

Introduction of General Text Parser

顏永泰
臺北醫學大學
醫學資訊研究所

Reference:

Latent Semantic Indexing Web Site <http://www.cs.utk.edu/~lsi/>

What is General Text Parser (GTP)?

- ❑ Object-oriented (C++, Java) integrated software package.
- ❑ Designed for Text Mining.
- ❑ Developed by S. Howard *et al.* at the University of Tennessee (Department of Computer Science).



Model Facilitated by GTP

- Tag filter to perform text parsing.
 - ASCII text, HTML, PostScript, PDF.
- Create sparse term-by-document matrices.
- Latent Semantic Indexing (LSI).
 - Singular Value Decomposition (SVD).
 - Semi Discrete Decomposition (SDD).



GTP Requirement

- ❑ Solaris 5.7 using **gcc** version 2.95.3.
- ❑ RedHat Linux 2.2.16-22 using **gcc** version 2.96.
- ❑ Javac, **JVM 1.4**.



Testing Environment

- Mandrake 9.1
- Microsoft Windows XP
- Java(TM) 2 Standard Edition 1.4.2



How to Get GTP?

- ❑ Fill out the request form.
 - <http://www.cs.utk.edu/~Isi/gtp-request.html>
- ❑ Wait for the instruction email.
- ❑ Download the encrypted file.
- ❑ Decrypt and decompress.
 - `crypt key < input file > output file`
 - `tar xzfv gtp.v4.0.tar.gz`



How to Run GTP? (Java Version)

- Install Java VM.
 - <http://java.sun.com>
- Modify shell profile to set CLASSPATH.
 - CLASSPATH=./usr/local/GTP
 - export CLASSPATH
- Two ways to run GTP
 - ../GTP/gtp/run/java gtp *parameter*
 - ../GTP/gui/java GTPgui



How to Run GTP? (cont.)

- `java GTP ../sample -c ../etc/common_words -t ./tmp -h -z svd1 sample -d 0 -g 0 -O -w log entropy`
 - `../sample`: text file or directory.
 - `-c`: assign a `common_words` (stop words) list.
 - `-t`: assign a temporary directory.



How to Run GTP? (cont.)

- `java GTP ../sample -c ../etc/common_words -t ./tmp -h -z svd1 sample -d 0 -g 0 -O -w log entropy`
 - `-h`: the Harwell-Boeing compressed matrix.
 - `-z svd`: semi-discrete decomposition.
(require `-h`)
 - `-z svd1`: singular value decomposition.
(require `-h`)
 - `sample`: a svd(1) name for description.



How to Run GTP? (cont.)

- `java GTP ../sample -c ../etc/common_words -t ../tmp -h -z svd1 sample -d 0 -g 0 -O -w log entropy`
- *-d*: threshold for local frequency of any term.
 - *-d 2*: a term must occur more than twice in the document.
- *-g*: threshold for global frequency of any term.
 - *-g 2*: a term must occur more than twice in an entire document collection.



How to Run GTP? (cont.)

- `java GTP ../sample -c ../etc/common_words -t ./tmp -h -z svd1 sample -d 0 -g 0 -O -w log entropy`
 - `-O`: output file is to be in one binary file for SVD. (if need to use GTPQUERY).
 - `-w`: specify a custom weighting scheme.



Weighting Scheme

□ -w *local global*

■ local weighting: *tf, log, binary*.

□ *tf*: the collection spans general topics.

□ *log*: $1 \div \sqrt{\log(1+tf)^2}$

□ *binary*: Only the presence (1) or absence (0) of a term is included in the vector.

■ the collection is small with few terms in the vocabulary.



Weighting Scheme (cont.)

- global weighting: none, normal, idf, idf2, entropy.
 - idf: the collection is static.
 - idf2: $\log(ndocs) / \log 2 - \log(df + 1) / \log 2$
 - entropy: consider to take out noises.



Generation of GTP

□ output

- A binary file contains all the vector (term and document) and singular value information produced by the SVD.
- Transfer to ASCII file.
 - `java gtp.ReadOutput output > output.ascii`



Content of output.ascii

Header

terms = 10
docs = 3
factors = 3
commentSize = 20
updatedTerms = 0 * not activated *
updatedDocs = 0 * not activated *
comment = Creating output file

Output file continued ...

Term Vectors

Term 0 (conference)

0	0.3908561766
1	-0.3573666215
2	-0.1973721683

Term 9 (text)

0	0.3908561766
1	0.3573666215
2	0.1973721683



Content of output.ascii (cont.)

Document Vectors

Document 0 (Numerical Libraries and the Grid)

0 0.0000000000
1 -0.4834610820
2 0.8753658533

...

Document 2 (GTP: Software for Text Mining)

0 0.7071067691
1 0.6189771295
2 0.3418586254

Singular Values

0 1.2537220716
1 1.2003111839
2 1.2003111839



Generation of keys

□ keys

- a database of all the terms that were parsed in the GTP run.

Key	ID	Global Weight
Conference	1	1.000000
Grid	2	1.000000
Gtp	3	1.000000
Libraries	4	1.000000
Mining	5	1.000000
Numerical	6	1.000000
Organizer	7	1.000000
Semantic	8	1.000000
Software	9	0.369070
Text	10	1.000000



Content of rawmatrix

- the term-by-document matrix.

```
7 161
doc 1:
123 1.0000 80 3.0000 10 1.0000 30 1.0000 150 1.0000 144 1.0000 83 2.0000 89
1.0000 24 1.0000 53 1.0000 62 1.0000 9 1.0000 25 1.0000 60 2.0000 64 2.0000
51 1.0000 142 1.0000 110 1.0000 34 1.0000 140 1.0000 54 1.0000 159 1.0000
147 2.0000 66 1.0000 118 1.0000 155 1.0000 161 1.0000 129 1.0000 49 1.0000
153 1.0000 47 1.0000
doc2:
```



Query Processing

- ❑ Cosine similarity measure between a query vector and document vectors.
- ❑ `java GTPQUERY queryfile -c ../etc/common_words -S`
- ❑ Require **output**, **keys**, **LAST_RUN**.
- ❑ -S: Scale the query vector by the singular values before calculating cosine similarity.



Query Processing (cont.)

- Each query is separated by a blank line.

Numerical Software

Text Mining Software

- Each result file has a prefix of `q_result.#`



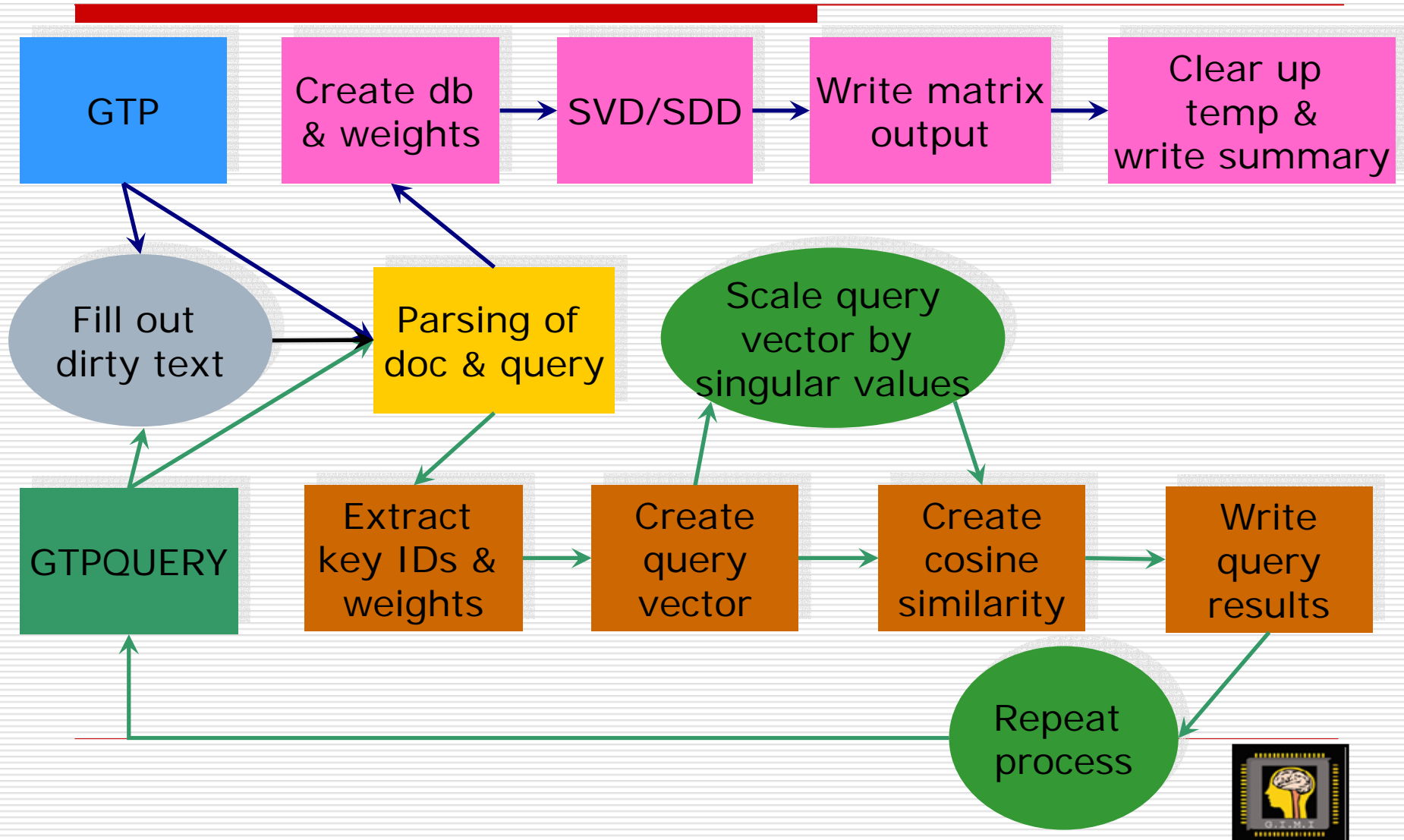
Content of A Result File

- A list of document IDs and cosine similarity measures.

Doc ID	Cos. Similarity
50	0.975631
2	0.756432
14	0.500123
...	...



Flowchart of GTP & GTPQUERY



How to Run GTP in MS Windows?

□ Set Classpath

- Set classpath=c:\gtp\

□ Run GTP

- C:\gtp\run\
C:\gtp\etc\common_words -t
C:\gtp\tmp\ -h -z svd1 sample -d 0 -g 0
-O -w log entropy

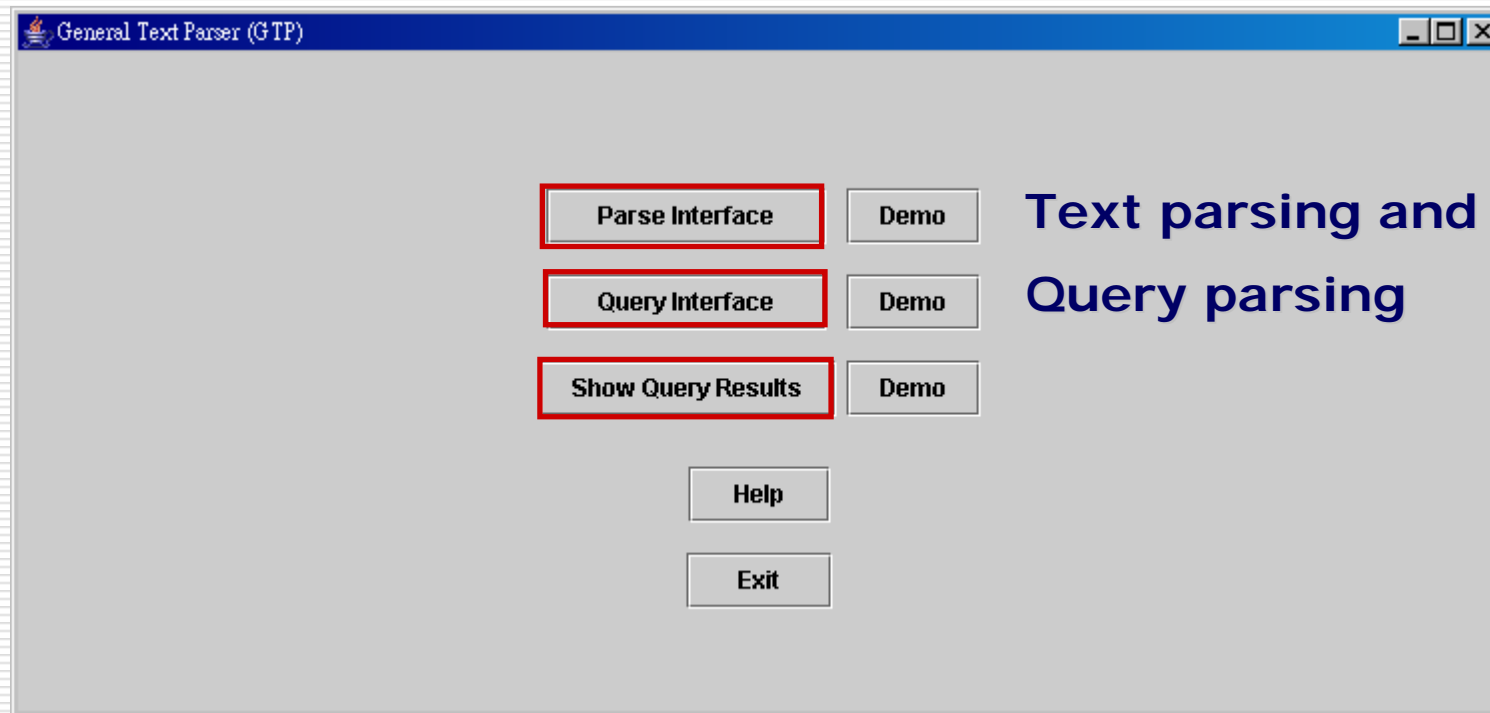


How to Run GTP with GUI?

- Copy `./GTP/gtp/GTP.class`
& `./GTP/query/GTPQUERY.class`
to `./GTP/gui`
- `./GTP/gui/java GTPgui`



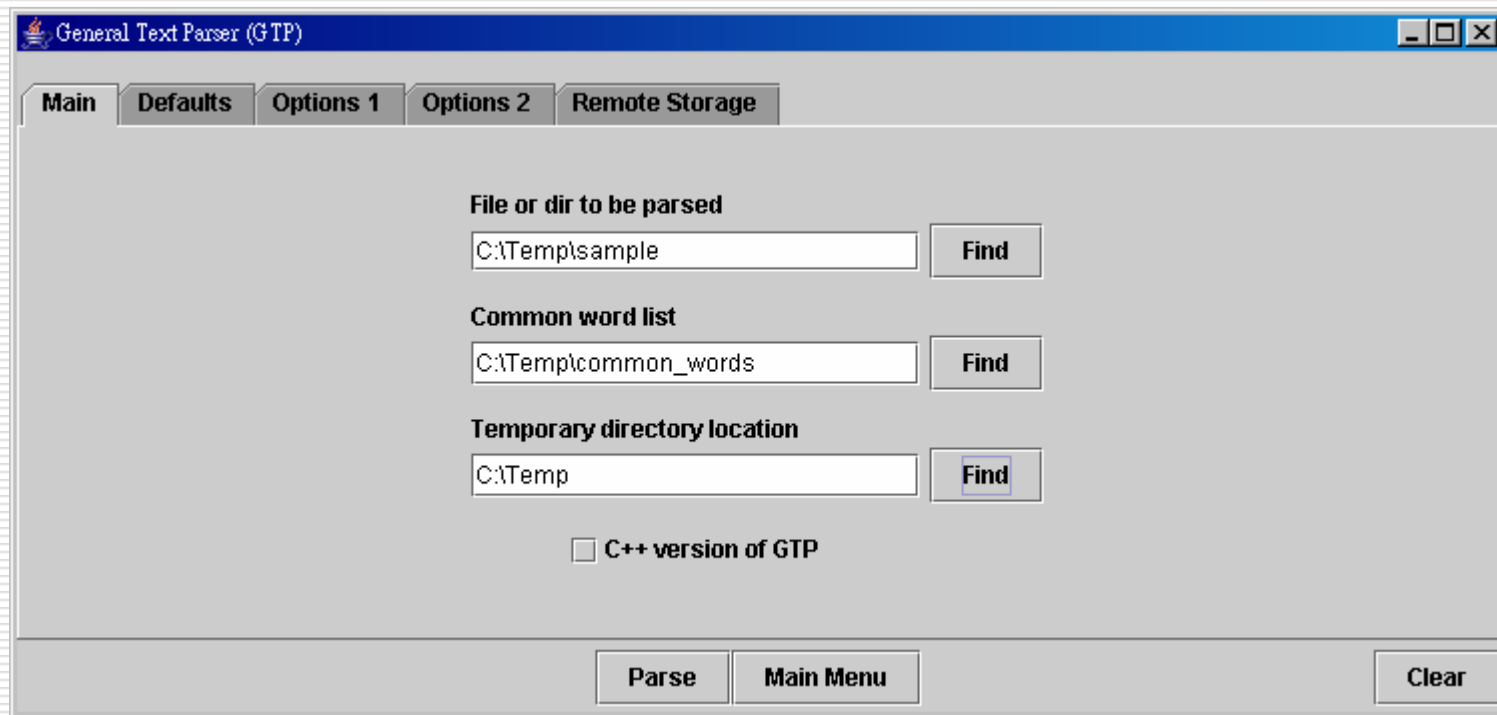
3 Parts of GTP GUI



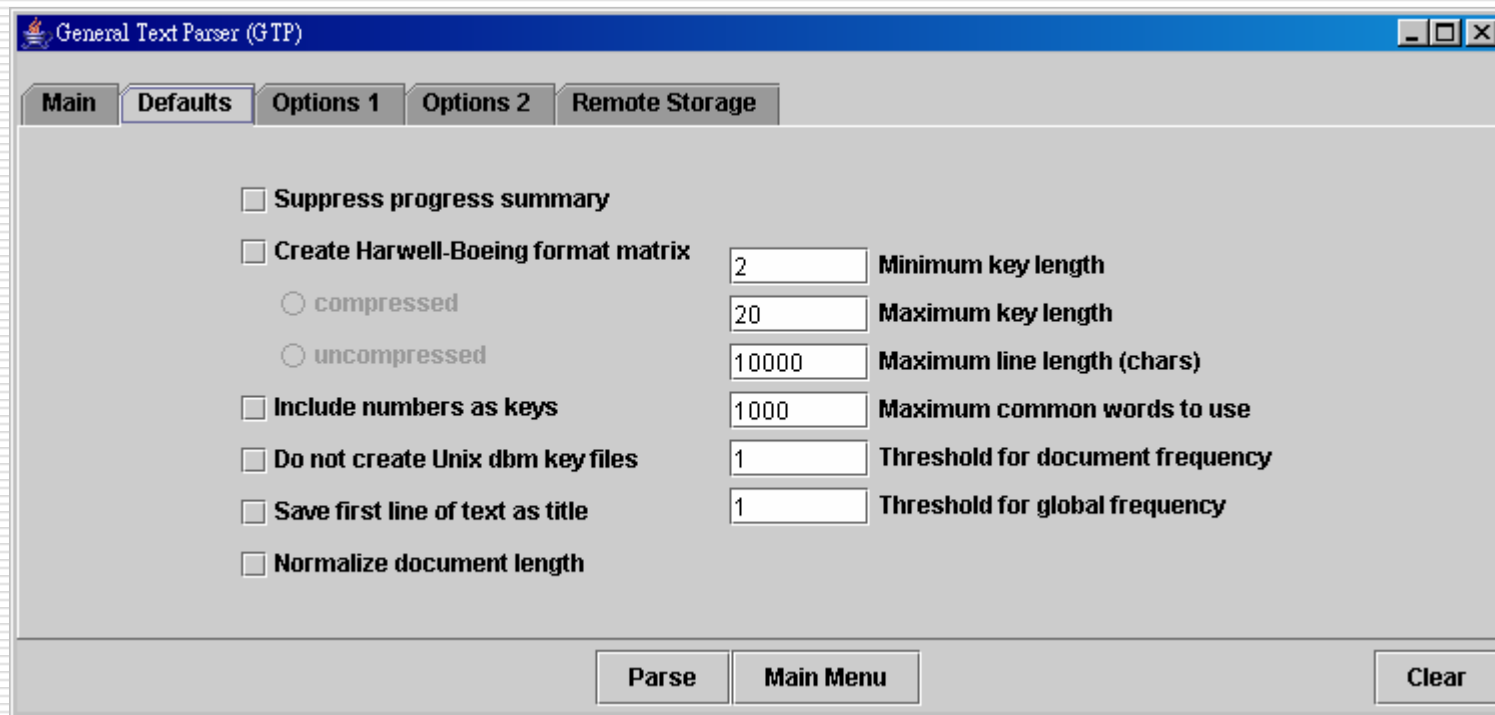
Text parsing and SVD/SDD
Query parsing



Parser Interface



Parser Interface (cont.)



The screenshot shows the 'General Text Parser (GTP)' window with the 'Defaults' tab selected. The interface includes several checkboxes and input fields for configuration. At the bottom, there are 'Parse', 'Main Menu', and 'Clear' buttons.

Option	Value	Description
<input type="checkbox"/> Suppress progress summary		
<input type="checkbox"/> Create Harwell-Boeing format matrix	2	Minimum key length
<input type="radio"/> compressed	20	Maximum key length
<input type="radio"/> uncompressed	10000	Maximum line length (chars)
<input type="checkbox"/> Include numbers as keys	1000	Maximum common words to use
<input type="checkbox"/> Do not create Unix dbm key files	1	Threshold for document frequency
<input type="checkbox"/> Save first line of text as title	1	Threshold for global frequency
<input type="checkbox"/> Normalize document length		



Parser Interface (cont.)

General Text Parser (GTP)

Main Defaults Options 1 Options 2 Remote Storage

Specify Local weighting scheme **log** Global weighting scheme **entropy**

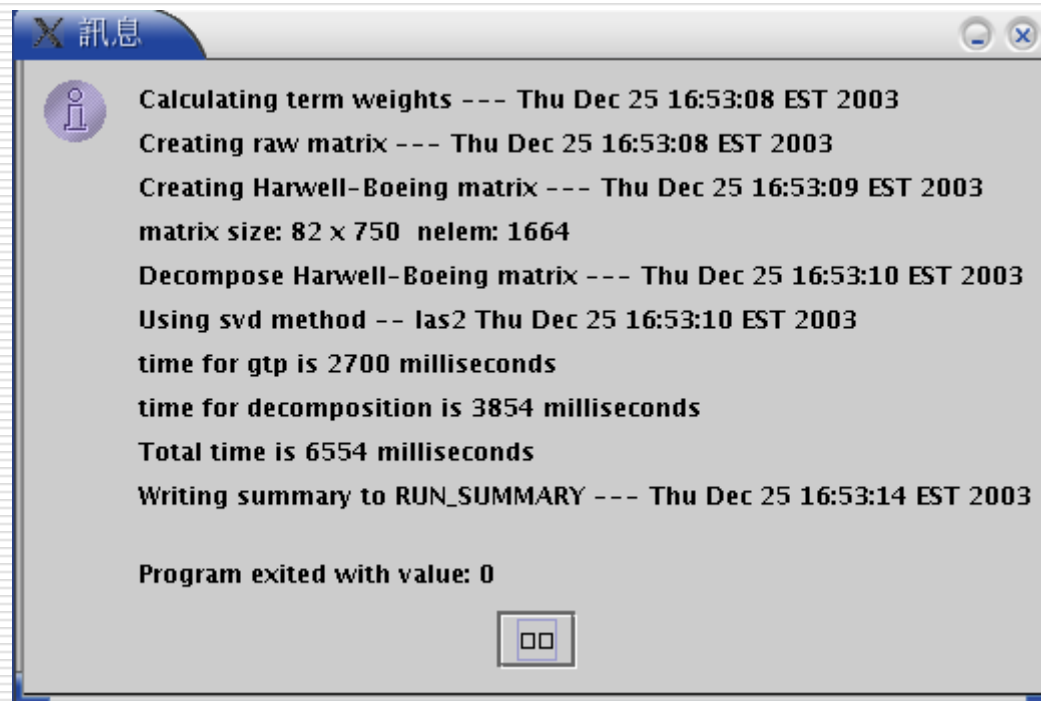
Specify the decomposition method
(sdd rank inner_loop_criteria tolerance) or (svd1 desc lanmax maxprs)

Skip parse procedure so matrix can be decomposed via SVD or SDD

Parse Main Menu Clear



Parser Interface (cont.)



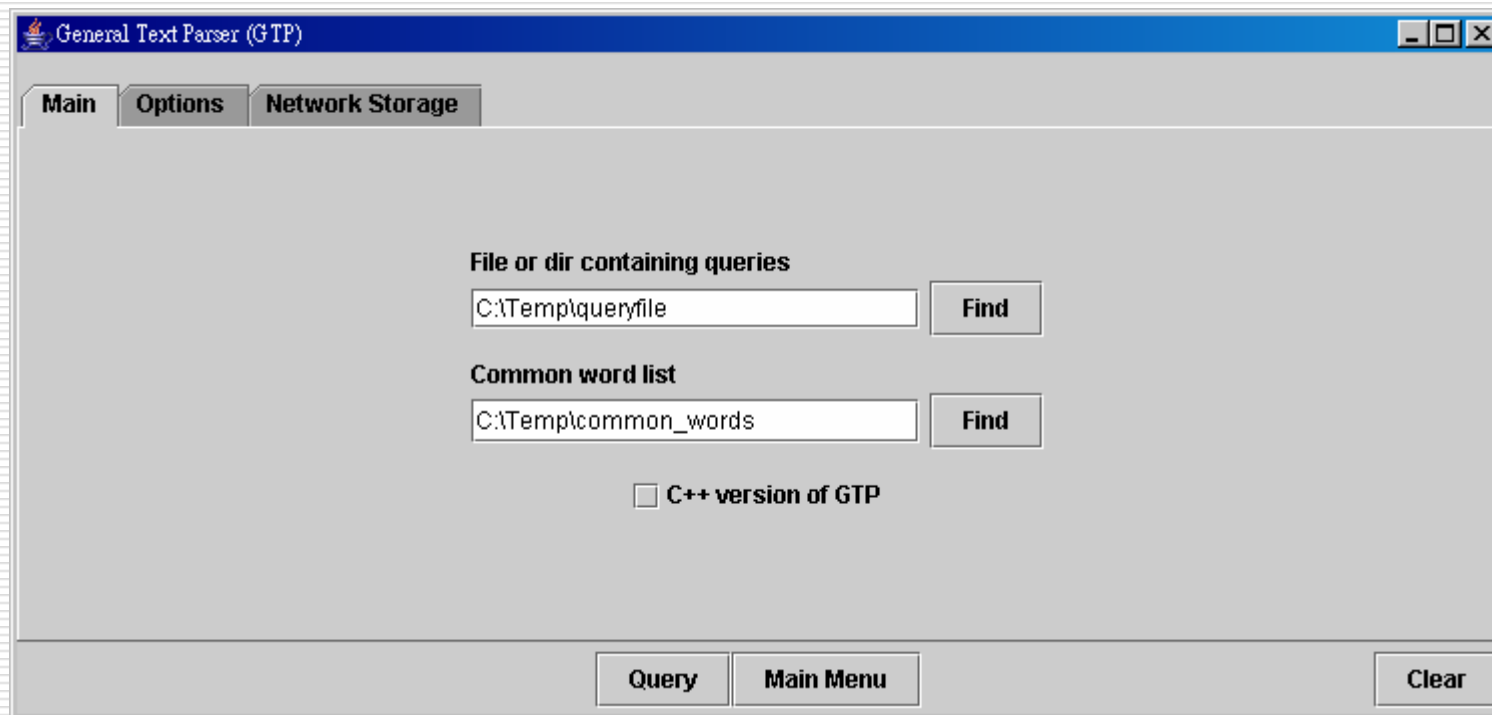
X 訊息

Calculating term weights --- Thu Dec 25 16:53:08 EST 2003
Creating raw matrix --- Thu Dec 25 16:53:08 EST 2003
Creating Harwell-Boeing matrix --- Thu Dec 25 16:53:09 EST 2003
matrix size: 82 x 750 nelem: 1664
Decompose Harwell-Boeing matrix --- Thu Dec 25 16:53:10 EST 2003
Using svd method -- las2 Thu Dec 25 16:53:10 EST 2003
time for gtp is 2700 milliseconds
time for decomposition is 3854 milliseconds
Total time is 6554 milliseconds
Writing summary to RUN_SUMMARY --- Thu Dec 25 16:53:14 EST 2003

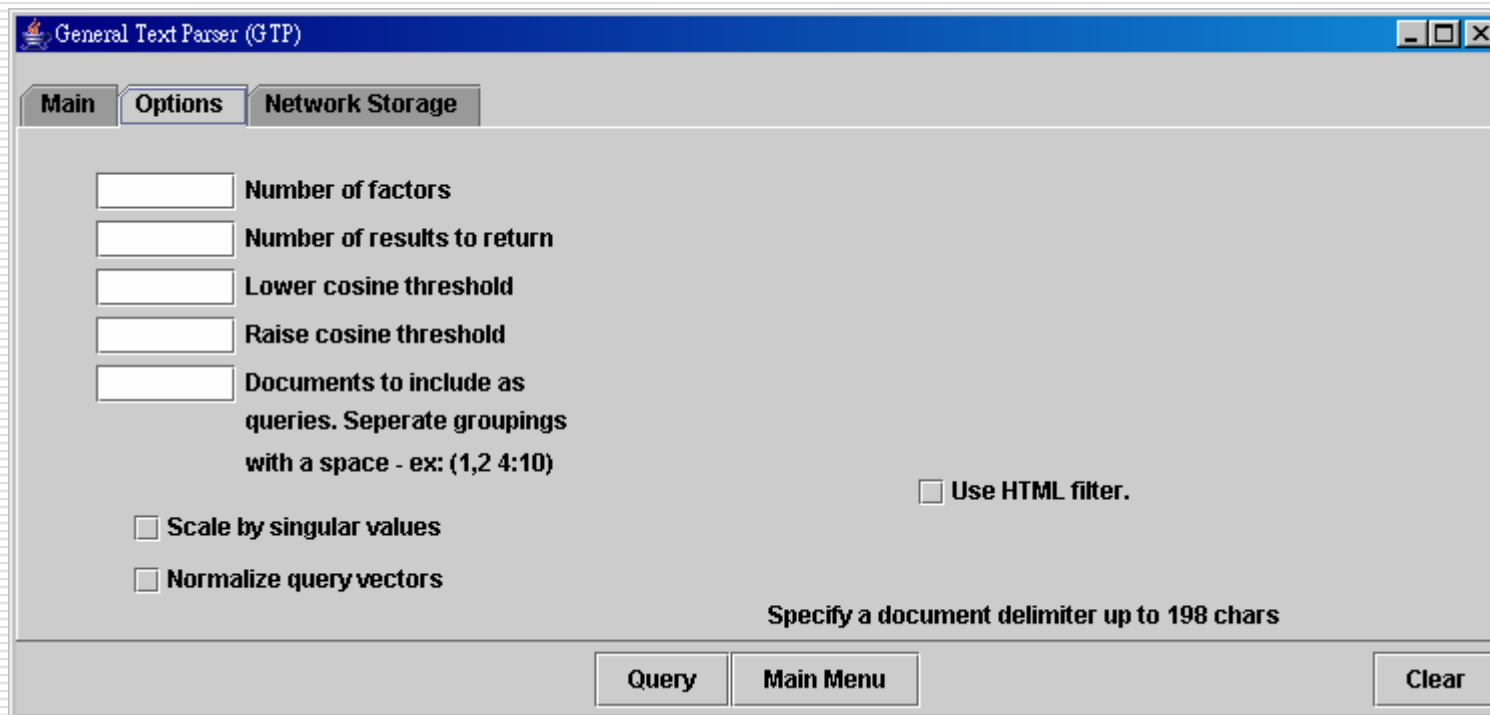
Program exited with value: 0



Query Interface



Query Interface (cont.)



The screenshot shows the 'Options' tab of the 'General Text Parser (GTP)' application. The window title bar includes the application name and standard window controls. The 'Options' tab is selected, and the 'Main' and 'Network Storage' tabs are also visible. The interface contains several input fields and checkboxes for configuring search parameters. At the bottom, there are buttons for 'Query', 'Main Menu', and 'Clear', along with a text prompt for a document delimiter.

General Text Parser (GTP)

Main Options Network Storage

Number of factors

Number of results to return

Lower cosine threshold

Raise cosine threshold

Documents to include as queries. Separate groupings with a space - ex: (1,2 4:10)

Use HTML filter.

Scale by singular values

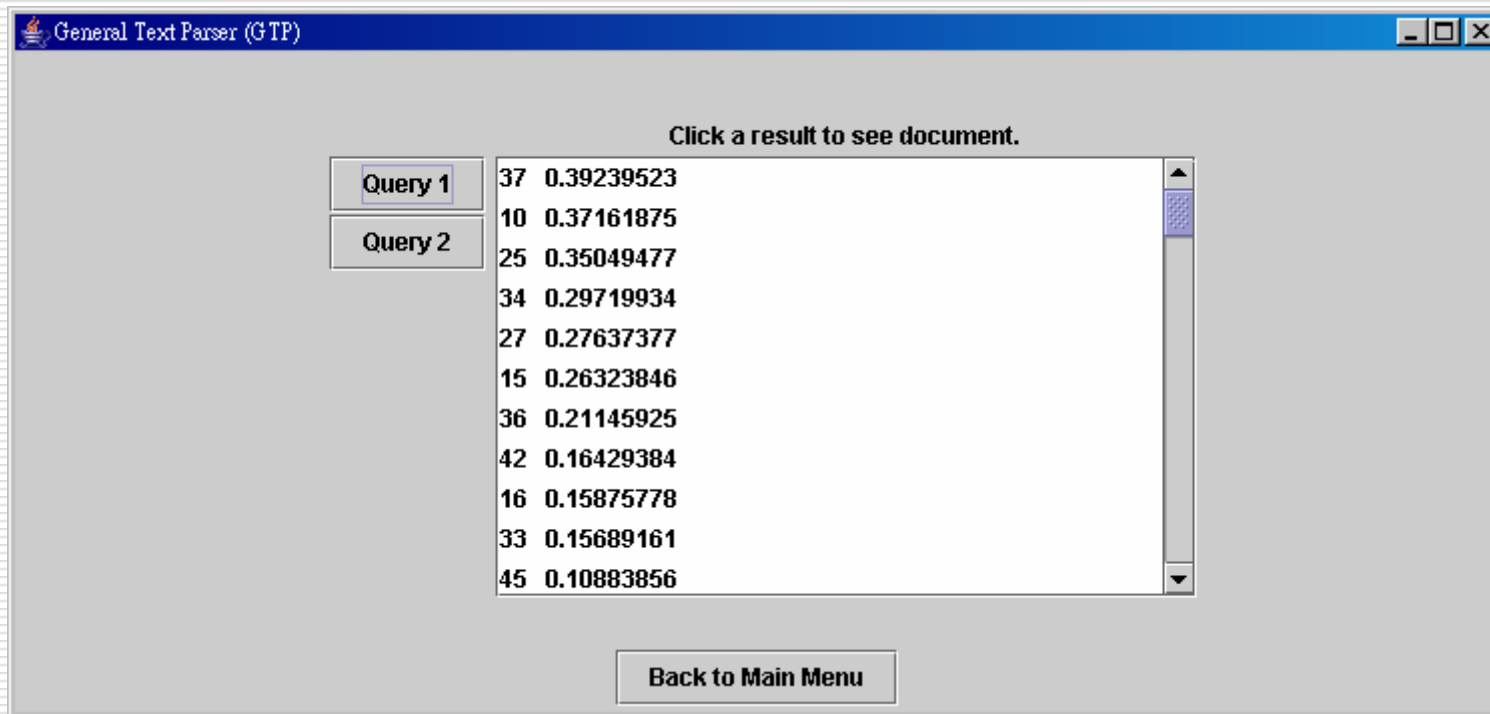
Normalize query vectors

Specify a document delimiter up to 198 chars

Query Main Menu Clear



Show Query Results



The screenshot shows a window titled "General Text Parser (GTP)". Inside the window, there is a list of results for two queries. The results are displayed in a table format with a vertical scrollbar on the right. The text "Click a result to see document." is centered above the table. Below the table is a button labeled "Back to Main Menu".

Query	Result ID	Score
Query 1	37	0.39239523
Query 1	10	0.37161875
Query 2	25	0.35049477
Query 2	34	0.29719934
Query 2	27	0.27637377
Query 2	15	0.26323846
Query 2	36	0.21145925
Query 2	42	0.16429384
Query 2	16	0.15875778
Query 2	33	0.15689161
Query 2	45	0.10883856

Click a result to see document.

Back to Main Menu



Tips

- ❑ Modify compress tool (Cleaner.java)
 - in line 301 & 335: “child =
Runtime.getRuntime().exec(“compress ”)”
 - `gzip -f` (Linux).
 - `C:\Program Files\WinRAR\rar a file1
file2` (Windows).
 - ❑ file1:
 - ❑ file2: rawmatrix, matrix.hb



Tips (cont.)

□ For Windows

- Comment line 230 & 265 in Cleaner.java
- `Runtime.getRuntime().exec("date >> RUN_SUMMARY");`

□ Recomplie class

- `javac Cleaner.java`
- `javac GTP.java`



Tips (cont.)

- ❑ Avoid Java OutOfMemory Exception
 - Larger document collection needs more memory.
 - Set maximum Java heap size:
 - ❑ java **-Xmx1024M** classname

