSLCertificate	SSL Certificate	SSL Certificate
MINstring	MI Nstring	MIN string
USERID	USERID	USER ID
currentsize	currentsize	current size
tolocale	tolocale	tol ocal e
imitating	imitating	imi ta ting
DEFMASKBit	DEFMASK Bit	DEF MASK Bit

*[Enslen'09]

Identifier Splitting Algorithms

Original Identifier	Camel Case	Samurai*
userId	user Id	user Id
setGID	set GID	set GID
print_file2device	print file 2 device	print file 2 d
SSLCertificate	SSL Certificate	SSL Certificate
MINstring	MI Nstring	MIN string
USERID	USERID	USER ID
currentsize	currentsize	current size
tolocale	tolocale	tol ocal e
imitating	imitating	imi ta ting
DEFMASKBit	DEFMASK Bit	DEF MASK Bit

Splits some cases that CamelCase cannot

Oversplits

*[Enslen'09]

Identifier Splitting Algorithms

Original Identifier	Camel Case	Samurai*
userId	user Id	user Id
setGID	set GID	set GID

Splits some cases that CamelCase

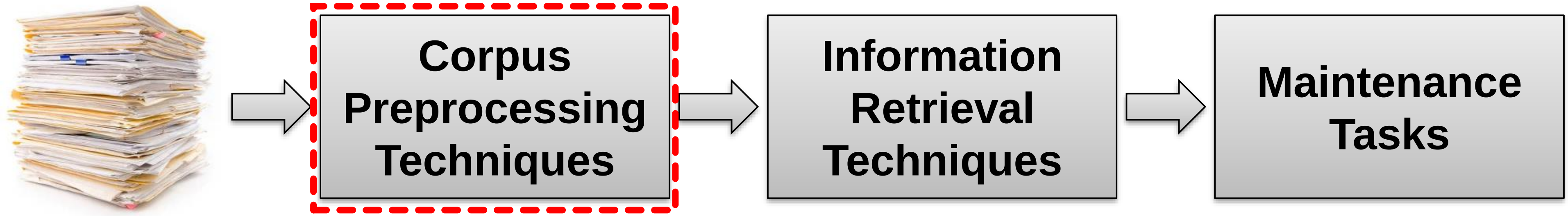
Choose meaningful identifier names

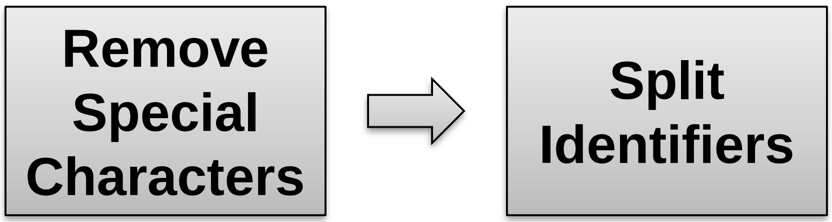
- Improves code readability

Use consistent naming conventions

- promotes shared code ownership (XP)
- Improves search results

*[Enslen'09]






```
synchronized void print TestResult
result long runTime  throws
IOException

standard header format
printHeader runTime

printErrors result
printFailures result
printFooter result


synchronized void printHeader long
runTime
throws IOException
```

```
synchronized void print test result
result long run time  throws
io exception

standard header format
print header run time

print errors result
print failures result
print footer result


synchronized void print header long
run time
throws io exception
```

Corpus Preprocessing Techniques → Information Retrieval Techniques → Maintenance Tasks

Remove Special Characters → Split Identifiers → Remove Stop Words

```
synchronized void print test result
result long run time  throws
io exception

standard header format
print header run time

print errors result
print failures result
print footer result


synchronized void print header long
run time
throws io exception
```

```
print test result
result run time
io exception

standard header format
print header run time

print errors result
print failures result
print footer result


print header
run time
io exception
```

Corpus Preprocessing Techniques → **Information Retrieval Techniques** → **Maintenance Tasks**

Remove Special Characters → **Split Identifiers** → **Remove Stop Words** → **Stem**

```
print test result
result run time
io exception

standard header format
print header run time

print errors result
print failures result
print footer result


print header
run time
io exception
```

→

```
print test result
result run time
io exception

standard head format
print head run time

print error result
print fail result
print foot result


print head
run time
io exception
```

Stemming Errors

- If stemmer is too aggressive
  - organization ⇒ organ
  - police ⇒ policy
  - army ⇒ arm
  - executive ⇒ execute

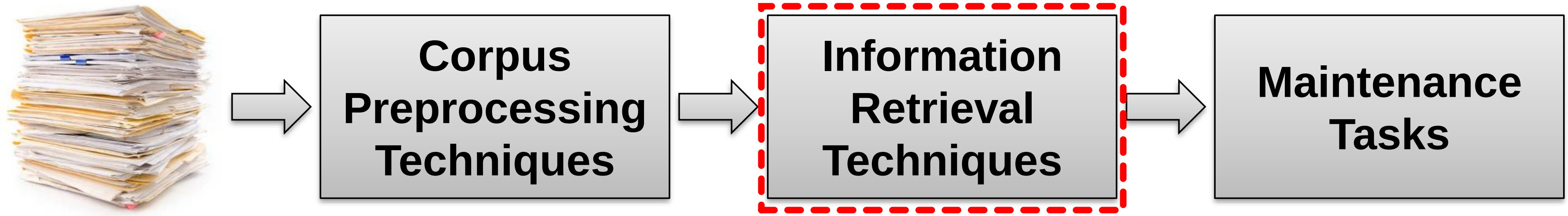
Corpus Preprocessing Techniques → Information Retrieval Techniques → Maintenance Tasks

Summary of Preprocessing (Extract only the "meaningful" words from the documents in the corpus)

```
synchronized void print(TestResult result,
          long runTime) throws IOException
{
    //standard header format
    printHeader(runTime);

    printErrors(result);
    printFailures(result);
    printFooter(result);
}

synchronized void printHeader(long runTime)
               throws IOException
{
```

→

```
print test result
result run time
io exception

standard head format
print head run time

print error result
print fail result
print foot result


print head
run time
io exception
```



m_1: print test result
result run time
io exception

standard head format
print head run time

print error result
print fail result
print foot result

m_2: print head
run time
io exception

Corpus Preprocessing Techniques → **Information Retrieval Techniques** → Maintenance Tasks

Term-by-Document Matrix

m_1:
```
print test result
result run time
io exception

standard head format
print head run time

print error result
print fail result
print foot result
```

m_2:
```
print head
run time
io exception
```

Corpus Preprocessing Techniques → **Information Retrieval Techniques** → **Maintenance Tasks**

Term-by-Document Matrix

m_1:
```
print test result
result run time
io exception

standard head format
print head run time

print error result
print fail result
print foot result
```

m_2:
```
print head
run time
io exception
```

Unique Terms

Methods

	print	test	result	time	…
m_1	5	1	5	2	
m_2	1	0	0	1	
…	…	…	…	…	
m_n	…	…	…	…	

Each document is represented as a vector of terms

Corpus Preprocessing Techniques → Information Retrieval Techniques → Maintenance Tasks

Term-by-Document Matrix

	print	test	result	...
m_1	**5**	**1**	**3**	...
m_2	**1**	**0**	**0**	...

Corpus Preprocessing Techniques

Information Retrieval Techniques

Maintenance Tasks

Term-by-Document Matrix

Term Weighting

	print	test	result	...
m_1	**5**	**1**	**3**	...
m_2	**1**	**0**	**0**	...

boolean
tf
tf-idf
log(tf+1)
tf-entropy

Corpus Preprocessing Techniques → Information Retrieval Techniques → Maintenance Tasks

Term-by-Document Matrix → Term Weighting → Weighted Term-by-Document Matrix

	print	test	result	...
m_1	**5**	**1**	**3**	...
m_2	**1**	**0**	**0**	...

boolean
tf
tf-idf
log(tf+1)
tf-entropy

	print	test	result	...
m_1	**0.78**	**0.2**	**0.9**	...
m_2	**0.45**	**0**	**0**	...

Corpus Preprocessing Techniques → **Information Retrieval Techniques** → Maintenance Tasks

Term-by-Document Matrix → Term Weighting → Weighted Term-by-Document Matrix

	print	test	result	...
m_1	**5**	**1**	**3**	...

boolean
tf
tf-idf
log(tf+1)

	print	test	result	...
m_1	**0.78**	**0.2**	**0.9**	...

Each term in the document vector is "weighted" based on its "importance" to the document (method) and corpus (collection of documents).

Why?

Corpus Preprocessing Techniques → Information Retrieval Techniques → Maintenance Tasks

Term-by-Document Matrix → Term Weighting → Weighted Term-by-Document Matrix

	print	test	result	...
m_1	**5**	**1**	**3**	...

boolean
tf
tf-idf
log(tf+1)

	print	test	result	...
m_1	**0.78**	**0.2**	**0.9**	...

Each term in the document vector is "weighted" based on its "importance" to the document (method) and corpus (collection of documents).

In different contexts, some terms are more important than others

Corpus Preprocessing Techniques → **Information Retrieval Techniques** → Maintenance Tasks

Term-by-Document Matrix → Term Weighting → Weighted Term-by-Document Matrix

	print	test	result	...
m_1	**5**	**1**	**3**	...

boolean
tf
tf-idf
log(tf+1)

	print	test	result	...
m_1	**0.78**	**0.2**	**0.9**	...

Each term in the document vector is "weighted" based on its "importance" to the document (method) and corpus (collection of documents).

In different contexts, some terms are more important than others

Term Weights

Term Frequency (TF)

+

Inverse Document Frequency (IDF)

=

TF-IDF

Assumption for Using Term Weights...

- Terms that appear in many *different* documents are …

Assumption for Using Term Weights…

- Terms that appear in many *different* documents are
  - less indicative of the overall topic
  - carry less “meaning”

- Term weights represent an indication of a term’s *discriminative* power

Inverse Document Frequency (IDF) Example

Doc_1

Doc_2

Doc_3

…

…

…

…

…

…

…

…

…

…

…

…

…

… … … … … … … … … … … … … … … … … …

…

Doc_{10K}

- In a corpus of 10,000 documents

Doc_1	a a a
Doc_2	a
Doc_3	a a
…	a a
…	a a a
…	a
…	a
…	a a
…	a a a
…	a a a
…	a
…	a a
…	a a
…	a a a
…	a
…	a a a
… … … … … …	… … … … … … … … … … … …
…	a
Doc_{10K}	a

10,000 documents

- In a corpus of 10,000 documents
- Term a:
  - Appears in 10,000 documents (it can appear multiple times in the same document)
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10,000}{10,000}\right) = 0$$

Doc_1	a a a
Doc_2	a
Doc_3	a a
…	a a
…	a a a
…	a
…	a
…	a a
…	a a a
…	a a a
…	a
…	a a
…	a a
…	a a a
…	a
…	a a a
… … … … … … … … … … … … … … … … … …	
…	a
Doc_{10K}	a

- In a corpus of 10,000 documents

- Term a:
  - Appears in 10,000 documents (it can appear multiple times in the same document)
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10,000}{10,000}\right) = 0$$

- Term b (appears in 5,000 documents)

Doc_1	a a a	b b
Doc_2	a	
Doc_3	a a	b
…	a a	
…	a a a	b b b b
…	a	
…	a	b b
…	a a	
…	a a a	b
…	a a a	
…	a	b b b
…	a a	
…	a a	b b
…	a a a	
…	a	b b b b
…	a a a	
… … … … … … …	… … … …	… … … … … … …
…	a	b b
Doc_{10K}	a	

- In a corpus of 10,000 documents
- Term <u>a</u>:
  - Appears in 10,000 documents (it can appear multiple times in the same document)
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10,000}{10,000}\right) = 0$$

- Term <u>b</u> (appears in 5,000 documents)

Doc_1	a a a	b b
Doc_2	a	
Doc_3	a a	b
…	a a	
…	a a a	b b b b
…	a	
…	a	b b
…	a a	
…	a a a	b
…	a a a	
…	a	b b b
…	a a	
…	a a	b b
…	a a a	
…	a	b b b b
…	a a a	
… … … … … … …	… … … …	… … … … … … …
…	a	b b
Doc_{10K}	a	

- In a corpus of 10,000 documents
- Term <u>a</u>:
  - Appears in 10,000 documents (it can appear multiple times in the same document)
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10,000}{10,000}\right) = 0$$

- Term <u>b</u> (appears in 5,000 documents)

$$idf_b = log\left(\frac{10,000}{5,000}\right) = 0.69$$

Doc_1	a a a	b b
Doc_2	a	
Doc_3	a a	b
…	a a	
…	a a a	b b b b
…	a	
…	a	b b
…	a a	
…	a a a	b
…	a a a	
…	a	b b b
…	a a	
…	a a	b b
…	a a a	
…	a	b b b b
…	a a a	
… … …	… … …	… … …
…	a	b b
Doc_{10K}	a	

- In a corpus of 10,000 documents
- Term a:
  - Appears in 10,000 documents (it can appear multiple times in the same document)
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10{,}000}{10{,}000}\right) = 0$$

- Term b (appears in 5,000 documents)

$$idf_b = log\left(\frac{10{,}000}{5{,}000}\right) = 0.69$$

- Term c (appears in 20 documents)

Doc_1	a a a	b b	c c
Doc_2	a		
Doc_3	a a	b	c
…	a a		
…	a a a	b b b b	
…	a		
…	a	b b	c c c
…	a a		
…	a a a	b	
…	a a a		
…	a	b b b	
…	a a		c c c c
…	a a	b b	
…	a a a		
…	a	b b b b	
…	a a a		
… … … … … …	… … … … … …	… … … …	… … … …
…	a	b b	
Doc_{10K}	a		c

- In a corpus of 10,000 documents
- Term a:
  - Appears in 10,000 documents (it can appear multiple times in the same document)
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10{,}000}{10{,}000}\right) = 0$$

- Term b (appears in 5,000 documents)

$$idf_b = log\left(\frac{10{,}000}{5{,}000}\right) = 0.69$$

- Term c (appears in 20 documents)

Doc_1	a a a	b b	c c
Doc_2	a		
Doc_3	a a	b	c
…	a a		
…	a a a	b b b b	
…	a		
…	a	b b	c c c
…	a a		
…	a a a	b	
…	a a a		
…	a	b b b	
…	a a		c c c c
…	a a	b b	
…	a a a		
…	a	b b b b	
…	a a a		
… … …	… … … … …	… … … …	… … … … …
…	a	b b	
Doc_{10K}	a		c

- In a corpus of 10,000 documents
- Term <u>a</u>:
  - Appears in 10,000 documents (it can appear multiple times in the same document)
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10{,}000}{10{,}000}\right) = 0$$

- Term <u>b</u> (appears in 5,000 documents)

$$idf_b = log\left(\frac{10{,}000}{5{,}000}\right) = 0.69$$

- Term <u>c</u> (appears in 20 documents)

$$idf_c = log\left(\frac{10{,}000}{20}\right) = 6.21$$

Doc_1	a a a	b b	c c
Doc_2	a		
Doc_3	a a	b	c
…	a a		
…	a a a	b b b b	
…	a		
…	a	b b	c c c
…	a a		
…	a a a	b	
…	a a a		
…	a	b b b	
…	a a		c c c c
…	a a	b b	
…	a a a		
…	a	b b b b	
…	a a a		
… … … … … … … … … … … … … … … … … …			
…	a	b b	
Doc_{10K}	a		c

- In a corpus of 10,000 documents

- Term a:
  - Appears in 10,000 documents (it can appear multiple times in the same document)
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10,000}{10,000}\right) = 0$$

- Term b (appears in 5,000 documents)

$$idf_b = log\left(\frac{10,000}{5,000}\right) = 0.69$$

- Term c (appears in 20 documents)

$$idf_c = log\left(\frac{10,000}{20}\right) = 6.21$$

- Term d (appears in 1 document)

Doc_1	a a a	b b	c c	
Doc_2	a			d d d
Doc_3	a a	b	c	
...	a a			
...	a a a	b b b b		
...	a			
...	a	b b	c c c	
...	a a			
...	a a a	b		
...	a a a			
...	a	b b b		
...	a a		c c c c	
...	a a	b b		
...	a a a			
...	a	b b b b		
...	a a a			
... ...				
...	a	b b		
Doc_{10K}	a		c	

- In a corpus of 10,000 documents
- Term <u>a</u>:
  - Appears in 10,000 documents (it can appear multiple times in the same document)
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10{,}000}{10{,}000}\right) = 0$$

- Term <u>b</u> (appears in 5,000 documents)

$$idf_b = log\left(\frac{10{,}000}{5{,}000}\right) = 0.69$$

- Term <u>c</u> (appears in 20 documents)

$$idf_c = log\left(\frac{10{,}000}{20}\right) = 6.21$$

- Term <u>d</u> (appears in 1 document)

Doc_1	a a a	b b	c c	
Doc_2	a			d d d
Doc_3	a a	b	c	
...	a a			
...	a a a	b b b b		
...	a			
...	a	b b	c c c	
...	a a			
...	a a a	b		
...	a a a			
...	a	b b b		
...	a a		c c c c	
...	a a	b b		
...	a a a			
...	a	b b b b		
...	a a a			
... ...	... ...	... ...	... ...	... ...
...	a	b b		
Doc_{10K}	a		c	

- In a corpus of 10,000 documents
- Term a:
  - Appears in 10,000 documents (it can appear multiple times in the same document)
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10,000}{10,000}\right) = 0$$

- Term b (appears in 5,000 documents)

$$idf_b = log\left(\frac{10,000}{5,000}\right) = 0.69$$

- Term c (appears in 20 documents)

$$idf_c = log\left(\frac{10,000}{20}\right) = 6.21$$

- Term d (appears in 1 document)

$$idf_d = log\left(\frac{10,000}{1}\right) = 9.21$$

Doc_1	a a a	b b	c c	
Doc_2	a			d d d
Doc_3	a a	b	c	
...	a a			
...	a a a	b b b b		
...	a			
...	a	b b	c c c	
...	a a			
...	a a a	b		
...				
...				
...				
...				
...				
...				
...				

... ...

...	a	b b		
Doc_{10K}	a		c	

Inverse Document Frequency (IDF) provides:

- low values for common words and
- high values for rare words (with discriminative power)

- In a corpus of 10,000 documents
- Term a:
  - Appears in 10,000 documents (it can appear multiple times in the same document)
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10{,}000}{10{,}000}\right) = 0$$

- Term d (appears in 1 document)

$$idf_d = log\left(\frac{10{,}000}{1}\right) = 9.21$$

Computing TF-IDF: An Example

Doc_1

Doc_2

Doc_3

...

...

...

...

...

...

...

...

...

...

...

...

...

... ...

...

Doc_{10K}

- In a corpus of 10,000 documents

Doc_1 a a a b b c

Doc_2

Doc_3

...

...

...

...

...

...

...

...

...

...

...

...

...

... ...

...

Doc_{10K}

- In a corpus of 10,000 documents

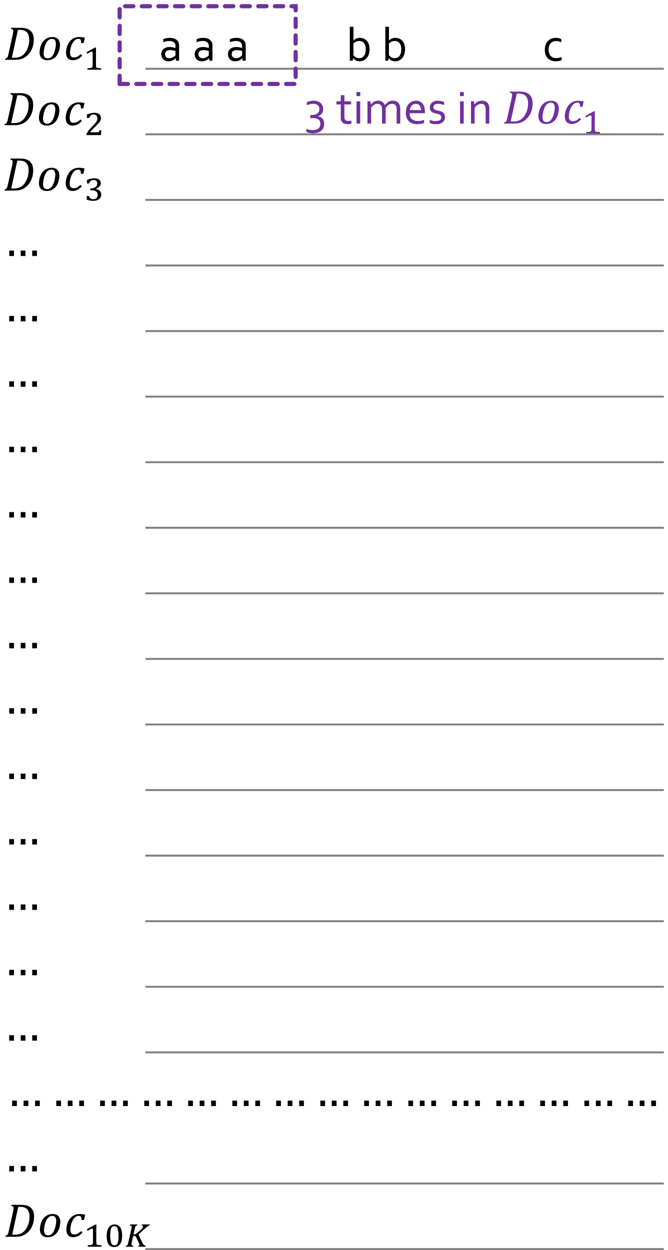
Doc_1 has only 6 terms…

Doc_1	a a a	b b	c
Doc_2			
Doc_3			
...			
...			
...			
...			
...			
...			
...			
...			
...			
...			
...			
...			
...			

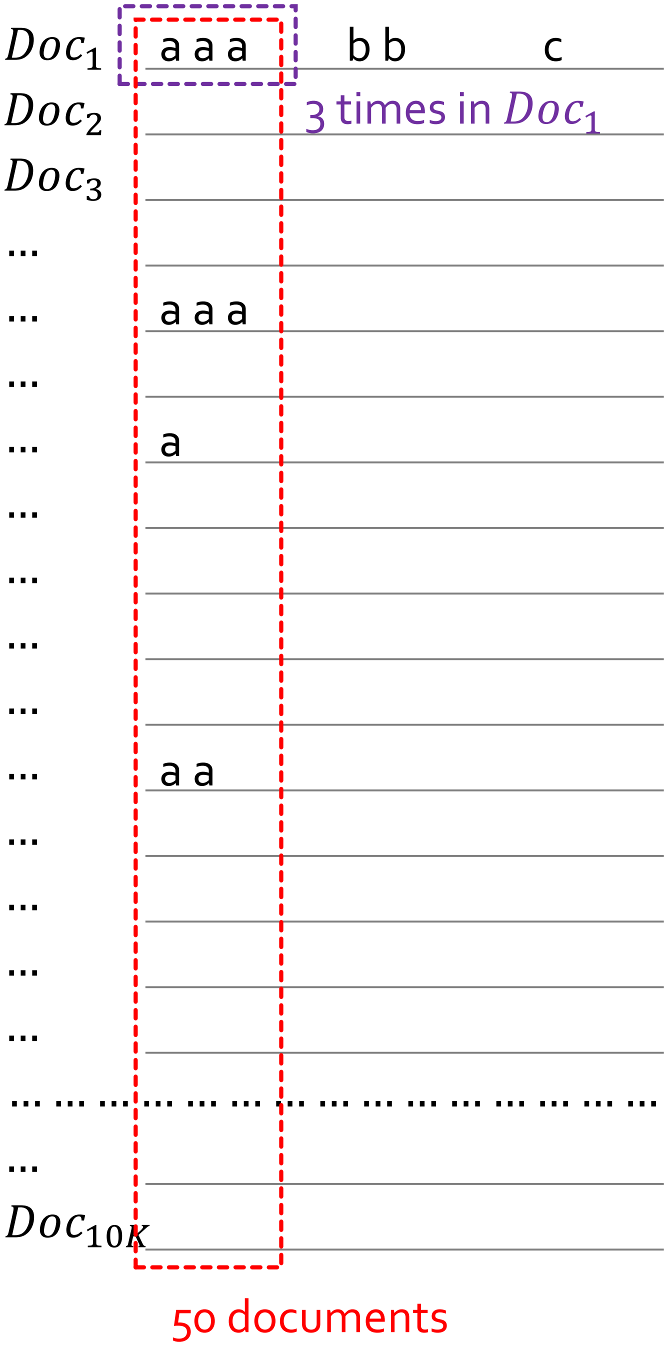
... ...

...			
Doc_{10K}			

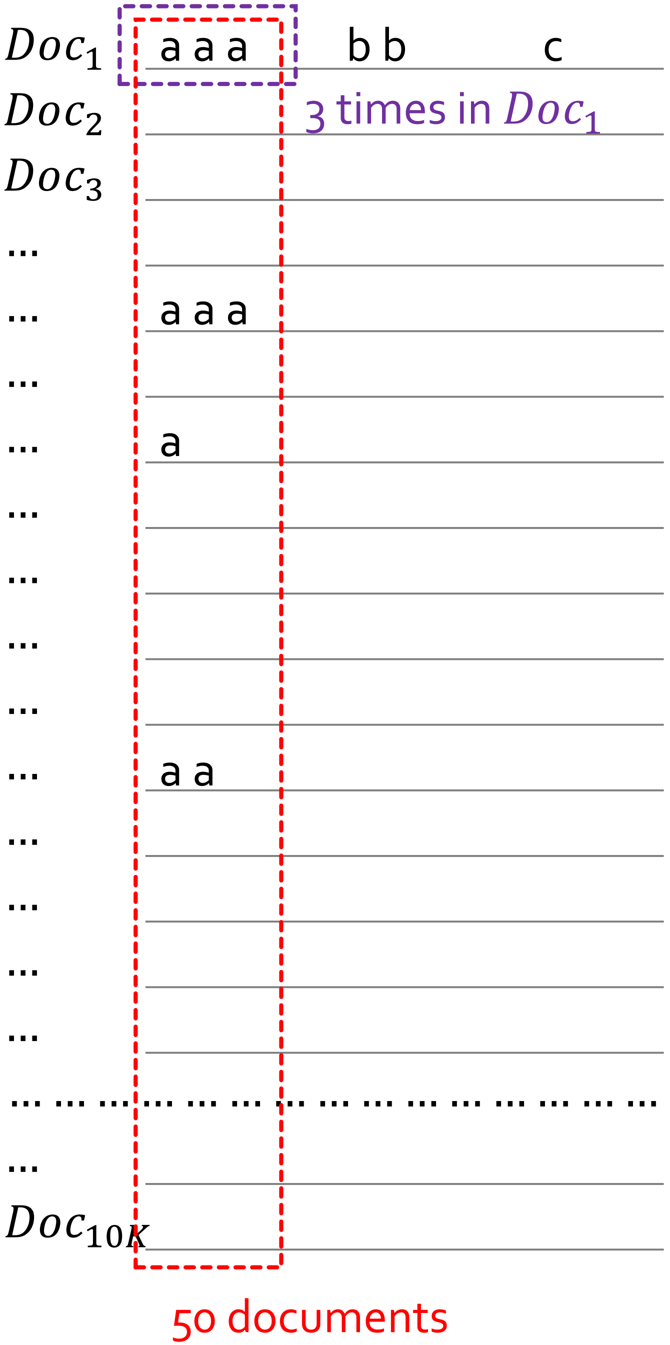
- In a corpus of 10,000 documents
- Term $\underline{a}$



- In a corpus of 10,000 documents
- Term $\underline{a}$
  - Appears 3 times in Doc_1



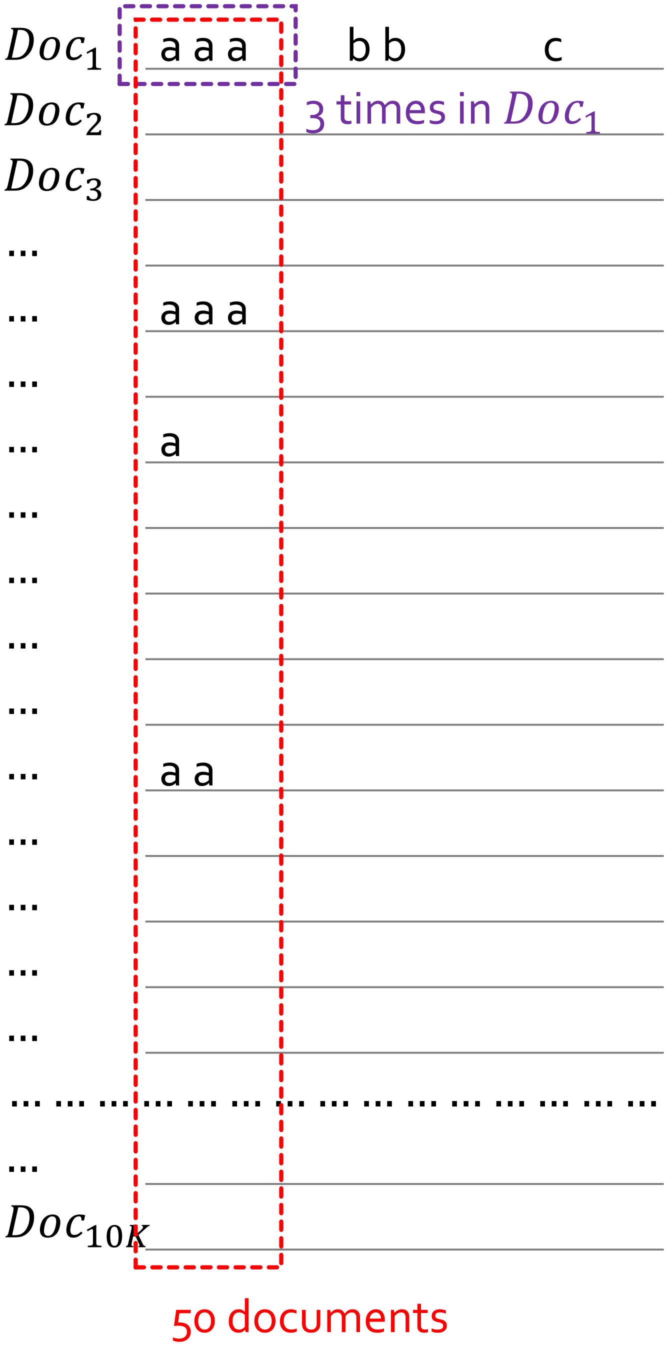
- In a corpus of 10,000 documents
- Term $\underline{a}$
  - Appears 3 times in Doc_1
  - Appears in 50 documents



- In a corpus of 10,000 documents
- Term $\underline{a}$
  - Appears 3 times in Doc_1
  - Its normalized term frequency (TF) in Doc_1 is:

$$tf_{Doc_1,a} = \frac{3}{3} = 1$$

  - Appears in 50 documents

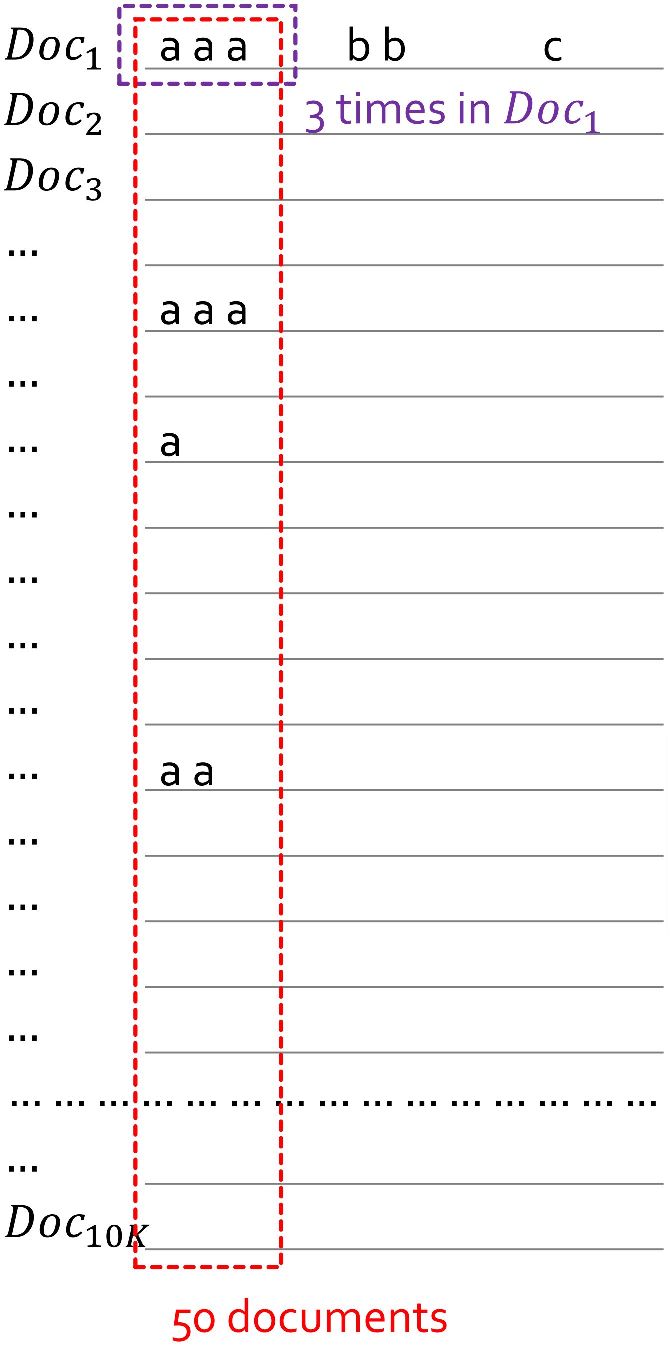


- In a corpus of 10,000 documents
- Term $\underline{a}$
  - Appears 3 times in Doc_1
  - Its normalized term frequency (TF) in Doc_1 is:

$$tf_{Doc_1,a} = \frac{3}{3} = 1$$

  - Appears in 50 documents
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10,000}{50}\right) = 5.29$$



- In a corpus of 10,000 documents
- Term $\underline{a}$
  - Appears 3 times in Doc_1
  - Its normalized term frequency (TF) in Doc_1 is:

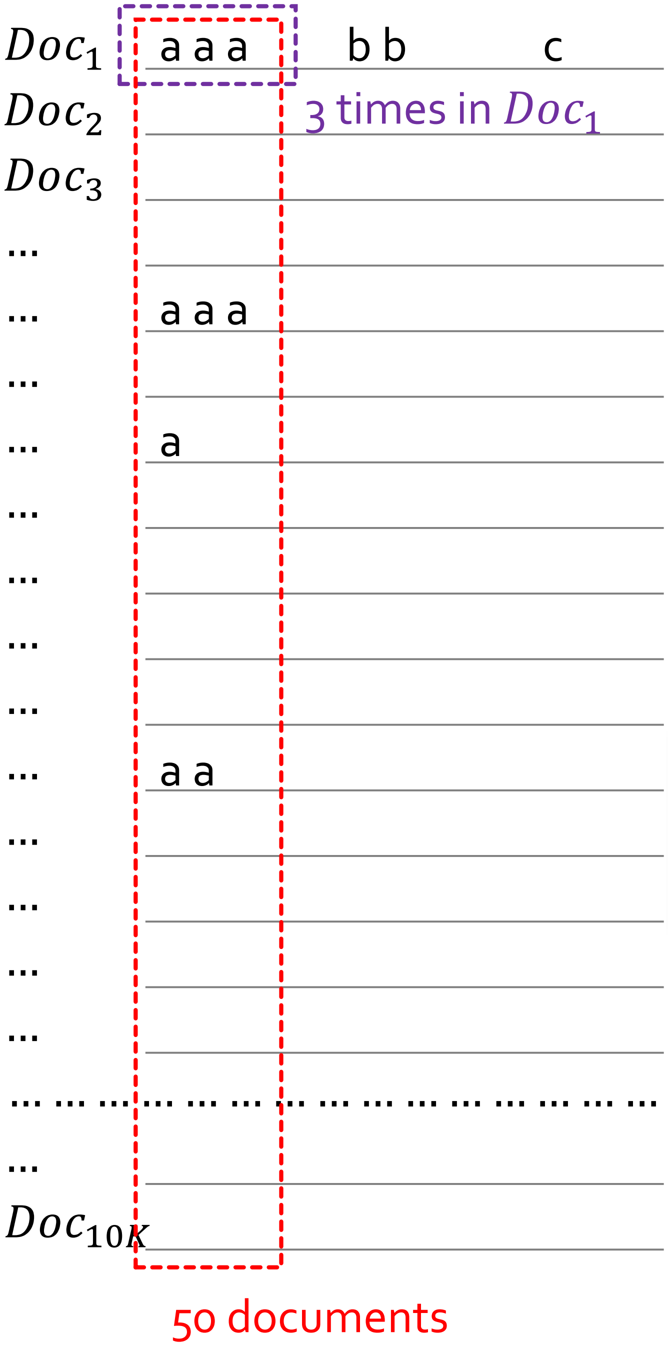
$$tf_{Doc_1,a} = \frac{3}{3} = 1$$

  - Appears in 50 documents
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10{,}000}{50}\right) = 5.29$$

  - $\mathrm{tfidf}_{\mathrm{Doc_1,a}} = tf_{Doc_1,a} * idf_a = 1 * 5.29 = \mathbf{5.29}$

"weight" ("importance") of term a in Doc_1 is 5.29 (i.e., the tf-idf)



- In a corpus of 10,000 documents
- Term $\underline{a}$
  - Appears 3 times in Doc_1
  - Its normalized term frequency (TF) in Doc_1 is:

$$tf_{Doc_1,a} = \frac{3}{3} = 1$$

  - Appears in 50 documents
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10{,}000}{50}\right) = 5.29$$

  - $\mathrm{tfidf}_{\mathrm{Doc_1,a}} = tf_{Doc_1,a} * idf_a = 1 * 5.29 = \mathbf{5.29}$

"weight" ("importance") of term a in Doc_1 is 5.29 (i.e., the tf-idf)

	a	b	c	…
Doc_1	$\boxed{\mathbf{5.29}}$	**1.36**	**1.22**	…
Doc_2	…	…	…	…

Weighted Term-by-Document Matrix

Doc_1	a a a b b c
Doc_2	
Doc_3	
...	
...	a a a
...	
...	a
...	
...	
...	
...	
...	a a
...	
...	
...	
...	
... ...	
...	
Doc_{10K}	

- In a corpus of 10,000 documents
- Term $\underline{a}$
  - Appears 3 times in Doc_1
  - Its normalized term frequency (TF) in Doc_1 is:

$$tf_{Doc_1,a} = \frac{3}{3} = 1$$

  - Appears in 50 documents
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10{,}000}{50}\right) = 5.29$$

  - $\text{tfidf}_{\text{Doc}_1,\text{a}} = tf_{Doc_1,a} * idf_a = 1 * 5.29 = \mathbf{5.29}$

- Term $\underline{b}$

2 times

Doc_1 a a a $\boxed{\text{b b}}$ c

Doc_2

Doc_3

…

… a a a

…

… a

…

…

…

…

… a a

…

…

…

…

… … … … … … … … … … … … … … …

…

Doc_{10K}

- In a corpus of 10,000 documents
- Term $\underline{a}$
  - Appears 3 times in Doc_1
  - Its normalized term frequency (TF) in Doc_1 is:

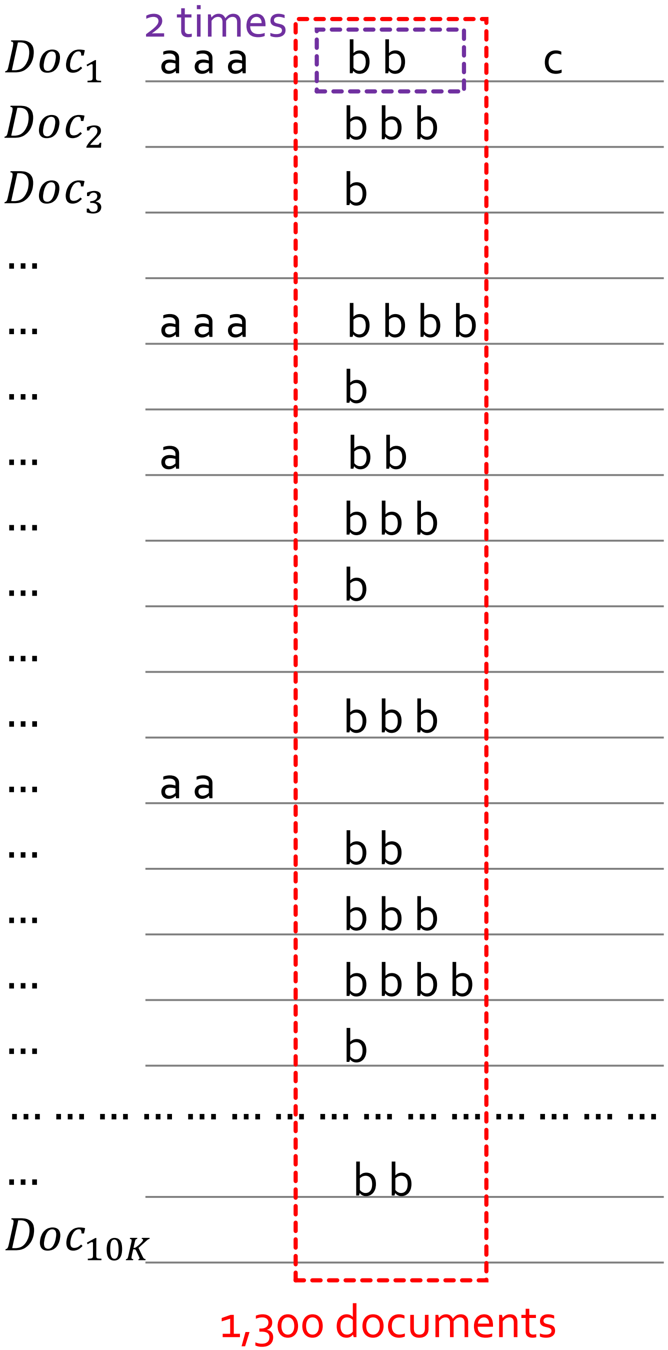
$$tf_{Doc_1,a} = \frac{3}{3} = 1$$

  - Appears in 50 documents
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10{,}000}{50}\right) = 5.29$$

  - $\text{tfidf}_{\text{Doc}_1,\text{a}} = tf_{Doc_1,a} * idf_a = 1 * 5.29 = \mathbf{5.29}$

- Term $\underline{b}$
  - Appears 2 times in Doc_1 $\Rightarrow tf_{Doc_1,b} = \frac{2}{3} = 0.66$



- In a corpus of 10,000 documents
- Term $\underline{a}$
  - Appears 3 times in Doc_1
  - Its normalized term frequency (TF) in Doc_1 is:

$$tf_{Doc_1,a} = \frac{3}{3} = 1$$

  - Appears in 50 documents
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10{,}000}{50}\right) = 5.29$$

  - $\mathrm{tfidf}_{\mathrm{Doc_1,a}} = tf_{Doc_1,a} * idf_a = 1 * 5.29 = \mathbf{5.29}$

- Term $\underline{b}$
  - Appears 2 times in Doc_1 $\Rightarrow tf_{Doc_1,b} = \frac{2}{3} = 0.66$
  - Appears in 1,300 documents$\Rightarrow idf_b = log\left(\frac{10{,}000}{1{,}300}\right) = 2.04$

		2 times	
Doc_1	a a a	b b	c
Doc_2		b b b	
Doc_3		b	
...			
...	a a a	b b b b	
...		b	
...	a	b b	
...		b b b	
...		b	
...			
...		b b b	
...	a a		
...		b b	
...		b b b	
...		b b b b	
...		b	
... ...	... ...	... ...	... ...
...		b b	
Doc_{10K}			

1,300 documents

- In a corpus of 10,000 documents
- Term $\underline{a}$
  - Appears 3 times in Doc_1
  - Its normalized term frequency (TF) in Doc_1 is:

$$tf_{Doc_1,a} = \frac{3}{3} = 1$$

  - Appears in 50 documents
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10{,}000}{50}\right) = 5.29$$

  - $\mathrm{tfidf}_{\mathrm{Doc_1,a}} = tf_{Doc_1,a} * idf_a = 1 * 5.29 = \mathbf{5.29}$

- Term $\underline{b}$
  - Appears 2 times in Doc_1 $\Rightarrow tf_{Doc_1,b} = \frac{2}{3} = 0.66$
  - Appears in 1,300 documents$\Rightarrow idf_b = log\left(\frac{10{,}000}{1{,}300}\right) = 2.04$
  - $\mathrm{tfidf}_{\mathrm{Doc_1,b}} = tf_{Doc_1,b} * idf_b = 0.66 * 2.04 = \mathbf{1.36}$

Doc_1	a a a	b b	c
Doc_2		b b b	
Doc_3		b	
…			
…	a a a	b b b b	
…		b	
…	a	b b	
…		b b b	
…		b	
…			
…		b b b	
…	a a		
…		b b	
…		b b b	
…		b b b b	
…		b	
… … … … …	… … … … …	… … … … …	
…		b b	
Doc_{10K}			

- In a corpus of 10,000 documents
- Term $\underline{a}$
  - Appears 3 times in Doc_1
  - Its normalized term frequency (TF) in Doc_1 is:
    $$tf_{Doc_1,a} = \frac{3}{3} = 1$$
  - Appears in 50 documents
  - Its *discriminative* power (IDF) will be
    $$idf_a = log\left(\frac{10{,}000}{50}\right) = 5.29$$
  - $\mathrm{tfidf}_{\mathrm{Doc_1,a}} = tf_{Doc_1,a} * idf_a = 1 * 5.29 = \mathbf{5.29}$
- Term $\underline{b}$
  - Appears 2 times in Doc_1 $\Rightarrow tf_{Doc_1,b} = \frac{2}{3} = 0.66$
  - Appears in 1,300 documents$\Rightarrow idf_b = log\left(\frac{10{,}000}{1{,}300}\right) = 2.04$
  - $\mathrm{tfidf}_{\mathrm{Doc_1,b}} = tf_{Doc_1,b} * idf_b = 0.66 * 2.04 = \mathbf{1.36}$
- Term $\underline{c}$

			1 time
Doc_1	a a a	b b	$\boxed{\text{c}}$
Doc_2		b b b	
Doc_3		b	
...			
...	a a a	b b b b	
...		b	
...	a	b b	
...		b b b	
...		b	
...			
...		b b b	
...	a a		
...		b b	
...		b b b	
...		b b b b	
...		b	
... ...	... ...	... ...	
...		b b	
Doc_{10K}			

- In a corpus of 10,000 documents
- Term $\underline{a}$
  - Appears 3 times in Doc_1
  - Its normalized term frequency (TF) in Doc_1 is:

$$tf_{Doc_1,a} = \frac{3}{3} = 1$$

  - Appears in 50 documents
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10{,}000}{50}\right) = 5.29$$

  - $\text{tfidf}_{\text{Doc}_1,\text{a}} = tf_{Doc_1,a} * idf_a = 1 * 5.29 = \mathbf{5.29}$

- Term $\underline{b}$
  - Appears 2 times in $Doc_1 \Rightarrow tf_{Doc_1,b} = \frac{2}{3} = 0.66$
  - Appears in 1,300 documents$\Rightarrow idf_b = log\left(\frac{10{,}000}{1{,}300}\right) = 2.04$
  - $\text{tfidf}_{\text{Doc}_1,\text{b}} = tf_{Doc_1,b} * idf_b = 0.66 * 2.04 = \mathbf{1.36}$
- Term $\underline{c}$
  - Appears 1 time in $Doc_1 \Rightarrow tf_{Doc_1,c} = \frac{1}{3} = 0.33$

1 time

Doc_1	a a a	b b	c
Doc_2		b b b	
Doc_3		b	c c c
…			c c
…	a a a	b b b b	
…		b	
…	a	b b	c c c
…		b b b	
…		b	
…			c c
…		b b b	
…	a a		c c c c
…		b b	
…		b b b	
…		b b b b	
…		b	
… … … …	… … … …	… … … …	… … … …
…		b b	
Doc_{10K}			c

250 documents

- In a corpus of 10,000 documents
- Term $\underline{a}$
  - Appears 3 times in Doc_1
  - Its normalized term frequency (TF) in Doc_1 is:

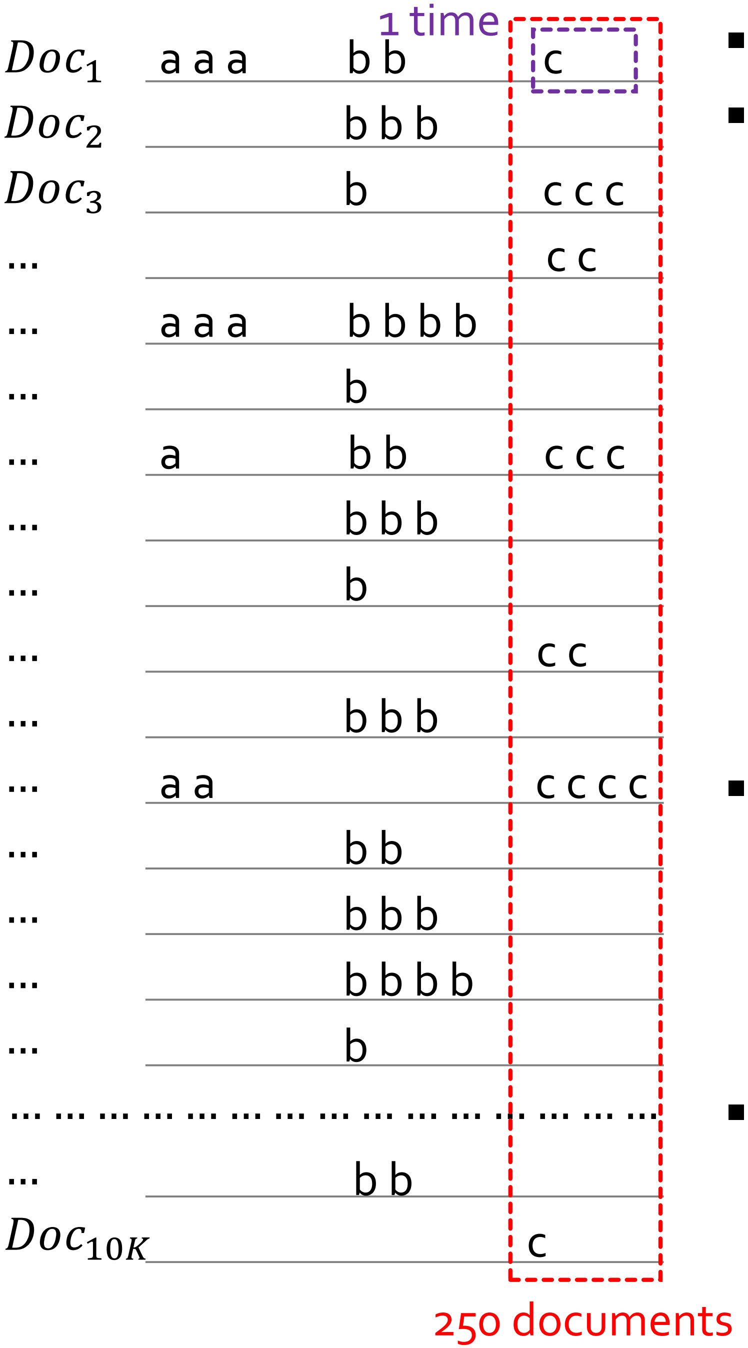
$$tf_{Doc_1,a} = \frac{3}{3} = 1$$

  - Appears in 50 documents
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10{,}000}{50}\right) = 5.29$$

  - $\mathrm{tfidf}_{\mathrm{Doc_1,a}} = tf_{Doc_1,a} * idf_a = 1 * 5.29 = \mathbf{5.29}$

- Term $\underline{b}$
  - Appears 2 times in Doc_1 $\Rightarrow tf_{Doc_1,b} = \frac{2}{3} = 0.66$
  - Appears in 1,300 documents$\Rightarrow idf_b = log\left(\frac{10{,}000}{1{,}300}\right) = 2.04$
  - $\mathrm{tfidf}_{\mathrm{Doc_1,b}} = tf_{Doc_1,b} * idf_b = 0.66 * 2.04 = \mathbf{1.36}$
- Term $\underline{c}$
  - Appears 1 time in Doc_1 $\Rightarrow tf_{Doc_1,c} = \frac{1}{3} = 0.33$
  - Appears in 250 documents $\Rightarrow idf_c = log\left(\frac{10{,}000}{250}\right) = 3.68$



- In a corpus of 10,000 documents
- Term $\underline{a}$
  - Appears 3 times in Doc_1
  - Its normalized term frequency (TF) in Doc_1 is:

$$tf_{Doc_1,a} = \frac{3}{3} = 1$$

  - Appears in 50 documents
  - Its *discriminative* power (IDF) will be

$$idf_a = log\left(\frac{10{,}000}{50}\right) = 5.29$$

  - $\mathrm{tfidf}_{\mathrm{Doc_1,a}} = tf_{Doc_1,a} * idf_a = 1 * 5.29 = \mathbf{5.29}$

- Term $\underline{b}$
  - Appears 2 times in Doc_1 $\Rightarrow tf_{Doc_1,b} = \frac{2}{3} = 0.66$
  - Appears in 1,300 documents$\Rightarrow idf_b = log\left(\frac{10{,}000}{1{,}300}\right) = 2.04$
  - $\mathrm{tfidf}_{\mathrm{Doc_1,b}} = tf_{Doc_1,b} * idf_b = 0.66 * 2.04 = \mathbf{1.36}$
- Term $\underline{c}$
  - Appears 1 time in Doc_1 $\Rightarrow tf_{Doc_1,c} = \frac{1}{3} = 0.33$
  - Appears in 250 documents $\Rightarrow idf_c = log\left(\frac{10{,}000}{250}\right) = 3.68$
  - $tfidf_{Doc_1,\mathrm{c}} = tf_{Doc_1,c} * idf_c = 0.33 * 3.68 = \mathbf{1.22}$

Doc_1	a a a	b b	c
Doc_2		b b b	
Doc_3		b	c c c
…			c c
…	a a a	b b b b	
…		b	
…	a	b b	c c c
…			
…			
…			
…			
…			
…			
…			
…			
…			
… … … … … … … … … … … … … … …			
…		b b	
Doc_{10K}			c

- In a corpus of 10,000 documents
- Term $\underline{a}$
  - Appears 3 times in Doc_1
  - Its normalized term frequency (TF) in Doc_1 is:

$$tf_{Doc_1,a} = \frac{3}{3} = 1$$

  - Appears in 50 documents
  - Its *discriminative* power (IDF) will be

Higher discriminating power is given to terms that

- appear a lot in the same document, but
- few times in the entire corpus

A word that appears in every document is basically meaningless

04

- Term $\underline{c}$
  - Appears 1 time in $Doc_1 \Rightarrow tf_{Doc_1,c} = \frac{1}{3} = 0.33$
  - Appears in 250 documents $\Rightarrow idf_c = log\left(\frac{10,000}{250}\right) = 3.68$
  - $tfidf_{Doc_1,c} = tf_{Doc_1,c} * idf_c = 0.33 * 3.68 = \mathbf{1.22}$

IDF Formulas

(FYI – Not Required for exams)

df_i = "document frequency" of term i

= number of documents containing term i

N = total number of documents in the corpus

$idf_i = log_2\left(\frac{N}{df_i}\right)$

= inverse document frequency of term i

- Log used to dampen the effect relative to *TF*

TF-IDF Formulas

(FYI – Not Required for exams)

$$w_{ik} = tf_{ik} * idf_k = tf_{ik} * \log\left(\frac{N}{n_k}\right)$$

w_{ik} = <u>"weight" ("importance")</u> of term k in doc. D_i

T_k = term k in document D_i

tf_{ik} = (normalized) frequency of term T_k in doc. D_i

idf_k = inverse document frequency of term T_k

$$= \log\left(\frac{N}{n_k}\right)$$

N = total number of documents in the corpus

n_k = the number of documents in the corpus that contain T_k

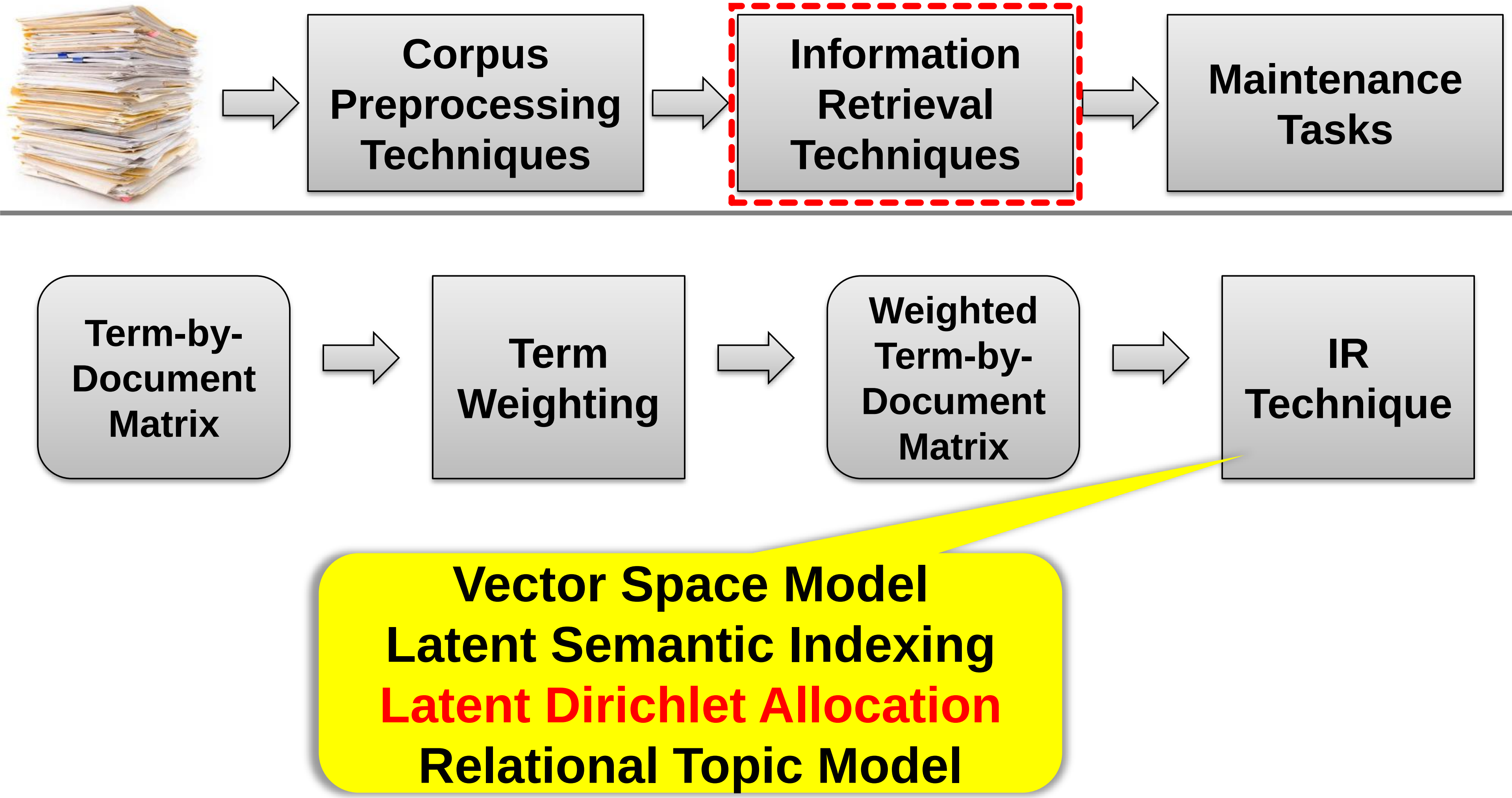
Corpus Preprocessing Techniques → **Information Retrieval Techniques** → Maintenance Tasks

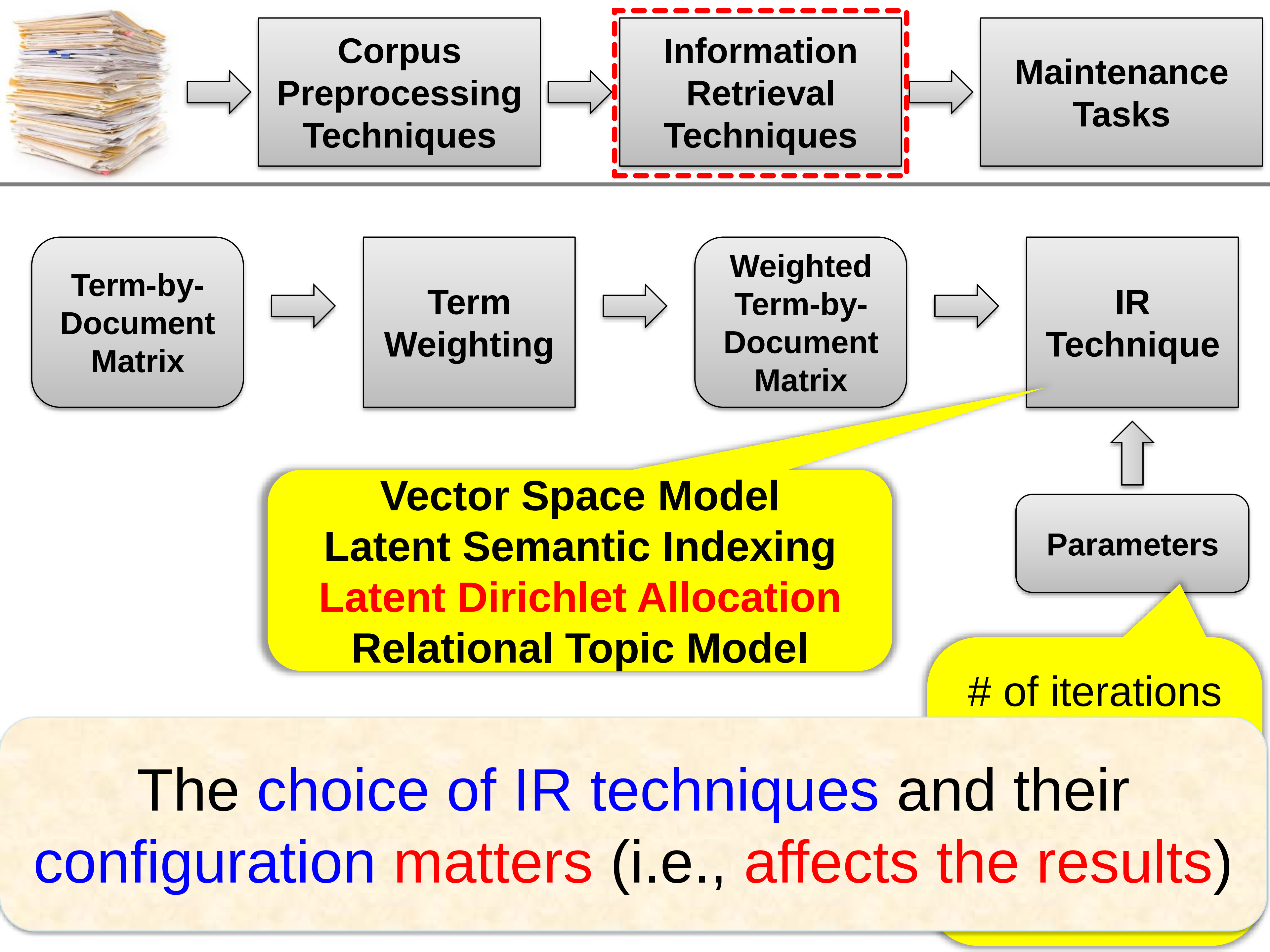
Term-by-Document Matrix → Term Weighting → Weighted Term-by-Document Matrix → IR Technique

	print	test	result	...
m_1	**5**	**1**	**3**	...
m_2	**1**	**0**	**0**	...

boolean
tf
tf-idf
log(tf+1)
tf-entropy

	print	test	result	...
m_1	**0.78**	**0.2**	**0.9**	...
m_2	**0.45**	**0**	**0**	...


Corpus Preprocessing Techniques
Information Retrieval Techniques
Maintenance Tasks
Term-by-Document Matrix
Term Weighting
Weighted Term-by-Document Matrix
IR Technique
Vector Space Model
Latent Semantic Indexing
Latent Dirichlet Allocation
Relational Topic Model



Corpus Preprocessing Techniques
Information Retrieval Techniques
Maintenance Tasks
Term-by-Document Matrix
Term Weighting
Weighted Term-by-Document Matrix
IR Technique
Vector Space Model
Latent Semantic Indexing
Latent Dirichlet Allocation
Relational Topic Model
Parameters
# of iterations
The choice of IR techniques and their configuration matters (i.e., affects the results)

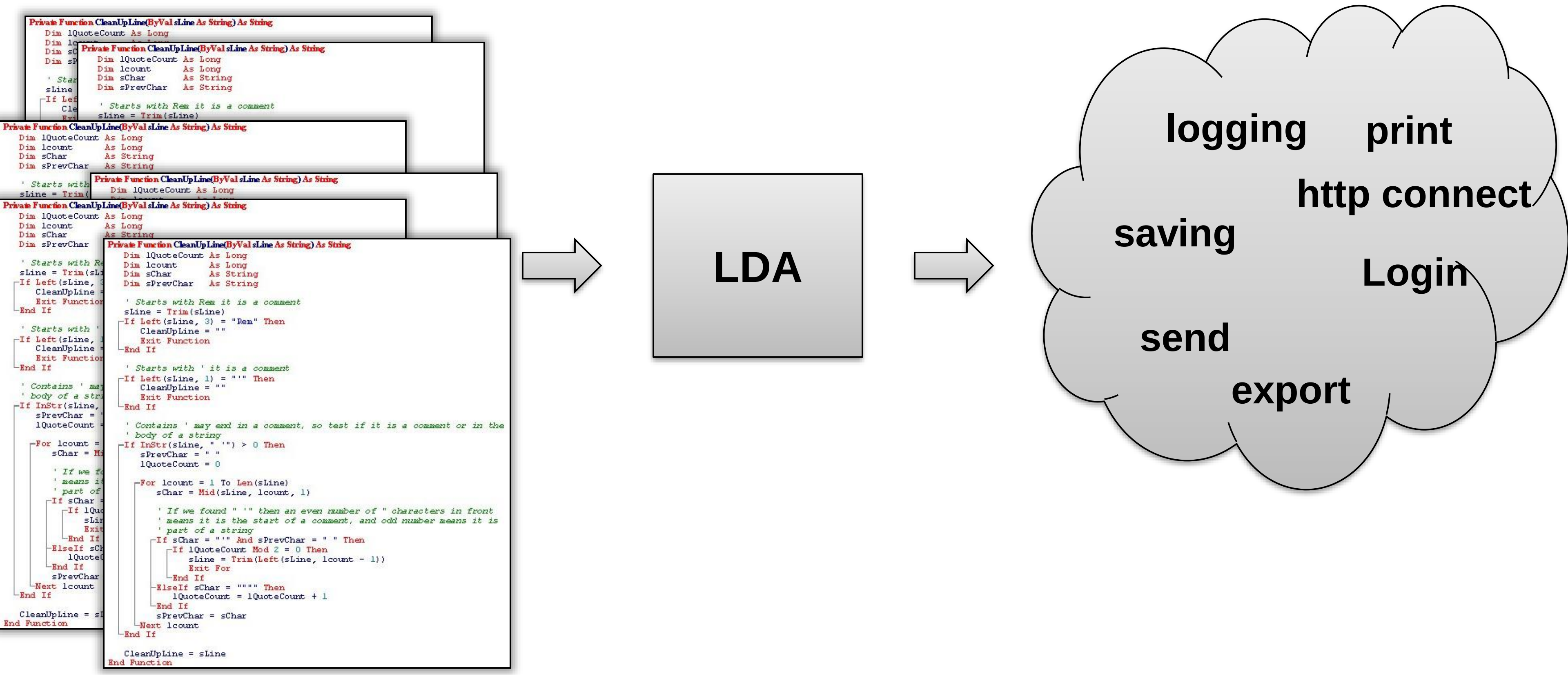
Latent Dirichlet Allocation (LDA)*

- Topic model that generates the distribution of latent topics from textual documents

*[Blei'03]

Latent Dirichlet Allocation (LDA)

- Topic model that generates the distribution of latent topics from textual documents



Example of Topics

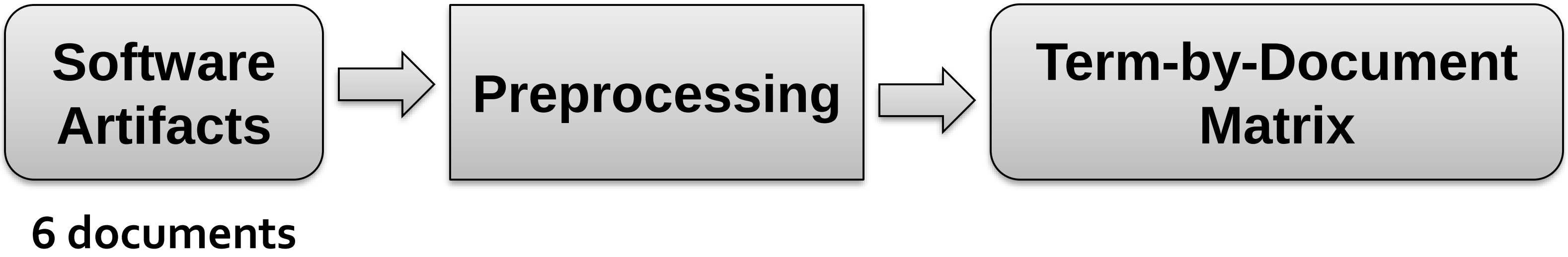
Topic 1	
Image	0.09
Player	0.09
Path	0.07
Data	0.04
Audio	0.03
...	...

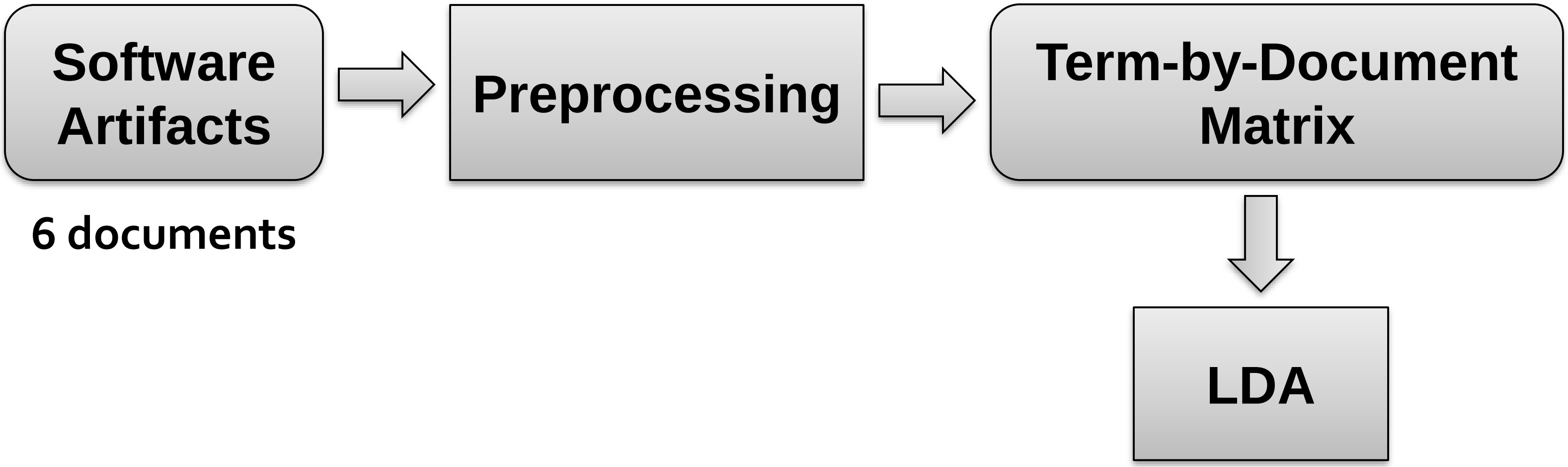
Topic 2	
android	0.13
activity	0.09
intent	0.08
thread	0.05
runtime	0.03
...	...

Topic 3	
database	0.11
string	0.11
table	0.07
sqllite	0.05
id	0.03
...	...

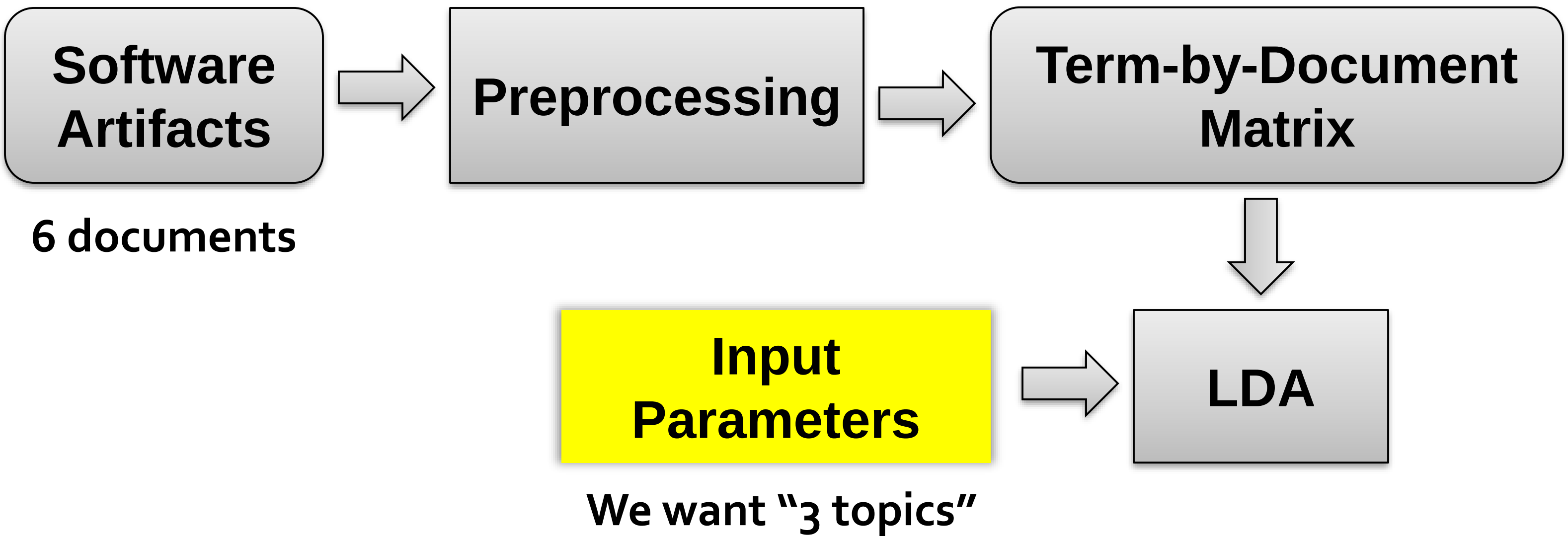
Software Artifacts

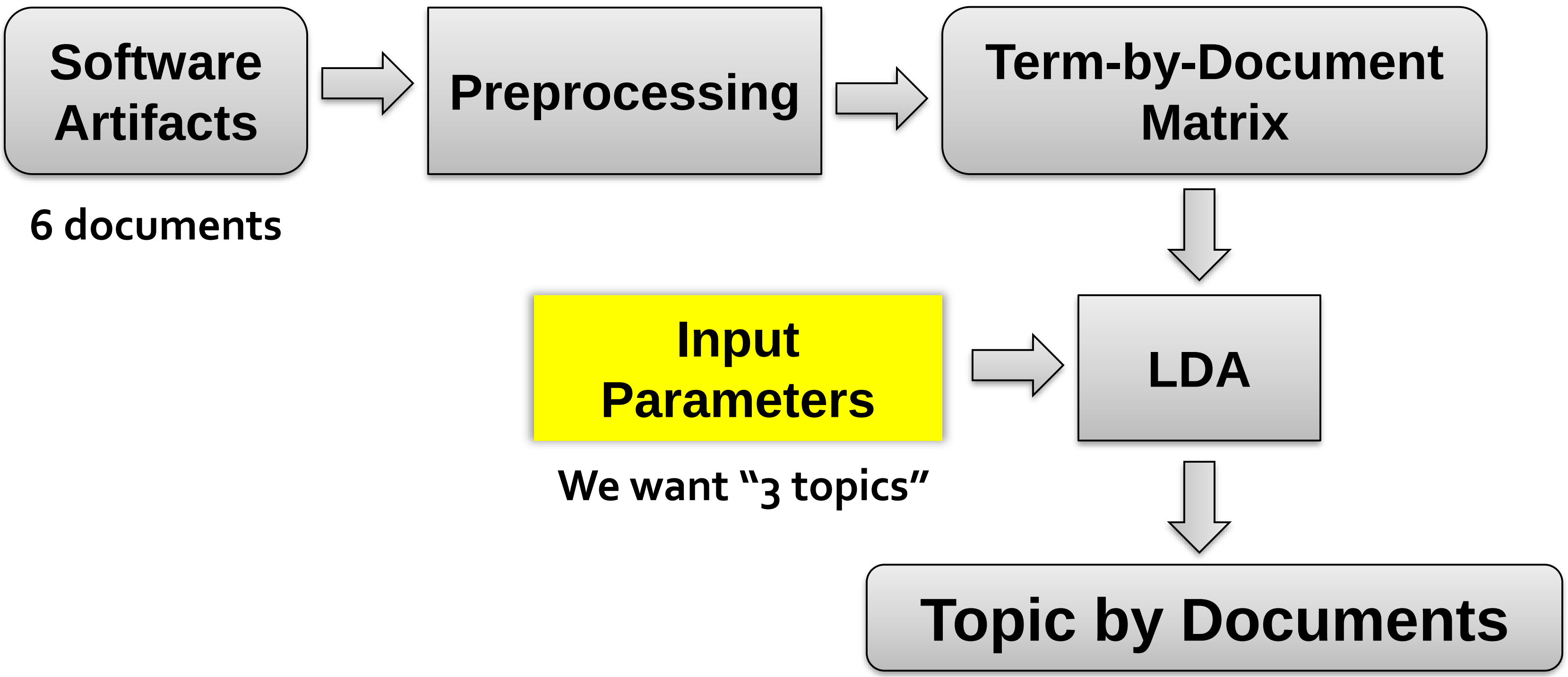
6 documents

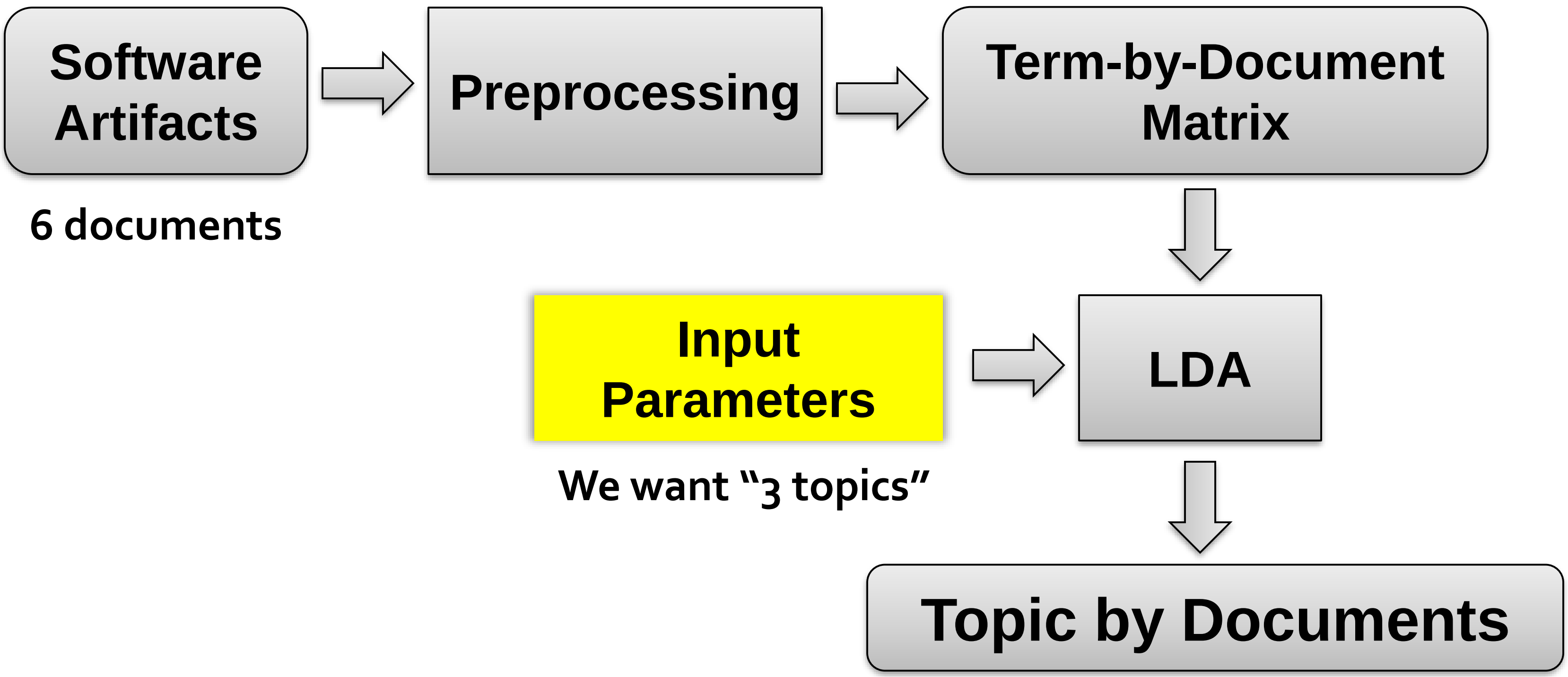

Software Artifacts
Preprocessing
Term-by-Document Matrix
6 documents


Software Artifacts
Preprocessing
Term-by-Document Matrix
LDA

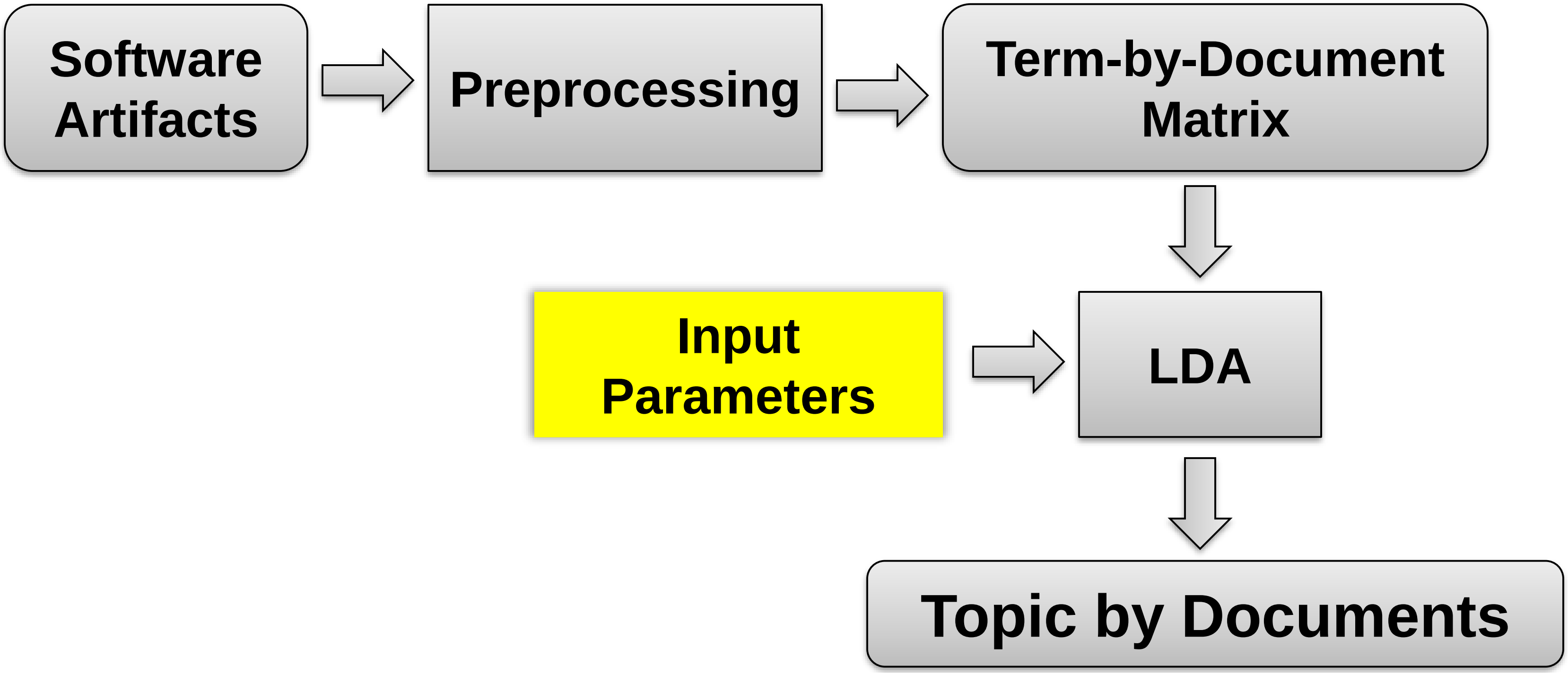
6 documents


Software Artifacts
Preprocessing
Term-by-Document Matrix
6 documents
Input Parameters
LDA
We want "3 topics"


Software Artifacts
6 documents
Preprocessing
Term-by-Document Matrix
Input Parameters
We want "3 topics"
LDA
Topic by Documents

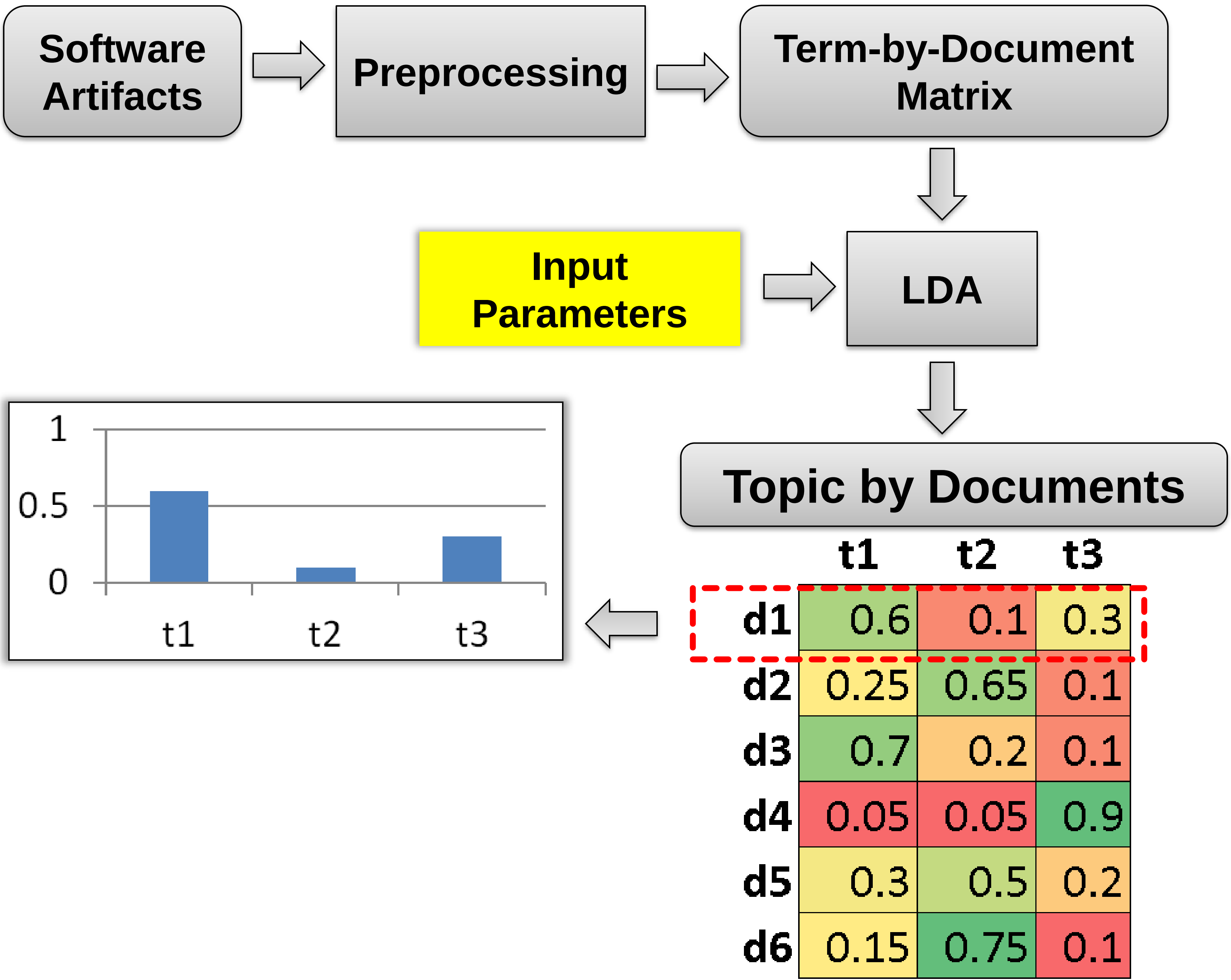


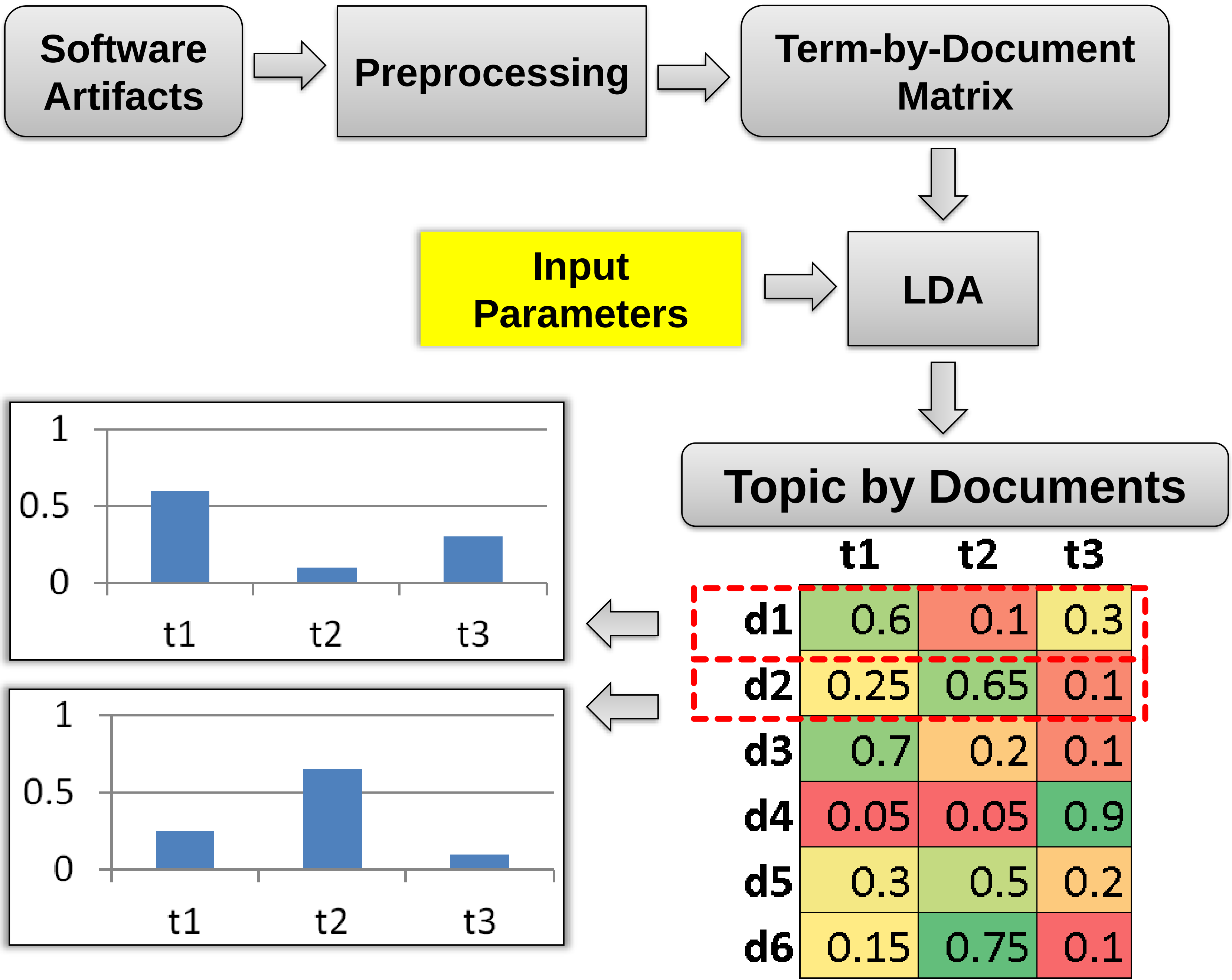
	t1	t2	t3
d1	0.6	0.1	0.3
d2	0.25	0.65	0.1
d3	0.7	0.2	0.1
d4	0.05	0.05	0.9
d5	0.3	0.5	0.2
d6	0.15	0.75	0.1



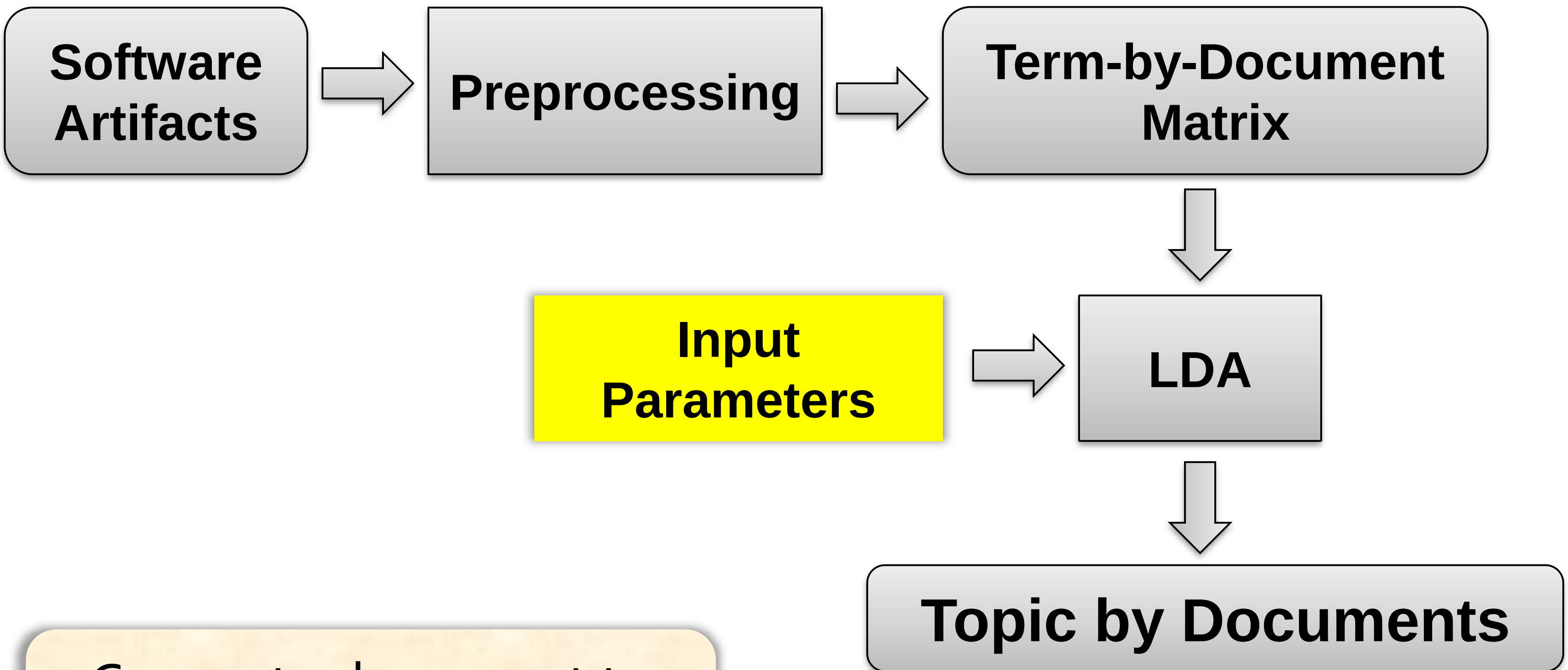
Probability that document is related to topic

	t1	t2	t3
d1	0.6	0.1	0.3
d2	0.25	0.65	0.1
d3	0.7	0.2	0.1
d4	0.05	0.05	0.9
d5	0.3	0.5	0.2
d6	0.15	0.75	0.1


Software Artifacts
Preprocessing
Term-by-Document Matrix
Input Parameters
LDA
Topic by Documents
t1
t2
t3
d1 0.6 0.1 0.3
d2 0.25 0.65 0.1
d3 0.7 0.2 0.1
d4 0.05 0.05 0.9
d5 0.3 0.5 0.2
d6 0.15 0.75 0.1
1
0.5
0
t1
t2
t3



Software Artifacts
Preprocessing
Term-by-Document Matrix
Input Parameters
LDA
Topic by Documents
t1 t2 t3
d1 0.6 0.1 0.3
d2 0.25 0.65 0.1
d3 0.7 0.2 0.1
d4 0.05 0.05 0.9
d5 0.3 0.5 0.2
d6 0.15 0.75 0.1
1
0.5
0
t1 t2 t3
1
0.5
0
t1 t2 t3

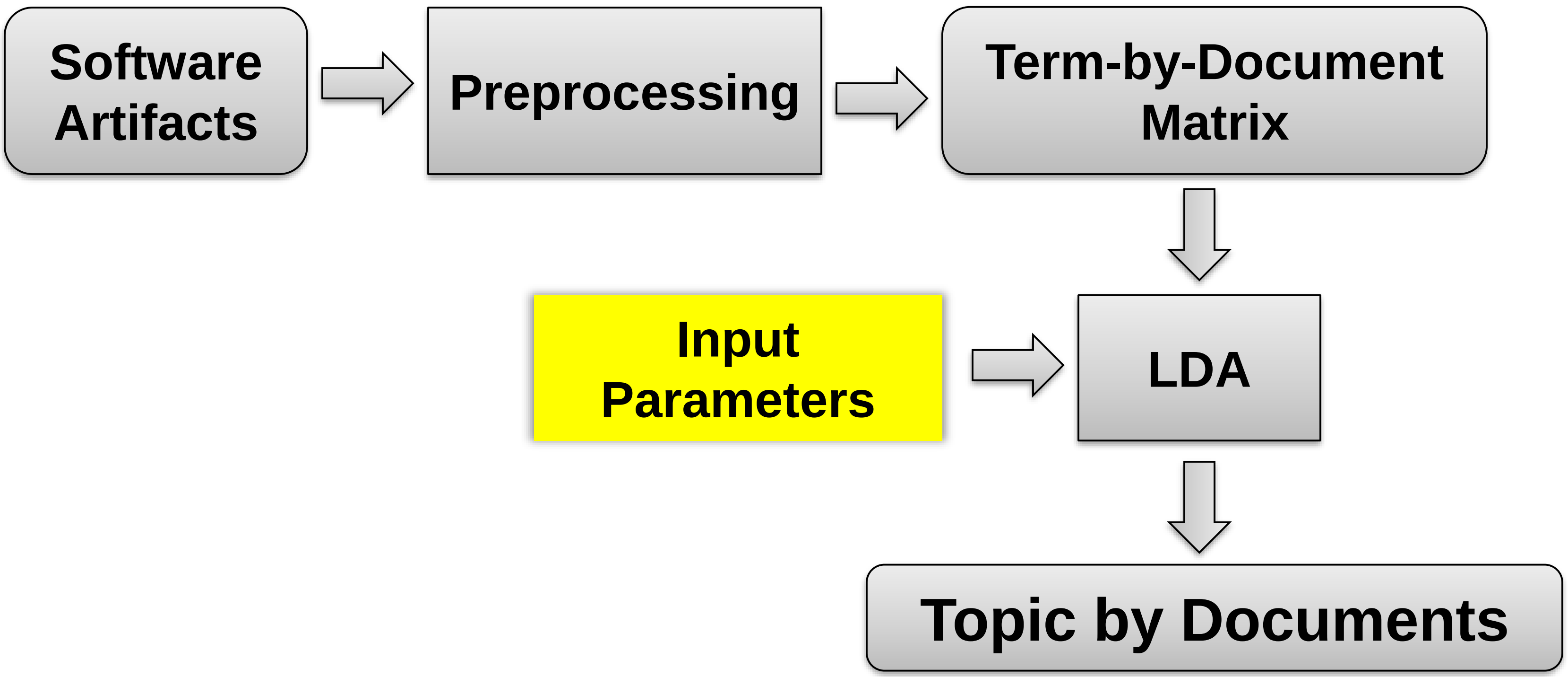


Compute document to document similarity

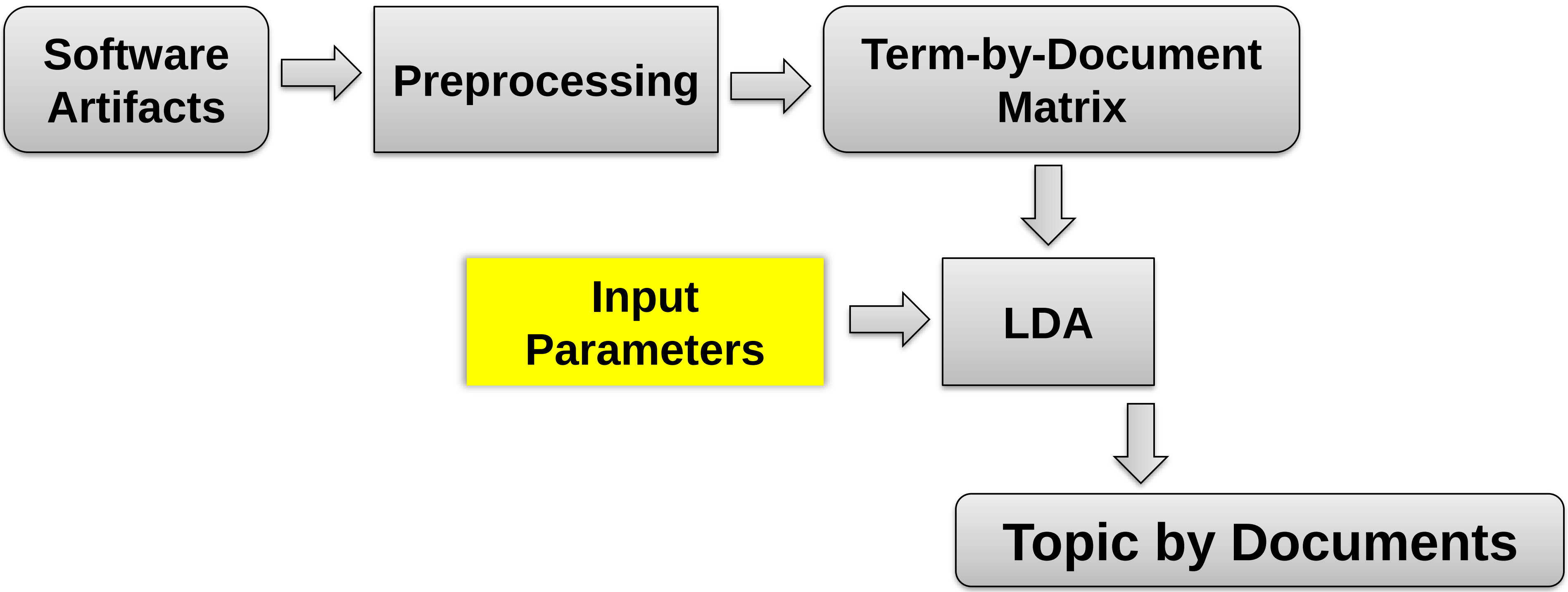
Compute query to document similarity

“Cluster” documents by topics

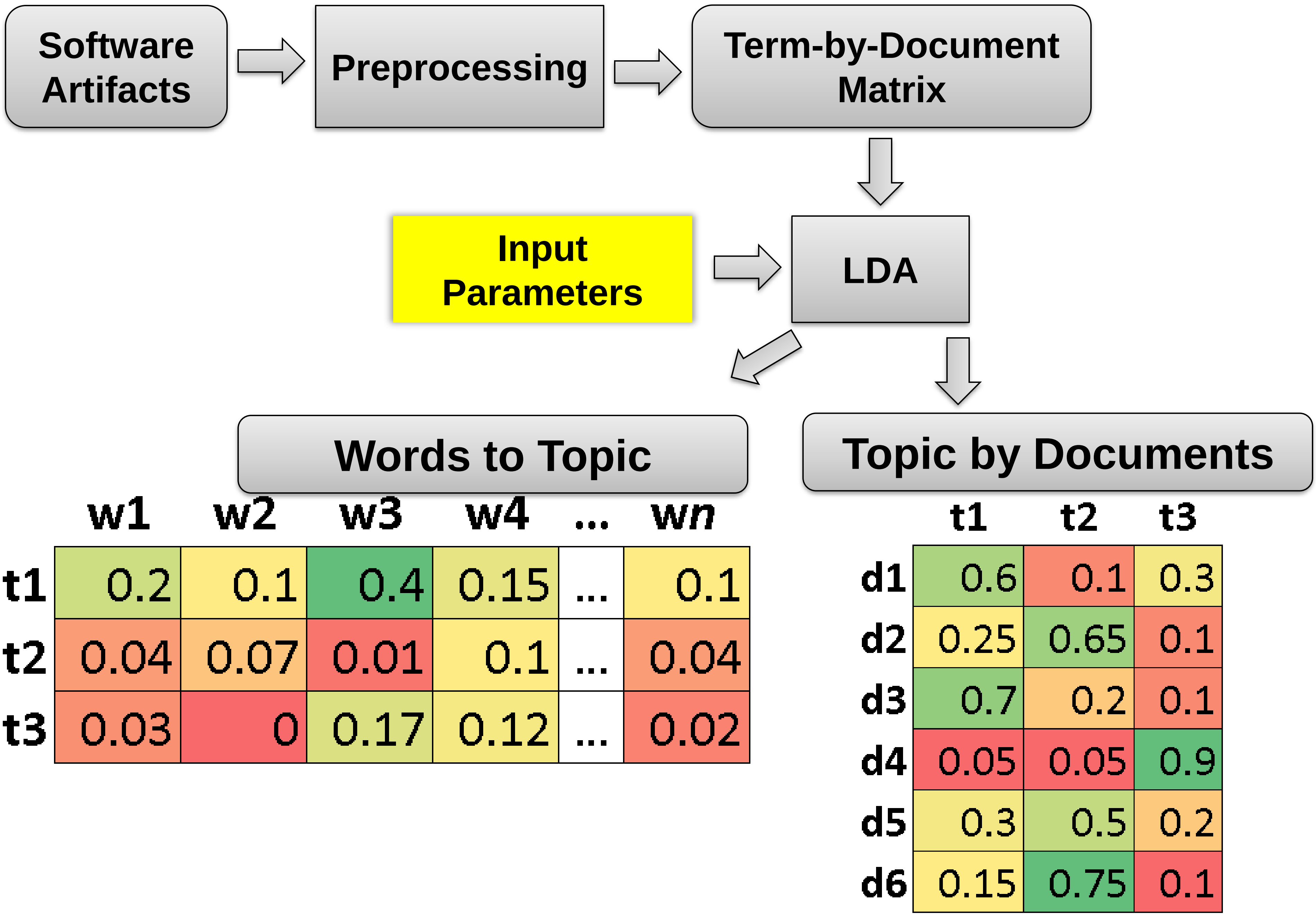
	t1	t2	t3
d1	0.6	0.1	0.3
d2	0.25	0.65	0.1
d3	0.7	0.2	0.1
d4	0.05	0.05	0.9
d5	0.3	0.5	0.2
d6	0.15	0.75	0.1

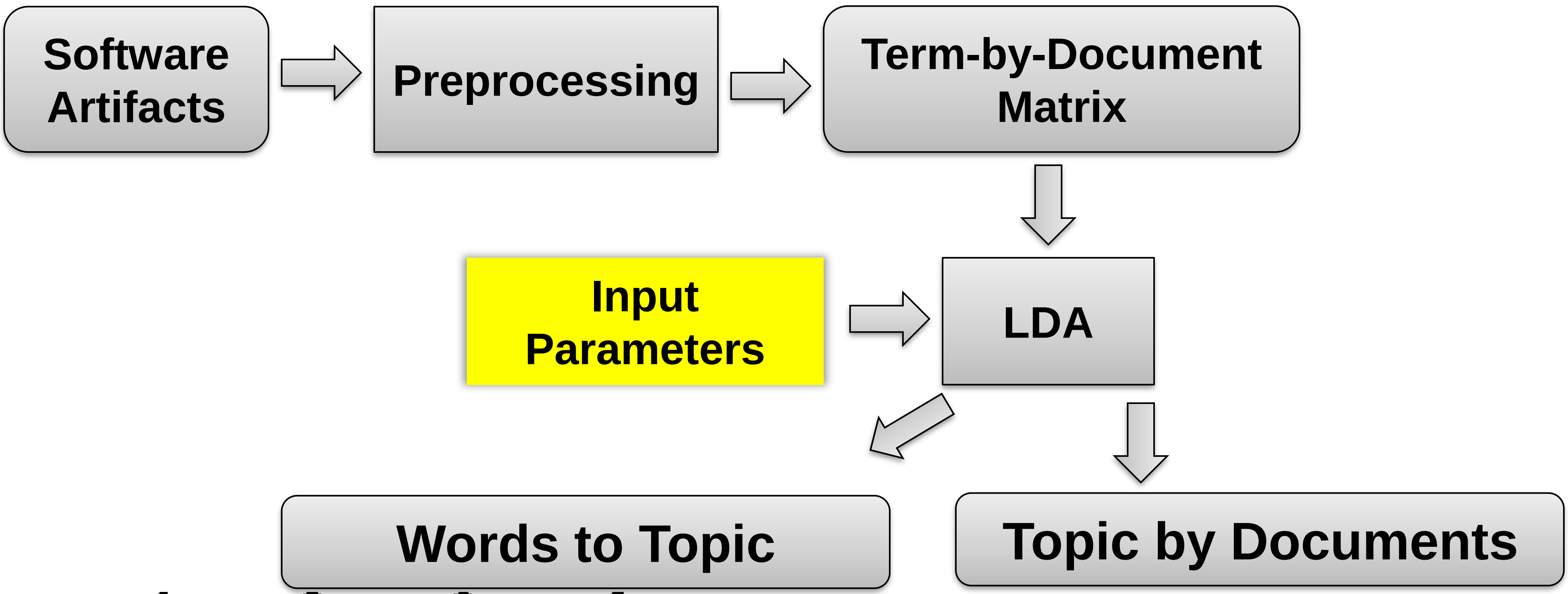


	t1	t2	t3
d1	0.6	0.1	0.3
d2	0.25	0.65	0.1
d3	0.7	0.2	0.1
d4	0.05	0.05	0.9
d5	0.3	0.5	0.2
d6	0.15	0.75	0.1

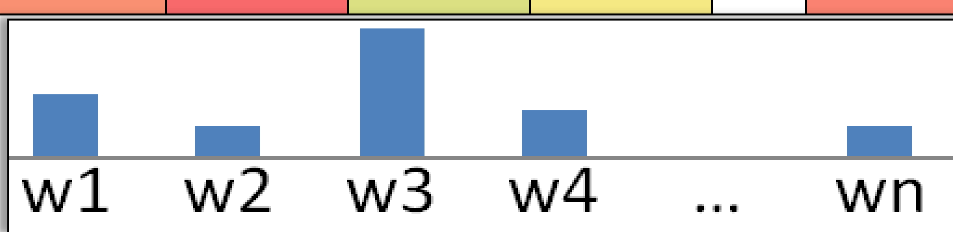


	t1	t2	t3
d1	0.6	0.1	0.3
d2	0.25	0.65	0.1
d3	0.7	0.2	0.1
d4	0.05	0.05	0.9
d5	0.3	0.5	0.2
d6	0.15	0.75	0.1

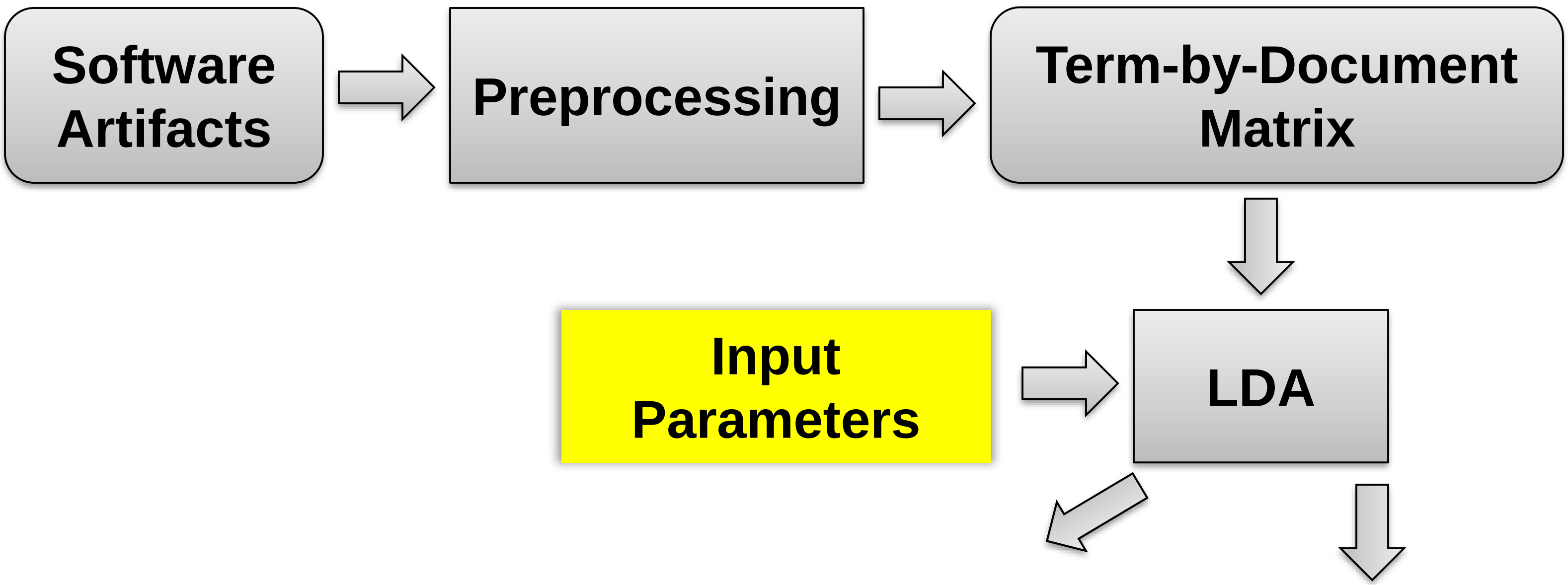

Software Artifacts
Preprocessing
Term-by-Document Matrix
Input Parameters
LDA
Words to Topic
w1 w2 w3 w4 ... wn
t1 0.2 0.1 0.4 0.15 ... 0.1
t2 0.04 0.07 0.01 0.1 ... 0.04
t3 0.03 0 0.17 0.12 ... 0.02
Topic by Documents
t1 t2 t3
d1 0.6 0.1 0.3
d2 0.25 0.65 0.1
d3 0.7 0.2 0.1
d4 0.05 0.05 0.9
d5 0.3 0.5 0.2
d6 0.15 0.75 0.1



	w1	w2	w3	w4	...	wn
t1	0.2	0.1	0.4	0.15	...	0.1
t2	0.04	0.07	0.01	0.1	...	0.04
t3	0.03	0	0.17	0.12	...	0.02



	t1	t2	t3
d1	0.6	0.1	0.3
d2	0.25	0.65	0.1
d3	0.7	0.2	0.1
d4	0.05	0.05	0.9
d5	0.3	0.5	0.2
d6	0.15	0.75	0.1



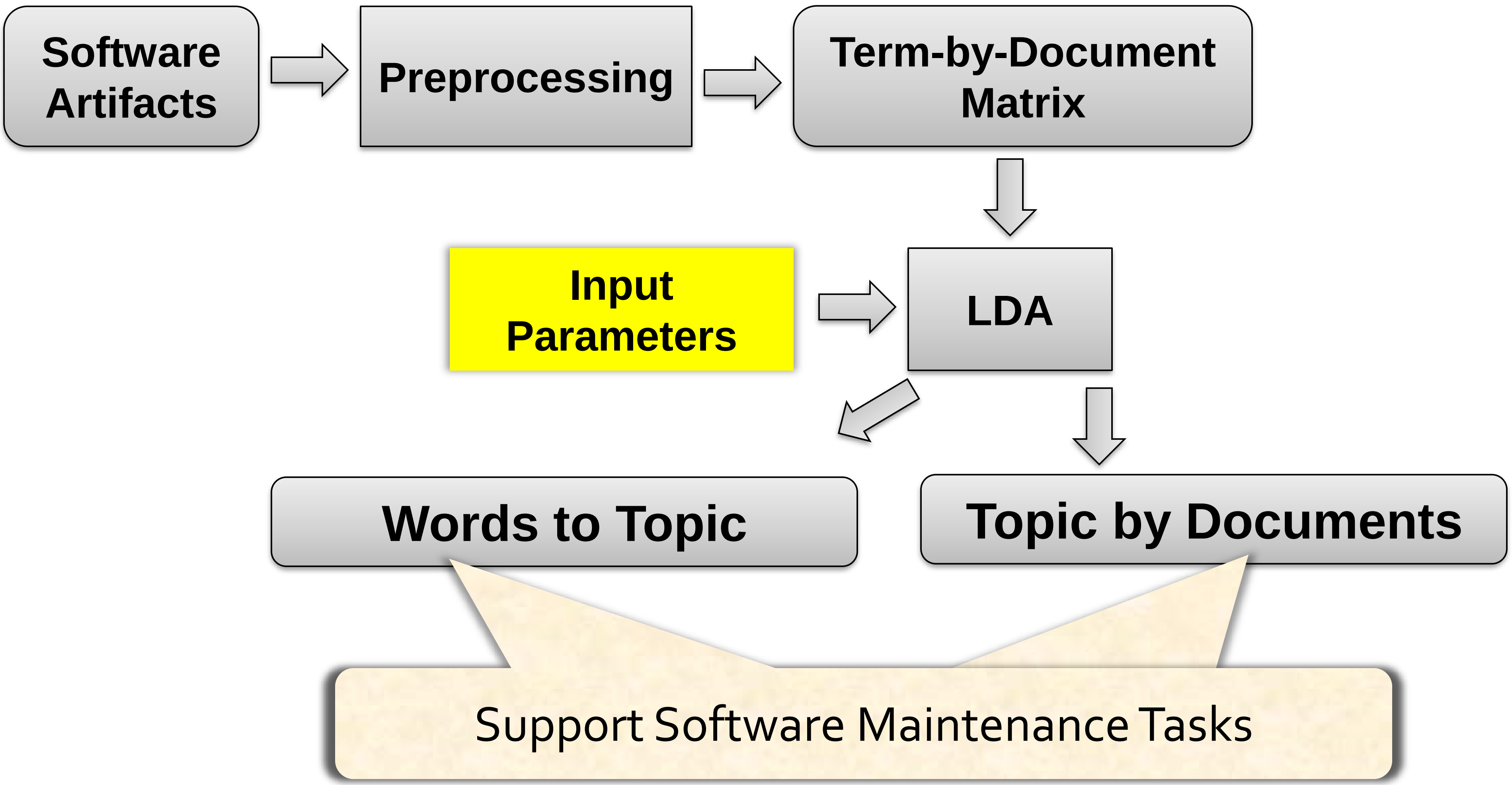
Words to Topic

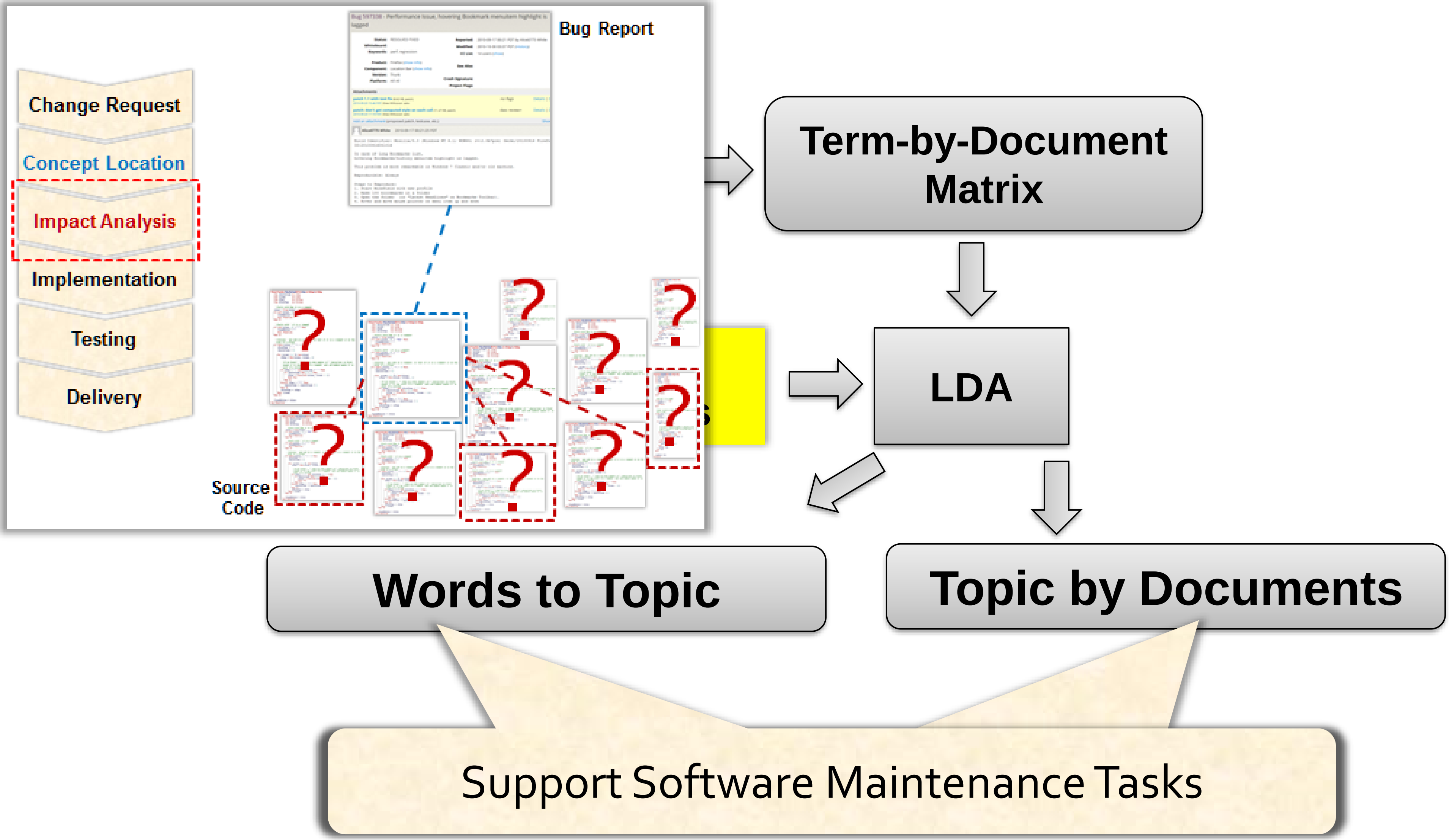
	w1	w2	w3	w4	...	wn
t1	0.2	0.1	0.4	0.15	...	0.1
t2	0.04	0.07	0.01	0.1	...	0.04
t3	0.03	0	0.17	0.12	...	0.02

Generate topic “labels”
(top 5 words)

Topic by Documents

	t1	t2	t3
d1	0.6	0.1	0.3
d2	0.25	0.65	0.1
d3	0.7	0.2	0.1
d4	0.05	0.05	0.9
d5	0.3	0.5	0.2
d6	0.15	0.75	0.1


Software Artifacts
Preprocessing
Term-by-Document Matrix
Input Parameters
LDA
Words to Topic
Topic by Documents
Support Software Maintenance Tasks


Change Request
Concept Location
Impact Analysis
Implementation
Testing
Delivery
Bug Report
Source Code
Term-by-Document Matrix
LDA
Words to Topic
Topic by Documents
Support Software Maintenance Tasks

Corpus Preprocessing Techniques

Information Retrieval Techniques

Maintenance Tasks

Term-by-Document Matrix

Term Weighting

Weighted Term-by-Document Matrix

IR Technique

Vector Space Model

Latent Semantic Indexing

Latent Dirichlet Allocation

Relational Topic Model

Graphic Representation
Vector Space Model

D_1: T_1 T_1 $T_2 T_2 T_2$ T_3 T_3 T_3 T_3 T_3

D_2: T_1 T_1 T_1 T_2 T_2 T_2 T_2 T_2 T_2 T_2 T_3

Search query Q : T_3 T_3

Graphic Representation
Vector Space Model

Example:

$D_1 = 2T_1 + 3T_2 + 5T_3$

$D_2 = 3T_1 + 7T_2 + T_3$

$Q = 0T_1 + 0T_2 + 2T_3$

D_1: T_1 T_1 $T_2 T_2 T_2$ T_3 T_3 T_3 T_3 T_3

D_2: T_1 T_1 T_1 T_2 T_2 T_2 T_2 T_2 T_2 T_2 T_3

Search query Q : T_3 T_3

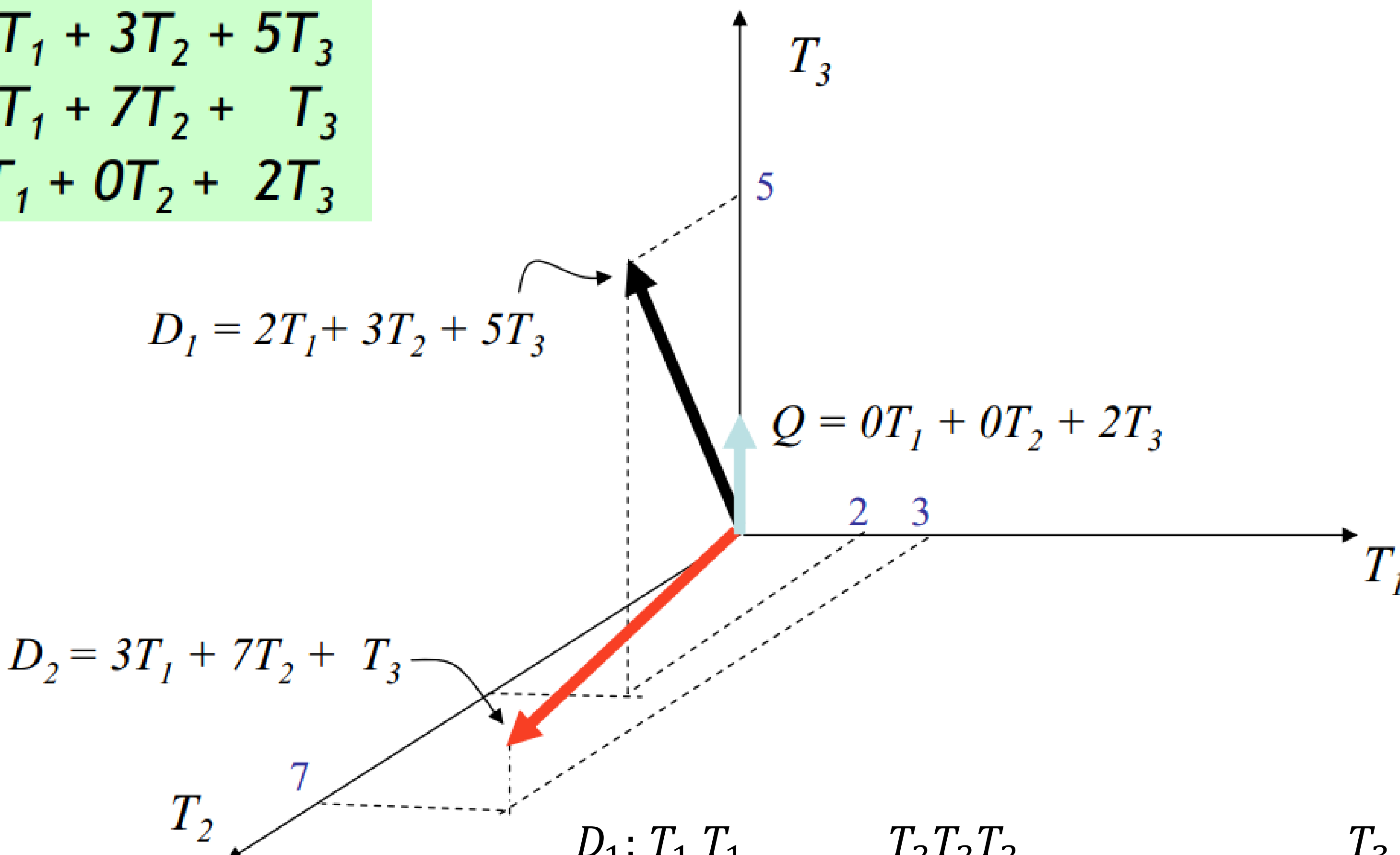
Graphic Representation
Vector Space Model

Example:

$D_1 = 2T_1 + 3T_2 + 5T_3$

$D_2 = 3T_1 + 7T_2 + T_3$

$Q = 0T_1 + 0T_2 + 2T_3$



D_1: $T_1\ T_1 \quad T_2 T_2 T_2 \quad T_3\ T_3\ T_3\ T_3\ T_3$

D_2: $T_1\ T_1\ T_1 \quad T_2\ T_2\ T_2\ T_2\ T_2\ T_2\ T_2 \quad T_3$

Search query Q : $T_3\ T_3$

Graphic Representation
Vector Space Model

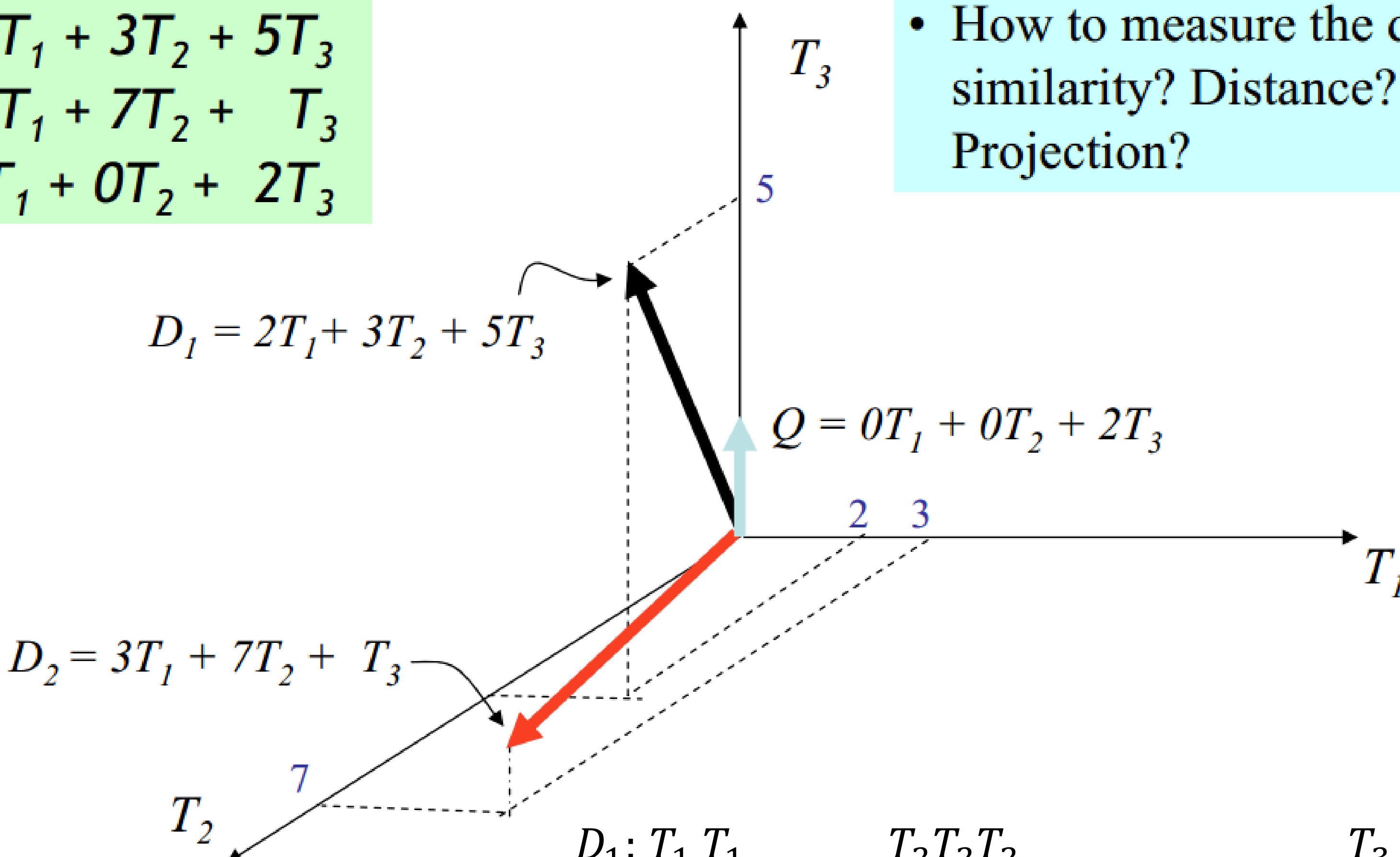
Example:

$D_1 = 2T_1 + 3T_2 + 5T_3$

$D_2 = 3T_1 + 7T_2 + T_3$

$Q = 0T_1 + 0T_2 + 2T_3$

- Is D_1 or D_2 more similar to Q?
- How to measure the degree of similarity? Distance? Angle? Projection?



D_1: T_1 T_1 $T_2 T_2 T_2$ T_3 T_3 T_3 T_3 T_3

D_2: T_1 T_1 T_1 T_2 T_2 T_2 T_2 T_2 T_2 T_2 T_3

Search query Q : T_3 T_3

How to measure degree of similarity? Use cosine similarity

$$sim(Q, D_i) = \frac{\sum_{j=1}^{t} w_{qj} * w_{d_{ij}}}{\sqrt{\sum_{j=1}^{t} (w_{qj})^2 * \sum_{j=1}^{t} (w_{d_{ij}})^2}}$$

MATLAB Scripts

```
function [cosineSimilarity]=calculateCosineSimilarity(vectorX,vectorY)

    cosineSimilarity=sum(vectorX.*vectorY)/sqrt(sum(vectorX.^2)*sum(vectorY.^2));

end
```

$$sim(Q, D_i) = \frac{\sum_{j=1}^{t} w_{qj} * w_{d_{ij}}}{\sqrt{\sum_{j=1}^{t} (w_{qj})^2 * \sum_{j=1}^{t} (w_{d_{ij}})^2}}$$

MATLAB Scripts

```
function [cosineSimilarity]=calculateCosineSimilarity(vectorX,vectorY)

    cosineSimilarity=sum(vectorX.*vectorY)/sqrt(sum(vectorX.^2)*sum(vectorY.^2));

end
```

```
>> D1=[2 3 5]
D1 =
     2     3     5

>> D2=[3 7 1]
D2 =
     3     7     1

>> Q=[0 0 2]
Q =
     0     0     2

>> calculateCosineSimilarity(Q,D1)
ans =
    0.8111

>> calculateCosineSimilarity(Q,D2)
ans =
    0.1302
```

```
cos(Q,D1)=0.8111 //D1 is more similar to Q
cos(Q,D2)=0.1302
```

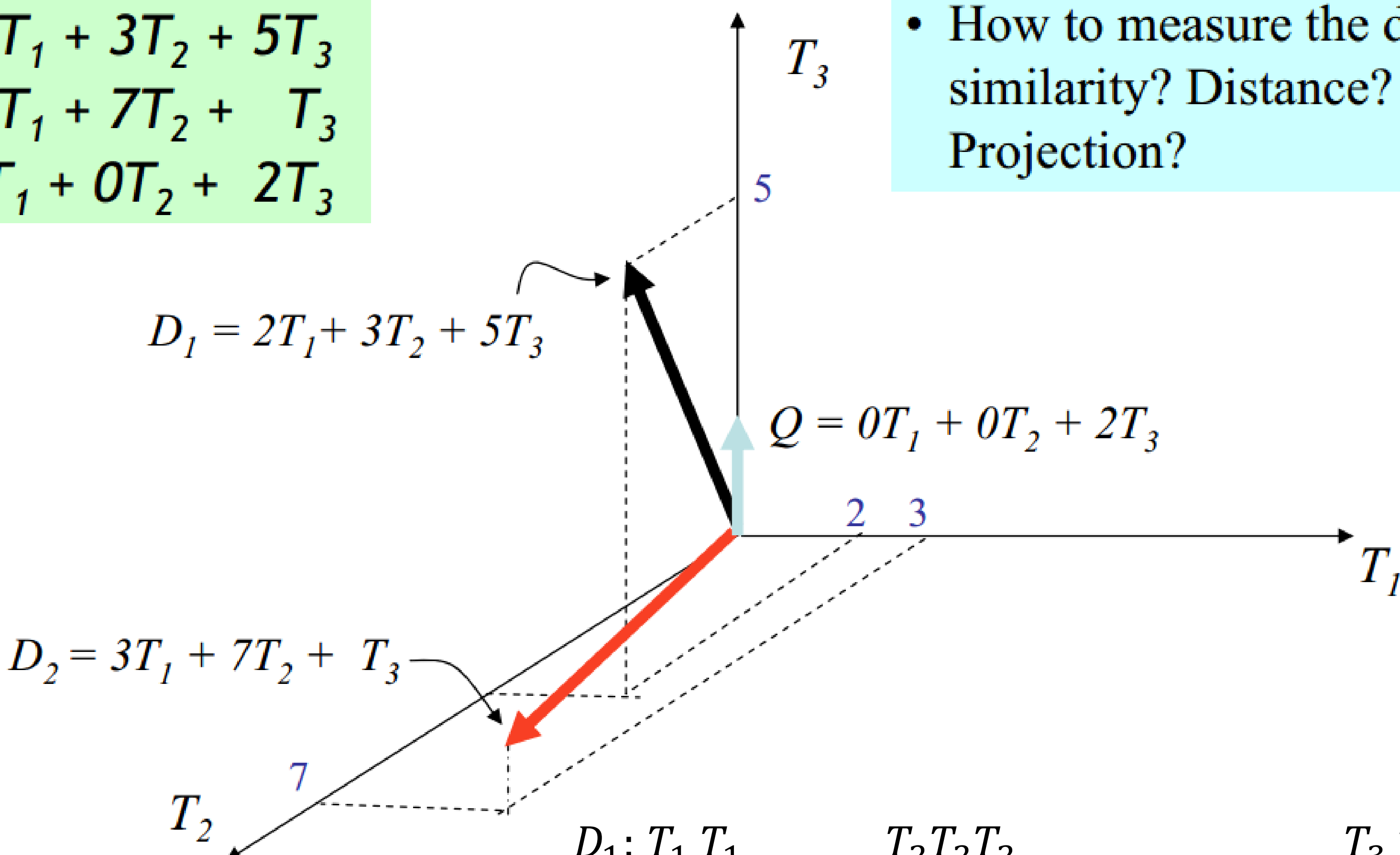
Example:

$D_1 = 2T_1 + 3T_2 + 5T_3$

$D_2 = 3T_1 + 7T_2 + T_3$

$Q = 0T_1 + 0T_2 + 2T_3$

- Is D_1 or D_2 more similar to Q?
- How to measure the degree of similarity? Distance? Angle? Projection?



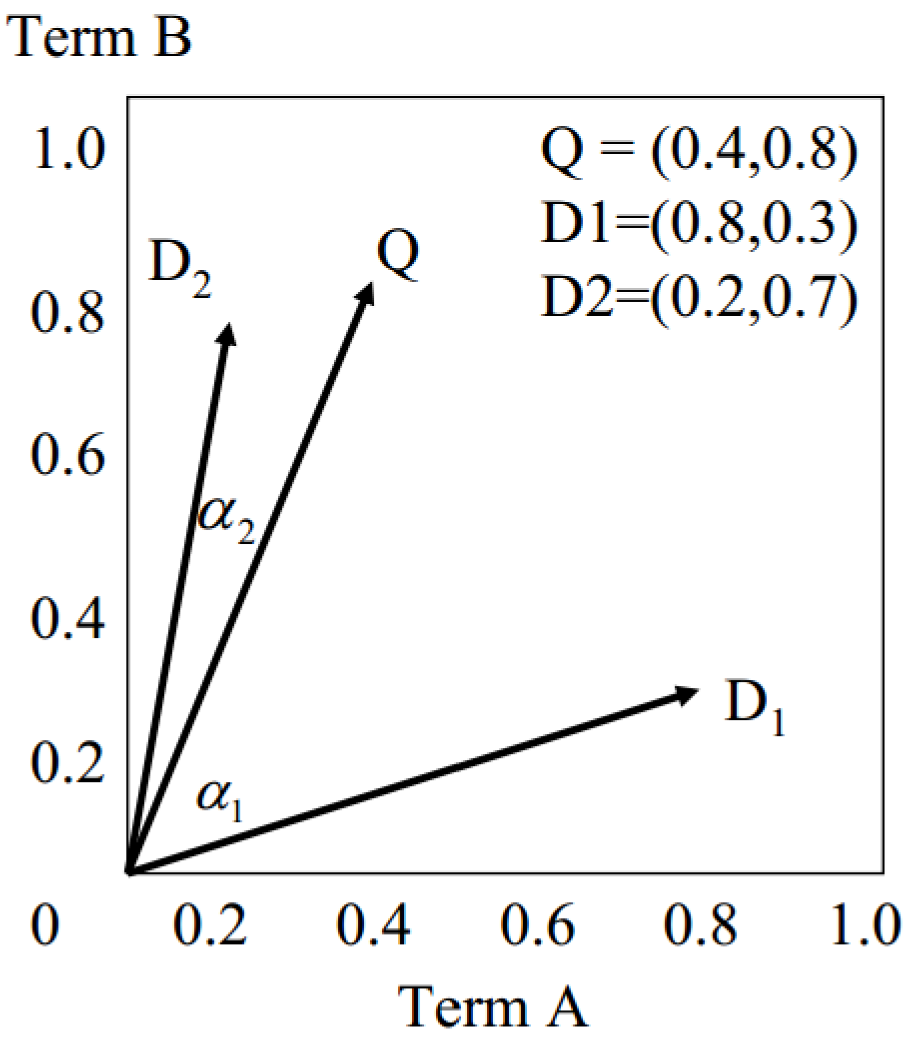
D_1: T_1 T_1 $T_2 T_2 T_2$ T_3 T_3 T_3 T_3 T_3

D_2: T_1 T_1 T_1 T_2 T_2 T_2 T_2 T_2 T_2 T_2 T_3

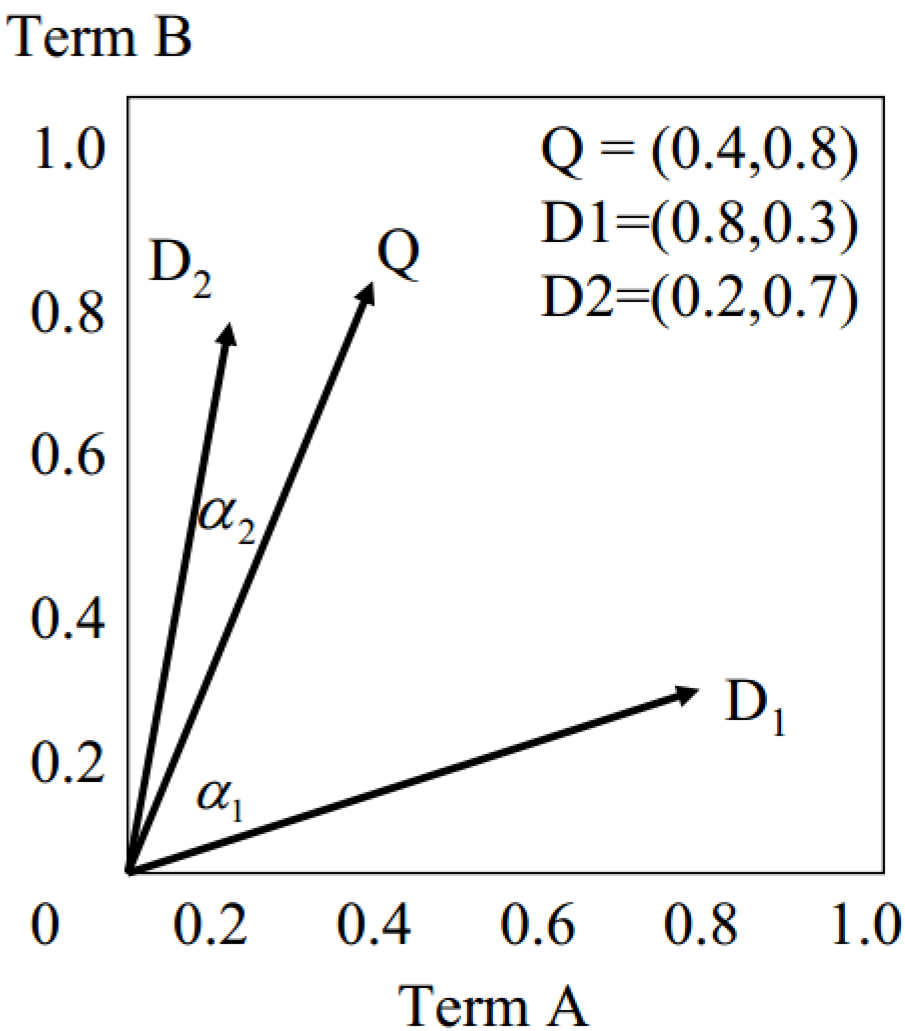
Search query Q : T_3 T_3

Another VSM Model (2D)

Vector Space with Term Weights and Cosine Matching



Vector Space with Term Weights and Cosine Matching



- Which document is most similar to Q?

- Which document is most similar to Q?

Computing Relevance Scores

Ex: Query vector $Q = (0.4, 0.8)$

Also, document $D_2 = (0.2, 0.7)$

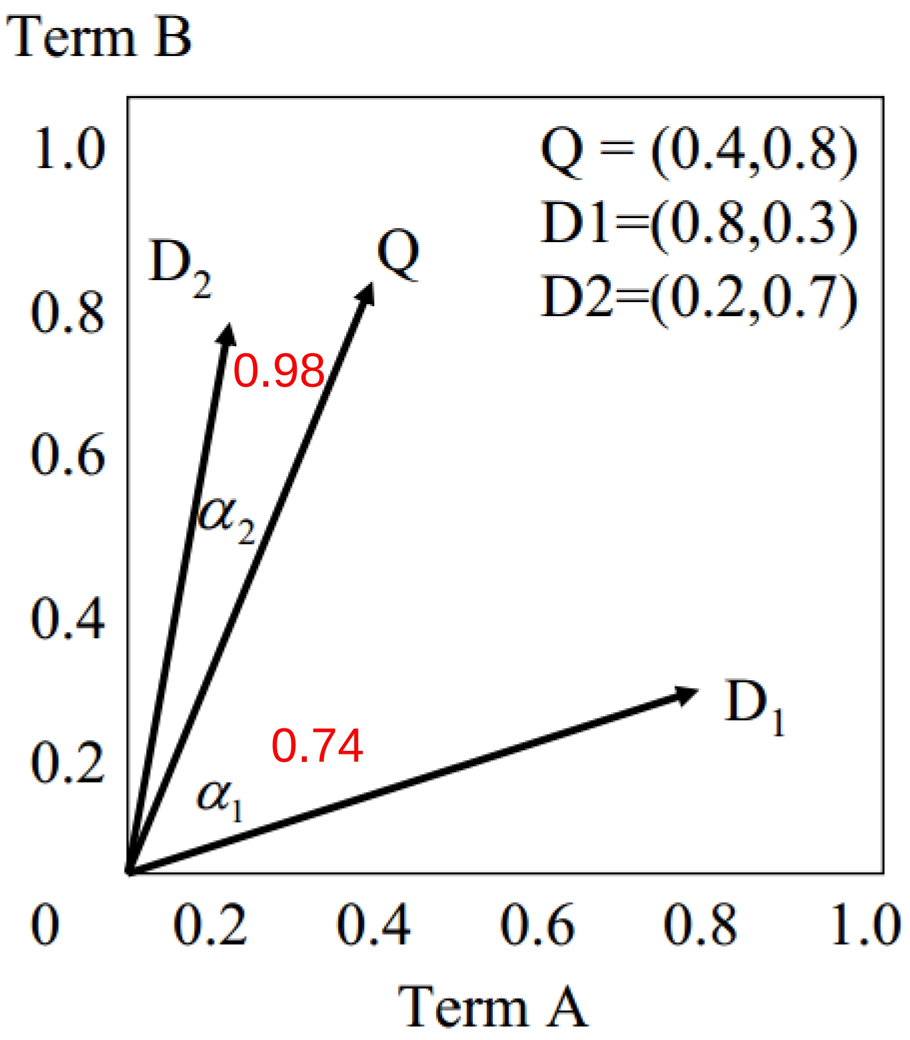
What does their similarity comparison yield?

$$sim(Q, D_i) = \frac{\sum_{j=1}^{t} w_{qj} * w_{d_{ij}}}{\sqrt{\sum_{j=1}^{t} (w_{qj})^2 * \sum_{j=1}^{t} (w_{d_{ij}})^2}}$$

$$sim(Q, D_2) = \frac{(0.4 * 0.2) + (0.8 * 0.7)}{\sqrt{(0.4)^2 + (0.8)^2} * \sqrt{(0.2)^2 + (0.7)^2}}$$

$$= \frac{0.64}{\sqrt{0.42}} = 0.98$$

Vector Space with Term Weights and Cosine Matching

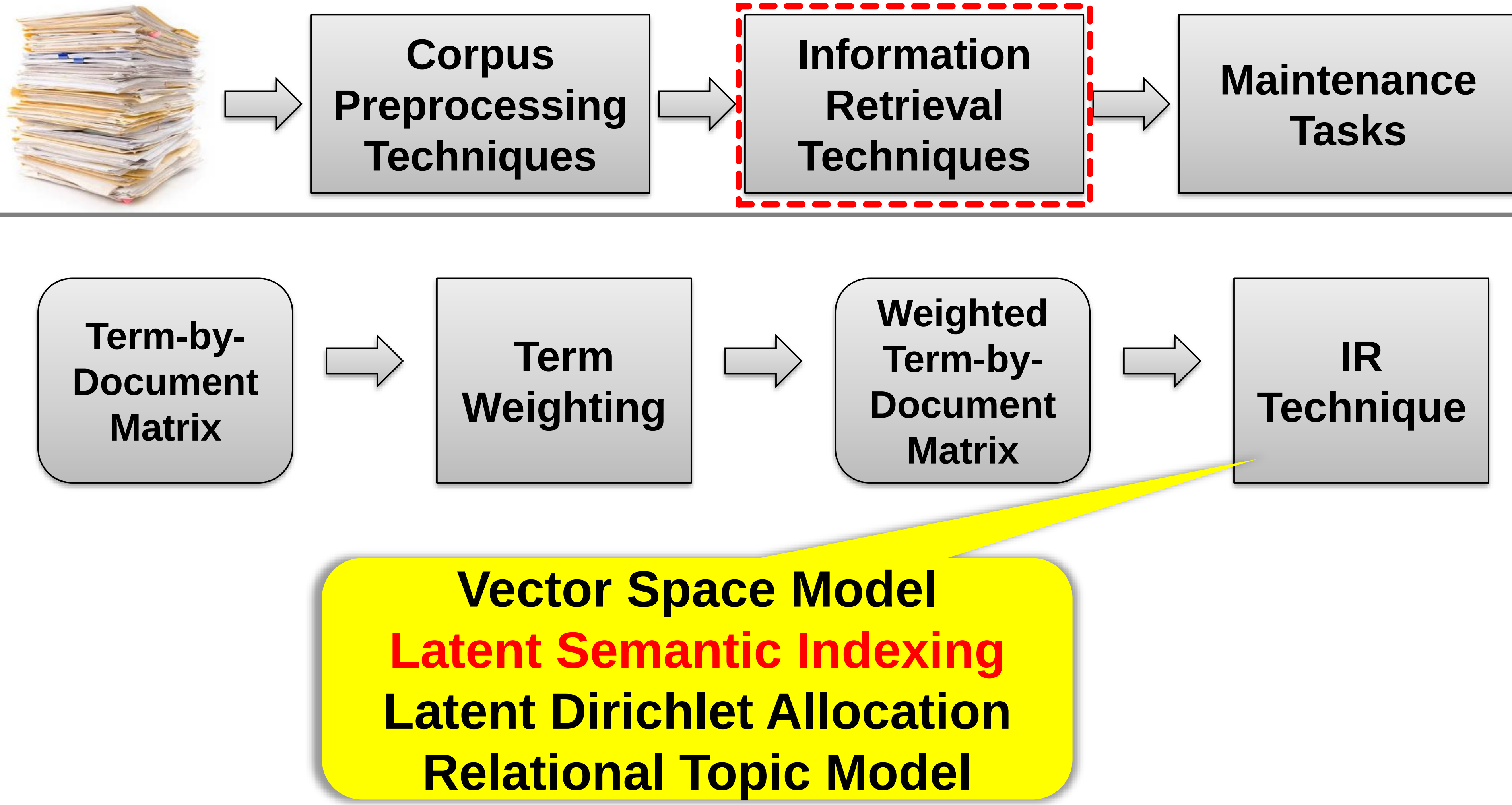


$$sim(Q, D_i) = \frac{\sum_{j=1}^{t} w_{q_j} w_{d_{ij}}}{\sqrt{\sum_{j=1}^{t} (w_{q_j})^2 \sum_{j=1}^{t} (w_{d_{ij}})^2}}$$

$$sim(Q, D2) = \frac{(0.4 \cdot 0.2) + (0.8 \cdot 0.7)}{\sqrt{[(0.4)^2 + (0.8)^2] \cdot [(0.2)^2 + (0.7)^2]}}$$

$$= \frac{0.64}{\sqrt{0.42}} = 0.98$$

$$sim(Q, D_1) = \frac{.56}{\sqrt{0.58}} = 0.74$$


Corpus Preprocessing Techniques
Information Retrieval Techniques
Maintenance Tasks
Term-by-Document Matrix
Term Weighting
Weighted Term-by-Document Matrix
IR Technique
Vector Space Model
Latent Semantic Indexing
Latent Dirichlet Allocation
Relational Topic Model

Why Latent Semantic Indexing (LSI)?

- Some problems for retrieval methods are based on term matching
  - vector-space similarity approach works only if the terms of the query are explicitly present in the relevant documents

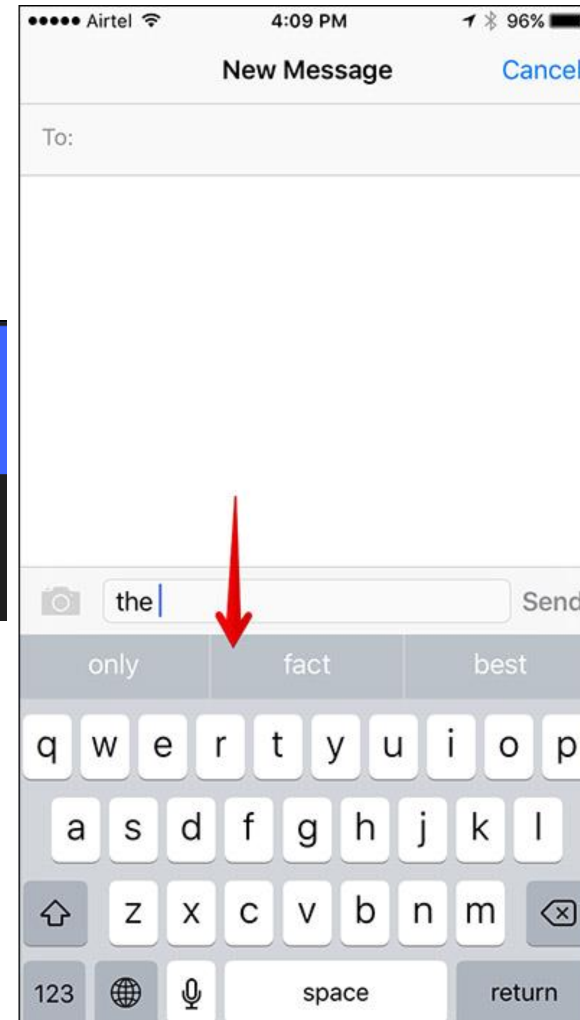
- Rich expressive power of natural language
  - often queries contain terms that express ***concepts*** related to text to be retrieved

Why Latent Semantic Indexing (LSI)?

- With the vector space model, we are assuming independence among terms in a document
  - ... however we know this is not true!!

Why Latent Semantic Indexing (LSI)?

- With the vector space model, we are assuming independence among terms in a document
  - … however we know this is not true!!



Two Problems – Synonyms

- The same concept can be expressed using different sets of terms (synonyms), e.g.,:
  - True, right, correct
  - right place vs. correct position
  - window vs. frame
  - sort vs. order

- Negatively affects recall

Two Problems – Homonyms

- Identical terms can be used in very different semantic contexts (homonyms), e.g.,:
  - bank (financial establishment vs. land near body of water)
  - chip
  - str, id, token

- Negatively affects precision

LSI Idea

- Idea (Deerwester et al.):

"We would like a representation in which a set of terms, which by itself is incomplete and unreliable evidence of the relevance of a given document, is replaced by some other set of entities which are more reliable indicants. We take advantage of the implicit higher-order (or latent) structure in the association of terms and documents to reveal such relationships."

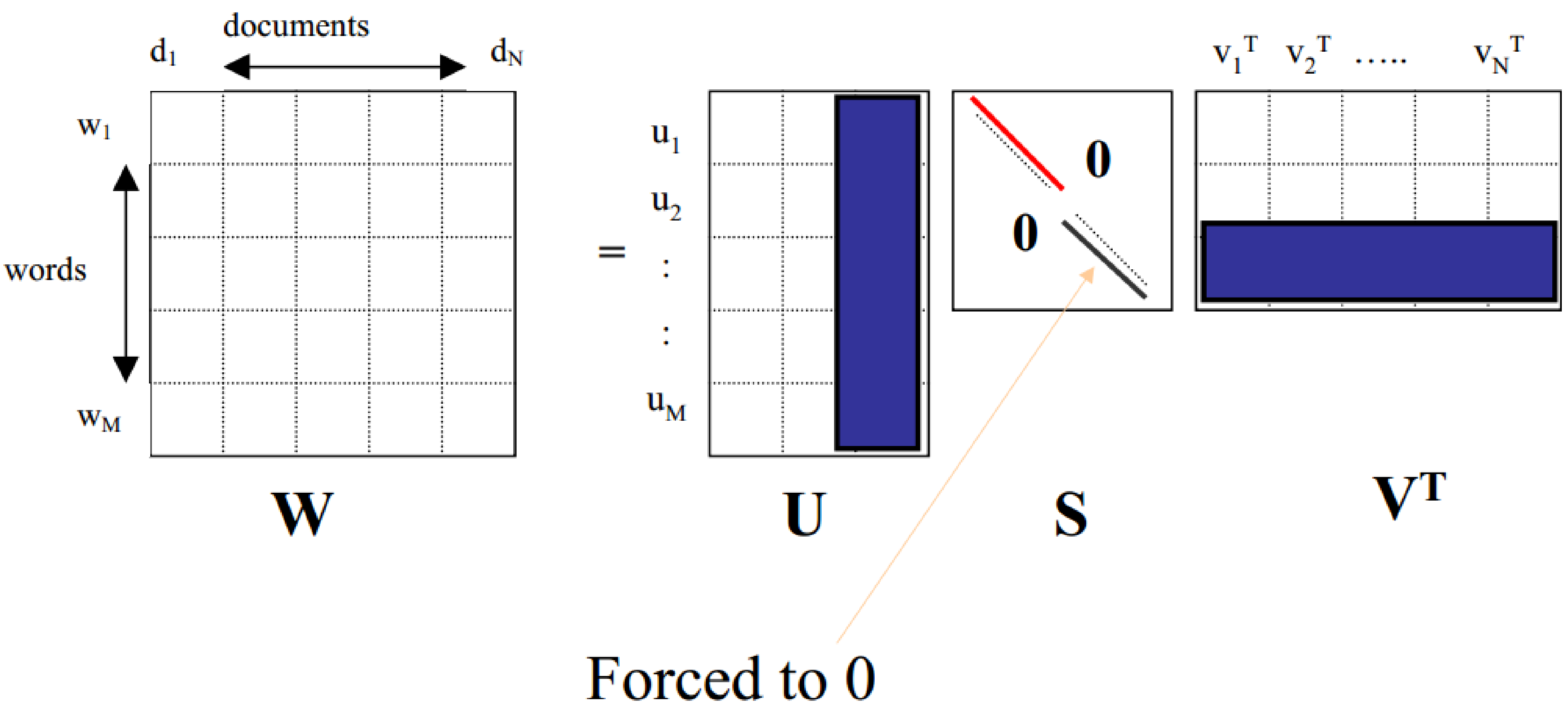
Using SVD

- LSI uses linear algebra technique called singular value decomposition (SVD)
  - attempts to estimate the hidden structure
  - discovers the most important associative patterns between words and concepts
- In other words…
  - The analysis is moved from the space of terms to the space of concepts/topics

Basically…

- Instead of representing documents as a set of correlated factors (terms), we represent documents as set of uncorrelated factors (concepts)

SVD: Dimensionality Reduction



LSI Example

- Given a collection of documents (corpus):

d1: Indian government goes for open-source software
d2: Debian 3.0 Woody released
d3: Wine 2.0 released with fixes for Gentoo 1.4 and Debian 3.0
d4: gnuPOD released: iPOD on Linux… with GPLed software
d5: Gentoo servers running at open-source mySQL database
d6: Dolly the sheep not totally identical clone
d7: DNA news: introduced low-cost human genome DNA chip
d8: Malaria-parasite genome database on the Web
d9: UK sets up genome bank to protect rare sheep breeds
d10: Dolly's DNA damaged

LSI Example

- Given a collection of documents (corpus):

d1: Indian government goes for open-source software
d2: Debian 3.0 Woody released
d3: Wine 2.0 released with fixes for Gentoo 1.4 and Debian 3.0
d4: gnuPOD released: iPOD on Linux… with GPLed software
d5: Gentoo servers running at open-source mySQL database
d6: Dolly the sheep not totally identical clone
d7: DNA news: introduced low-cost human genome DNA chip
d8: Malaria-parasite genome database on the Web
d9: UK sets up genome bank to protect rare sheep breeds
d10: Dolly's DNA damaged

d3 is about (open source) Linux versions…
d8 is about (DNA) manipulation…

LSI Example: term-documents matrix

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10
open-source	1	0	0	0	1	0	0	0	0	0
software	1	0	0	1	0	0	0	0	0	0
Linux	0	0	0	1	0	0	0	0	0	0
released	0	1	1	1	0	0	0	0	0	0
Debian	0	1	1	0	0	0	0	0	0	0
Gentoo	0	0	1	0	1	0	0	0	0	0
database	0	0	0	0	1	0	0	1	0	0
Dolly	0	0	0	0	0	1	0	0	0	1
sheep	0	0	0	0	0	1	0	0	0	0
genome	0	0	0	0	0	0	1	1	1	0
DNA	0	0	0	0	0	0	2	0	0	1

LSI Example:
term-documents matrix

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10
open-source	1	0	0	0	1					
software	1	0	0	1	0					
Linux	0	0	0	1	0					
released	0	1	1	1	0					
Debian	0	1	1	0	0					
Gentoo	0	0	1	0	1					
database	0	0	0	0	1					
Dolly	0	0	0	0	0					
sheep	0	0	0	0	0					
genome	0	0	0	0	0					
DNA	0	0	0	0	0	0	2	0	0	1

d3 is about (open source) Linux versions, but…

LSI Example: term-documents matrix

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10
open-source	1	0	0	0	1					
software	1	0	0	1	0					
Linux	0	0	0	1	0					
released	0	1	1	1	0					
Debian	0	1	1	0	0					
Gentoo	0	0	1	0	1					
database	0	0	0	0	1					
Dolly	0	0	0	0	0					
sheep	0	0	0	0	0					
genome	0	0	0	0	0					
DNA	0	0	0	0	0	0	2	0	0	1

d3 is about (open source) Linux versions, but…

sim(d1,d3)=0

sim(d1,d5)=0.7

LSI Example: term-documents matrix

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10
open-source	1					0	0	0	0	0
software							0	0	0	0
Linux							0	0	0	0
released							0	0	0	0
Debian							0	0	0	0
Gentoo							0	0	0	0
database							0	1	0	0
Dolly							0	0	0	1
sheep							0	0	0	0
genome	0					0	1	1	1	0
DNA	0	0	0	0	0	0	2	0	0	1

d8 is about (DNA) manipulation, but…

LSI Example: term-documents matrix

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10
open-source	1					0	0	0	0	0
software							0	0	0	0
Linux							0	0	0	0
released							0	0	0	0
Debian							0	0	0	0
Gentoo							0	0	0	0
database							0	1	0	0
Dolly							0	0	0	1
sheep							0	0	0	0
genome	0					0	1	1	1	0
DNA	0	0	0	0	0	0	2	0	0	1

d8 is about (DNA) manipulation, but…

sim(d8,d10)=0

sim(d7,d10)=0.63

Reconstructed Term-Document Matrix from the decomposed SVD matrices (k=2)

$$X' = U' * \Sigma' * V'^T$$

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10
open-source	0.119	0.253	0.346	0.26	0.188	0.002	0.05	0.062	0.02	0.017
software	0.149	0.319	0.435	0.328	0.23	-0.01	-0.02	0.054	0.005	-0.02
Linux	0.102	0.22	0.299	0.226	0.158	-0	-0.02	0.035	0.002	-0.02
released	0.337	0.724	0.986	0.744	0.522	-0.01	-0.06	0.118	0.008	-0.05
Debian	0.235	0.505	0.687	0.519	0.364	-0.01	-0.04	0.083	0.006	-0.03
Gentoo	0.208	0.445	0.606	0.457	0.326	-0	0.028	0.092	0.021	0.002
database	0.092	0.187	0.259	0.193	0.159	0.025	0.258	0.111	0.067	0.117
Dolly	-0	-0.02	-0.02	-0.02	0.022	0.044	0.419	0.12	0.098	0.198
sheep	-0	-0	-0.01	-0	0.003	0.008	0.077	0.022	0.018	0.036
genome	0.025	0.014	0.034	0.014	0.107	0.116	1.107	0.329	0.262	0.521
DNA	0.002	-0.06	-0.06	-0.06	0.114	0.19	1.795	0.518	0.422	0.846

Reconstructed Term-Document Matrix from the decomposed SVD matrices (k=2)

$$X' = U' * \Sigma' * V'^T$$

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10
open-source	0.119	0.253	0.346	0.26	0.188	0.002	0.05	0.062	0.02	0.017
software	0.149	0.319	0.435	0.328	0.23	-0.01	-0.02	0.054	0.005	-0.02
Linux	0.102	0.22	0.299	0.226	0.158	-0	-0.02	0.035	0.002	-0.02
released	0.337	0.724	0.986	0.744	0.522	-0.01	-0.06	0.118	0.008	-0.05
Debian	0.235	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	-0.03
Gentoo	0.208	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.002
database	0.092	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.117
Dolly	-0	-0.02	-0.02	-0.02	0.022	0.044	0.419	0.12	0.098	0.198
sheep	-0	-0	-0.01	-0	0.003	0.008	0.077	0.022	0.018	0.036
genome	0.025	0.014	0.034	0.014	0.107	0.116	1.107	0.329	0.262	0.521
DNA	0.002	-0.06	-0.06	-0.06	0.114	0.19	1.795	0.518	0.422	0.846

d3 is related to open-source…

d8 is related to DNA…

LSI: Pros and Cons

+ Able to deal with synonymy and homonymy

+ Stemming could be avoided (but works better with stemming!)

+ Increases similarity between documents of the same cluster

+ Decreases similarity between documents of different clusters

- More expensive than traditional Vector Space Models (SVD computation)

- Difficult to add new documents

- Determining the optimal k is a crucial issue

- Often needs a large document corpus

Concept Location

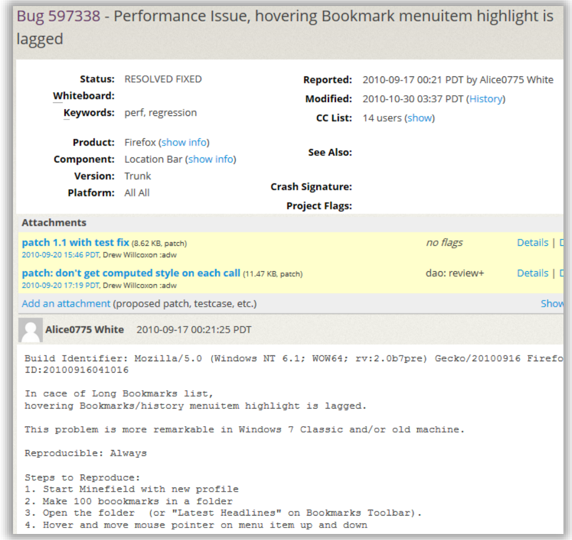
Impact Analysis

Incremental Change

[Rajlich'04]

- Change Request
- Concept Location
- Impact Analysis
- Implementation
- Testing
- Delivery

Bug Report Task



Bug 597338 - Performance Issue, hovering Bookmark menuitem highlight is lagged

Status:	RESOLVED FIXED	Reported:	2010-09-17 00:21 PDT by Alice0775 White
Whiteboard:		Modified:	2010-10-30 03:37 PDT (History)
Keywords:	perf, regression	CC List:	14 users (show)
Product:	Firefox (show info)	See Also:	
Component:	Location Bar (show info)		
Version:	Trunk	Crash Signature:	
Platform:	All All	Project Flags:	

Attachments

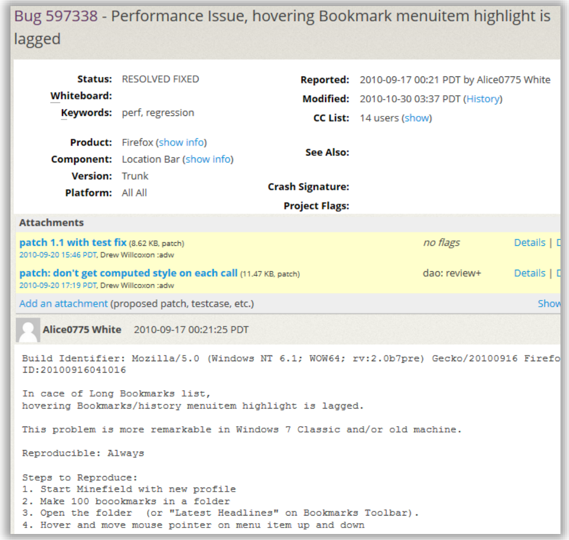
patch 1.1 with test fix (8.62 KB, patch) no flags Details | D
2010-09-20 15:46 PDT, Drew Willcoxon :adw

patch: don't get computed style on each call (11.47 KB, patch) dao: review+ Details | D
2010-09-20 17:19 PDT, Drew Willcoxon :adw

Add an attachment (proposed patch, testcase, etc.) Show

Alice0775 White 2010-09-17 00:21:25 PDT

```
Build Identifier: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:2.0b7pre) Gecko/20100916 Firefo
ID:20100916041016

In cace of Long Bookmarks list,
hovering Bookmarks/history menuitem highlight is lagged.

This problem is more remarkable in Windows 7 Classic and/or old machine.

Reproducible: Always

Steps to Reproduce:
1. Start Minefield with new profile
2. Make 100 booookmarks in a folder
3. Open the folder  (or "Latest Headlines" on Bookmarks Toolbar).
4. Hover and move mouse pointer on menu item up and down
```



Bug 597338 - Performance Issue, hovering Bookmark menuitem highlight is lagged

Status:	RESOLVED FIXED	Reported:	2010-09-17 00:21 PDT by Alice0775 White
Whiteboard:		Modified:	2010-10-30 03:37 PDT (History)
Keywords:	perf, regression	CC List:	14 users (show)
Product:	Firefox (show info)	See Also:	
Component:	Location Bar (show info)		
Version:	Trunk		
Platform:	All All	Crash Signature:	
		Project Flags:	

Attachments

patch 1.1 with test fix (8.62 KB, patch) 2010-09-20 15:46 PDT, Drew Willcoxon :adw — no flags — Details |

patch: don't get computed style on each call (11.47 KB, patch) 2010-09-20 17:19 PDT, Drew Willcoxon :adw — dao: review+ — Details |

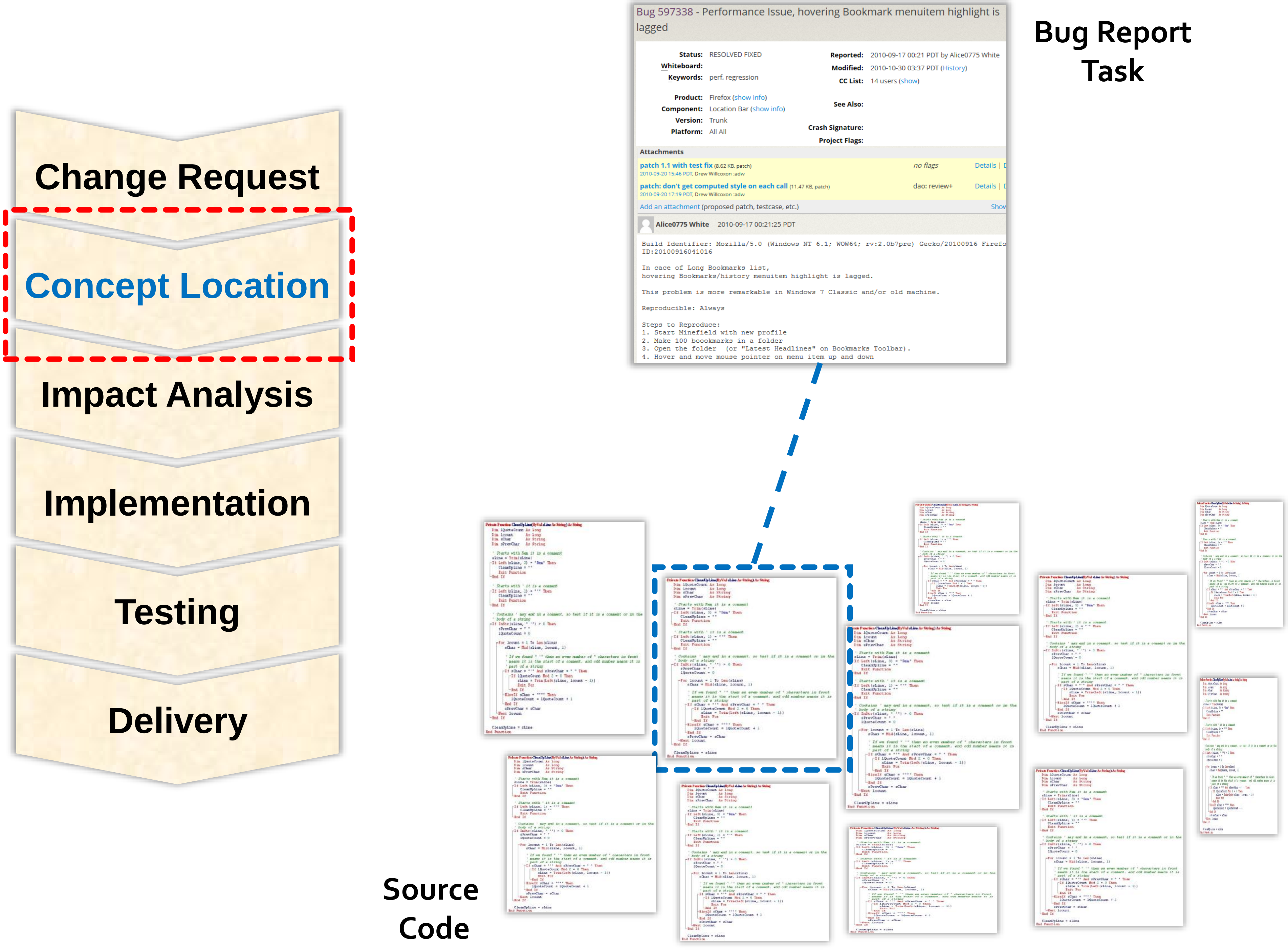
Add an attachment (proposed patch, testcase, etc.) Show

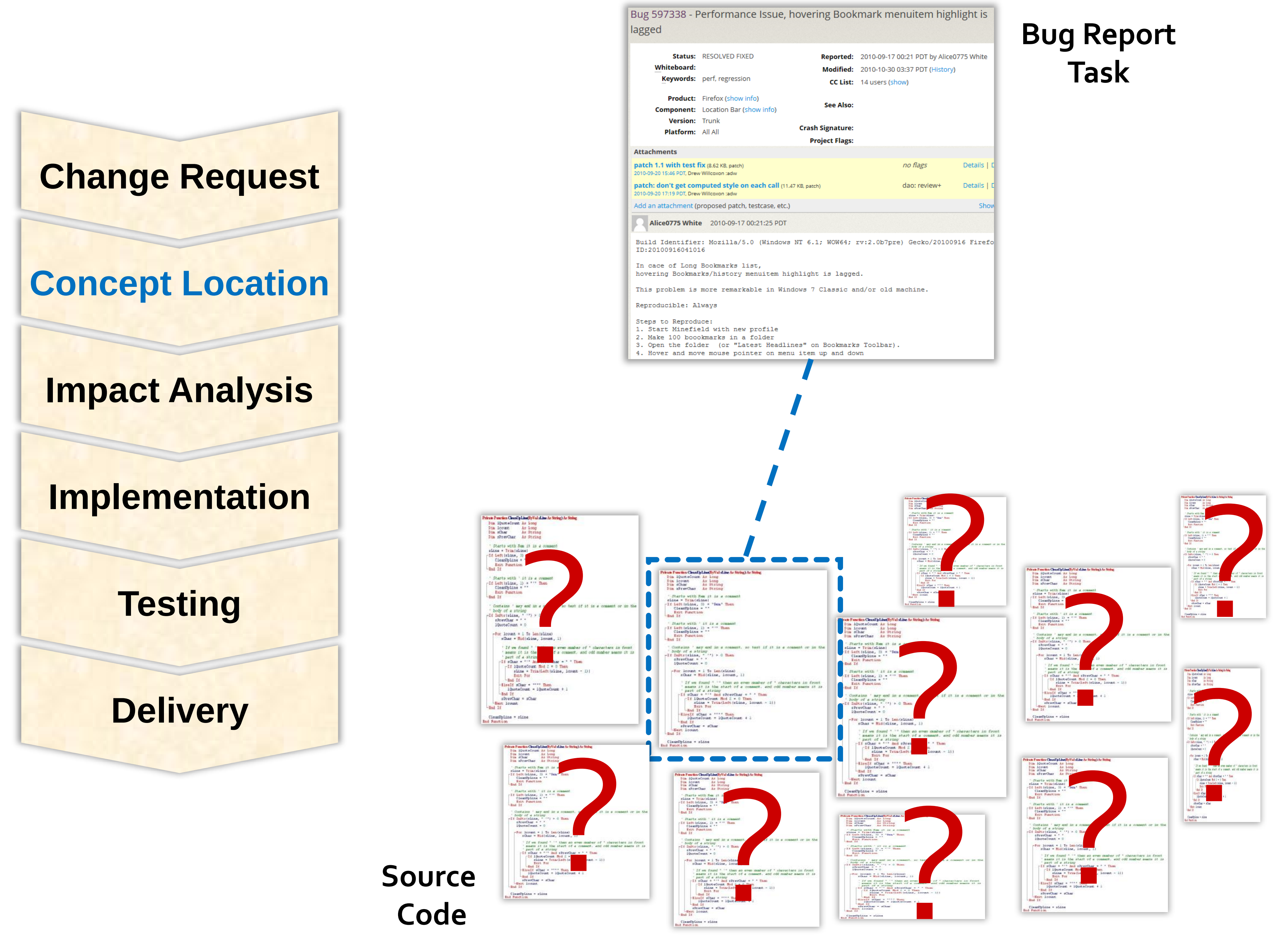
Alice0775 White 2010-09-17 00:21:25 PDT

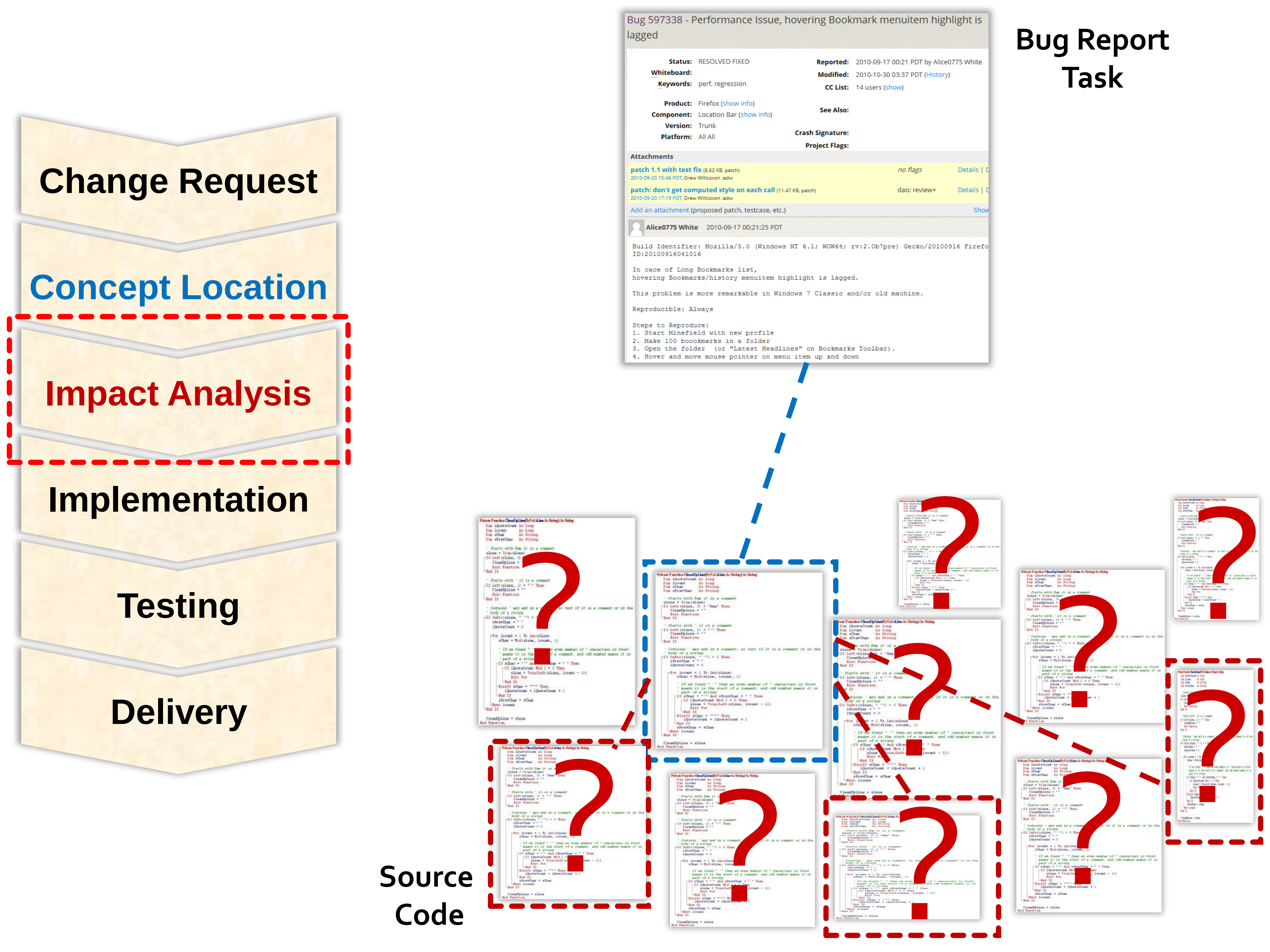
```
Build Identifier: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:2.0b7pre) Gecko/20100916 Firefo
ID:20100916041016

In cace of Long Bookmarks list,
hovering Bookmarks/history menuitem highlight is lagged.

This problem is more remarkable in Windows 7 Classic and/or old machine.

Reproducible: Always

Steps to Reproduce:
1. Start Minefield with new profile
2. Make 100 boookmarks in a folder
3. Open the folder  (or "Latest Headlines" on Bookmarks Toolbar).
4. Hover and move mouse pointer on menu item up and down
```

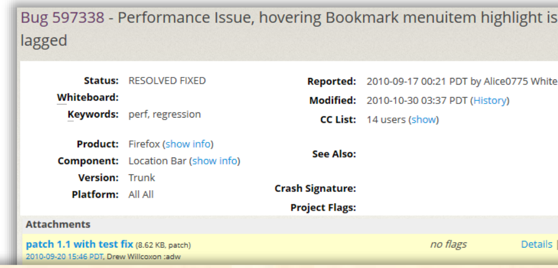
Bug Report Task

Source Code


Change Request
Concept Location
Impact Analysis
Implementation
Testing
Delivery
Bug Report Task
Bug 597338 - Performance Issue, hovering Bookmark menuitem highlight is lagged
Status: RESOLVED FIXED
Whiteboard:
Keywords: perf, regression
Reported: 2010-09-17 00:21 PDT by Alice0775 White
Modified: 2010-10-30 03:37 PDT (History)
CC List: 14 users (show)
Product: Firefox (show info)
Component: Location Bar (show info)
Version: Trunk
Platform: All All
See Also:
Crash Signature:
Project Flags:
Attachments
patch 1.1 with test fix (8.62 KB, patch)
2010-09-20 15:46 PDT, Drew Willcoxon :adw
no flags
Details
patch: don't get computed style on each call (11.47 KB, patch)
2010-09-20 17:19 PDT, Drew Willcoxon :adw
dao: review+
Details
Add an attachment (proposed patch, testcase, etc.)
Alice0775 White 2010-09-17 00:21:25 PDT
Build Identifier: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:2.0b7pre) Gecko/20100916
ID:20100916041016
In cace of Long Bookmarks list,
hovering Bookmarks/history menuitem highlight is lagged.
This problem is more remarkable in Windows 7 Classic and/or old machine.
Reproducible: Always
Steps to Reproduce:
1. Start Minefield with new profile
2. Make 100 boookmarks in a folder
3. Open the folder (or "Latest Headlines" on Bookmarks Toolbar).
4. Hover and move mouse pointer on menu item up and down
Source Code


Change Request
Concept Location
Impact Analysis
Implementation
Testing
Delivery
Bug Report Task
Bug 597338 - Performance Issue, hovering Bookmark menuitem highlight is lagged
Status: RESOLVED FIXED
Whiteboard:
Keywords: perf, regression
Product: Firefox (show info)
Component: Location Bar (show info)
Version: Trunk
Platform: All All
Reported: 2010-09-17 00:21 PDT by Alice0775 White
Modified: 2010-10-30 03:37 PDT (History)
CC List: 14 users (show)
See Also:
Crash Signature:
Project Flags:
Attachments
patch 1.1 with test fix (8.62 KB, patch)
2010-09-20 15:46 PDT, Drew Willcoxon :adw
no flags
Details
patch: don't get computed style on each call (11.47 KB, patch)
2010-09-20 17:19 PDT, Drew Willcoxon :adw
dao: review+
Details
Add an attachment (proposed patch, testcase, etc.)
Alice0775 White 2010-09-17 00:21:25 PDT
Build Identifier: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:2.0b7pre) Gecko/20100916 Firefo
ID:20100916041016
In cace of Long Bookmarks list,
hovering Bookmarks/history menuitem highlight is lagged.
This problem is more remarkable in Windows 7 Classic and/or old machine.
Reproducible: Always
Steps to Reproduce:
1. Start Minefield with new profile
2. Make 100 boookmarks in a folder
3. Open the folder (or "Latest Headlines" on Bookmarks Toolbar).
4. Hover and move mouse pointer on menu item up and down
Source Code


Change Request
Concept Location
Impact Analysis
Implementation
Testing
Delivery
Bug Report Task
Bug 597338 - Performance Issue, hovering Bookmark menuitem highlight is lagged
Status: RESOLVED FIXED
Whiteboard:
Keywords: perf, regression
Reported: 2010-09-17 00:21 PDT by Alice0775 White
Modified: 2010-10-30 03:37 PDT (History)
CC List: 14 users (show)
Product: Firefox (show info)
Component: Location Bar (show info)
Version: Trunk
Platform: All All
See Also:
Crash Signature:
Project Flags:
Attachments
patch 1.1 with test fix (8.62 KB, patch)
2010-09-20 15:46 PDT, Drew Willcoxon :adw
no flags
patch: don't get computed style on each call (11.47 KB, patch)
2010-09-20 17:19 PDT, Drew Willcoxon :adw
dao: review+
Add an attachment (proposed patch, testcase, etc.)
Alice0775 White 2010-09-17 00:21:25 PDT
Build Identifier: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:2.0b7pre) Gecko/20100916 Firefo
ID:20100916041016
In cace of Long Bookmarks list,
hovering Bookmarks/history menuitem highlight is lagged.
This problem is more remarkable in Windows 7 Classic and/or old machine.
Reproducible: Always
Steps to Reproduce:
1. Start Minefield with new profile
2. Make 100 boookmarks in a folder
3. Open the folder (or "Latest Headlines" on Bookmarks Toolbar).
4. Hover and move mouse pointer on menu item up and down
Source Code



Bug Report

Change Request

Concept Location

Impact Analysis

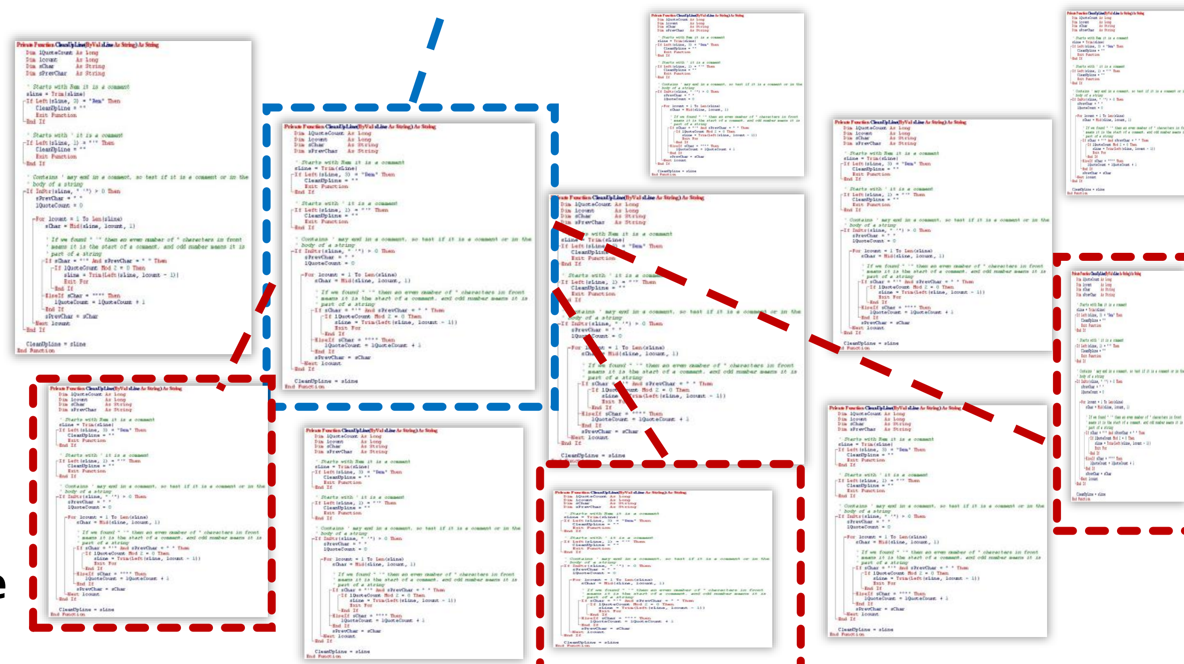
Implementation

Testing

Delivery

Extremely challenging tasks:

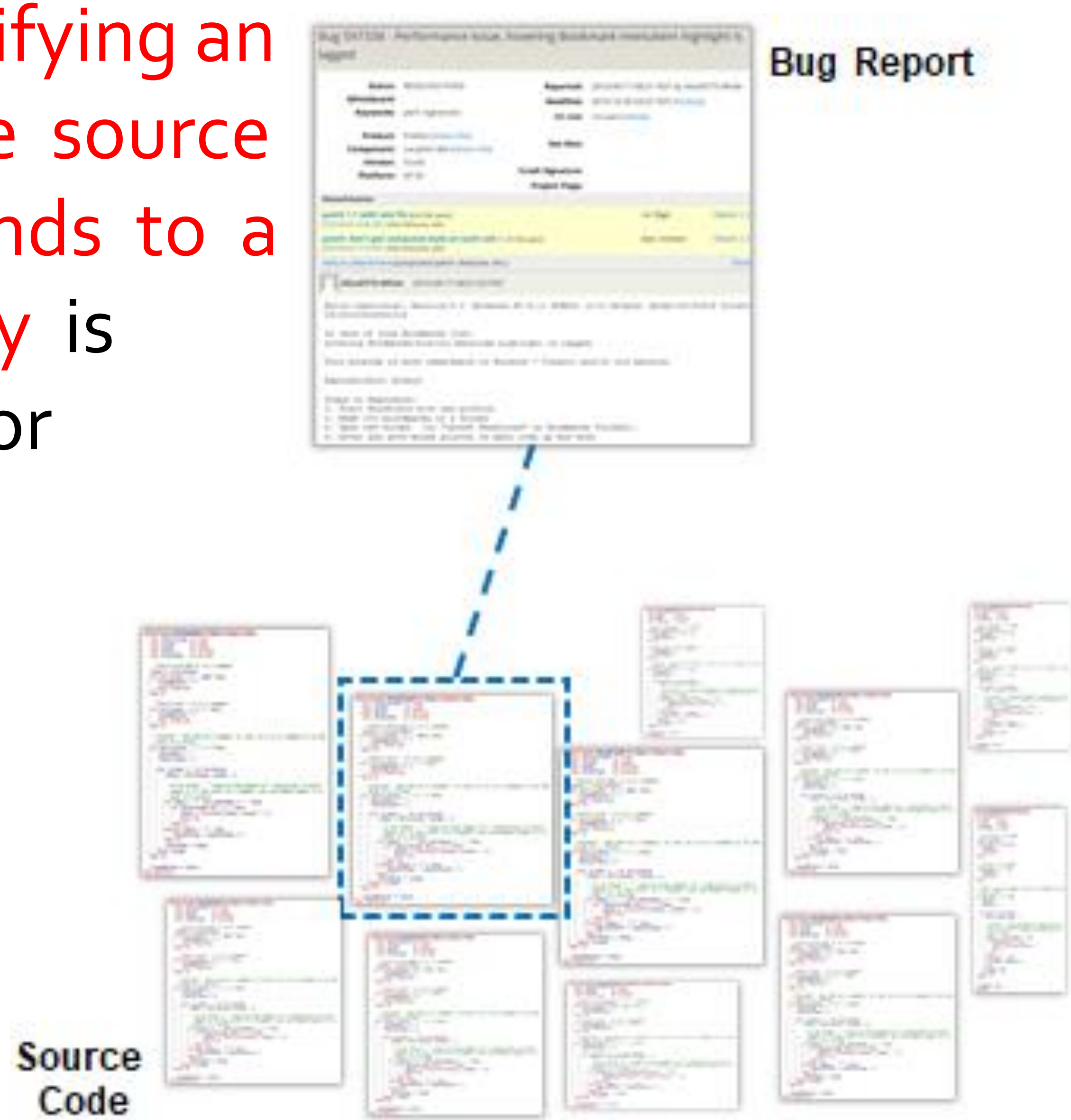
- **Millions of Lines of Code**
- **Absence of original developer**
- **Missing or outdated documentation**
- **etc.**



Source Code

Concept (Feature) Location “Definition”*

- The activity of identifying an initial location in the source code that corresponds to a specific functionality is known as concept (or feature) location



*[Biggerstaff'94, Rajlich'02]

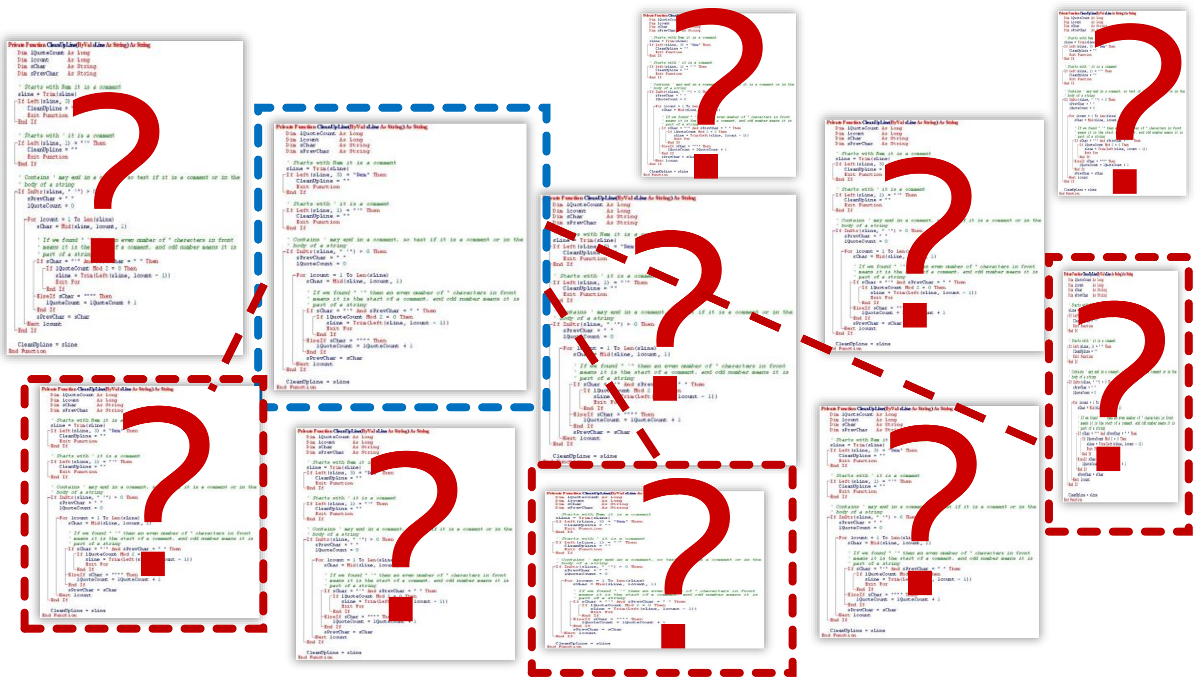
Concept (Feature) Location

- Concept location is needed whenever a change is to be made

- Change requests are most often formulated in terms of domain concepts
  - example: “Correct error that arises when trying to paste a text”
  - the programmer must find in the code the locations where concept “paste” is located
  - this is the start of the change

Impact Analysis "Definition"*

- The activity of:
  - estimating what needs to be modified to accomplish a change, or
  - identifying the potential consequences of a change



*Bohner and Arnold 1996

Strategies for Concept Location

Strategies for Concept Location: Familiarize with the software

- User manual/documentation
- Domain knowledge artifacts (e.g., medical process)
- Design documents and diagrams

Strategies for Concept Location: Software Structure Comprehension

- Developer resources/wiki
- Prerequisites/dependencies
- Development workflow
- Code organization/structure and architecture
- Testing infrastructure
- Feedback channels
- Style guidelines
- APIs
- etc.

Strategies for Concept Location: Software Build/Run

- Build/compile/resolve dependencies
- Execute the software
  - Try different scenarios/features
- Examine log files (if any)

Strategies for Concept Location: Examine Source Code

- Search for concepts/features
- Read existing code
  - Build a mental model of the software
- Navigate dependencies
- Browse and examine code in specific locations
- Use traceability links
- Ask existing developers familiar with the code