

CAP6776 – Information Retrieval

**Wumpus Information Retrieval
System**

Written by:

Christopher Foley
Z15092976

Academic Year: 2017-2018

1 Introduction

The Wumpus Search Engine was developed at the University of Waterloo (Canada) to study desktop indexing in an environment where files are changing.¹ This allows us to use the Wumpus search in a wide variety of environments.

2 Overview of Wumpus

The Wumpus search engine was developed by Stefan Buttcher, currently at Google Inc., while at the University of Waterloo. It is intended as a small compact search engine, freely available under GPL, which will permit users to explore searching. The documentation², tutorial³ and related publication⁴ lists are available on the www.wumpus-search.org web site. Experience has indicated that it is best to read and work through the examples in the documentation before the tutorial.

The Wumpus search engine is also incorporated into student exercises in the text “Information Retrieval Implementing and Evaluating Search Engines” by Buttcher, et al.⁵

3 Issues

The source provided at www.wumpus-search.org did initially compile due to 2011 changes to the C++ language specification. Although a web based interface exists, all examples and testing were done through a command line, due to lack of a web server until 4-Dec-2017 (7 PM).

PHP

The web interface was verified on 5-Dec-2017. The web interface requires a login with username/password combination. As of 5-Dec-2017, after successfully logging in, queries via the web interface were not successful. Still under investigation.

4 Installation

Following the directions from the Wumpus—search.org .

- The download was unpacked (tar)
- make was executed

1 Wumpus homepage, www.wumpus-search.org

2<http://www.wumpus-search.org/docs/index.html>

3http://www.wumpus-search.org/docs/wumpus_tutorial.pdf

4<http://www.wumpus-search.org/publications.html>

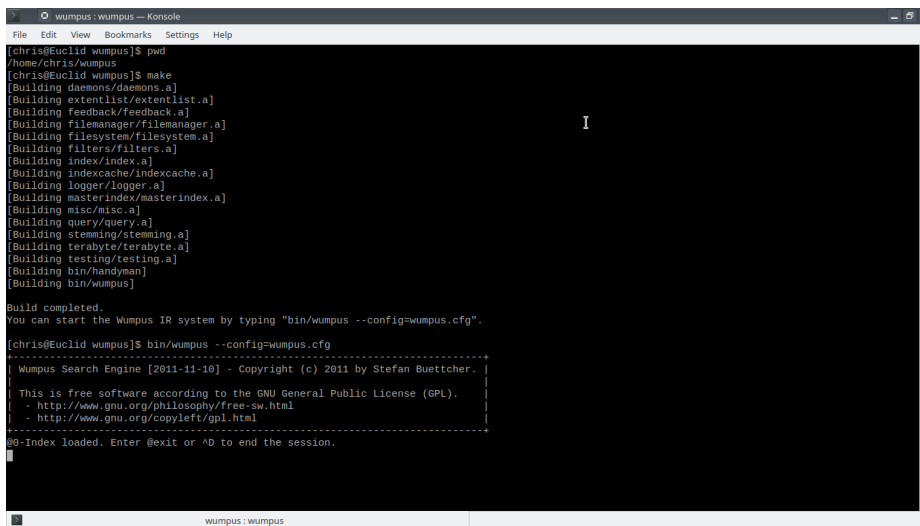
5Stefan Buttcher, Charles L.A. Clarke and Gordon V. Cormack, “Information Retrieval Implementing and Evaluating Search Engines”, MIT Press (2010 – Now Listed as Out of Print).

Information Retrieval

Due to changes in C++ the 2011 source would not compile with the gcc compiler on the target system. The following files were modified to create a build:

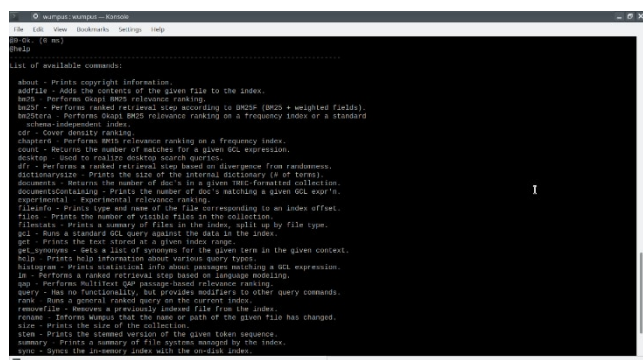
- /query/bm25query.h
- /query/xpath_primitives.cpp
- /query/cdrquery.h
- /query/rankedquery.h
- /query/desktopquery.h
- /query/npquery.h
- /query/language_model_query.h
- /query/qapquery.h
- /daemons/client_connection.cpp
- /misc/macros.h
- /misc/general_avltree.h
- /filemanager/filemanager.h
- /feedback/language_model.h
- /filesystem/bucketfilesystem.h
- /index/my_inplace_index.h
- /index/ondisk_index_manager.cpp
- /index/compressed_lexicon.h
- /index/realloc_lexicon.h
- /index/index_types.h
- /index/compactindex.h

Information Retrieval

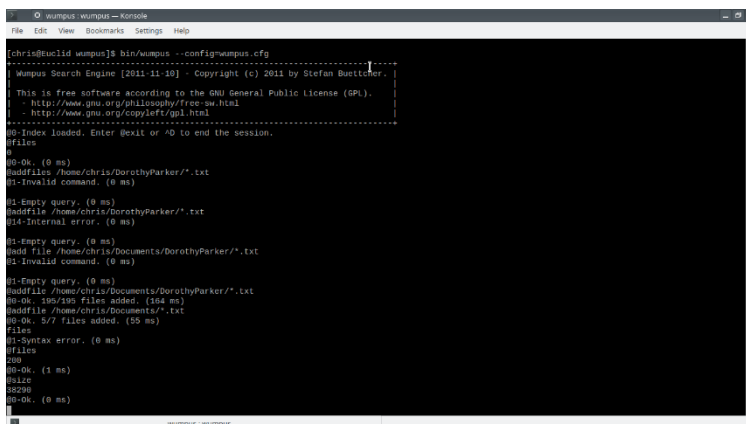


After fixing syntax errors and updating the code a make clean; make was successful.

The Wumpus command line interpreter is one of many ways to access the Wumpus engine. Using the command line @help will give a list of the commands available, and complicated commands are given through use of the GCL (*generalized concordance lists*) query language.⁶



The Wumpus engine is designed for local search and response. Therefore files must be added to its local index. This is accomplished through use of the @addfile command to the command line. The following shows the addition of 195 .txt files using a wildcard designator.



Wumpus is also capable of parsing and interpreting .xml files, therefore a library of Shakespeare plays was also loaded into the index. XML files were loaded to test the Wumpus capability of searching/filtering based on tags only. For example, if your tagged document

6C.L.A. Clarke, G.V. Cormack and F.J. Burkowski. "An Algebra for Structured Text Search and a Framework for its Implementation", *The Computer Journal*, 38(1):43-56, 1995.

Information Retrieval

contains a play it could be delimited with the “<play>” .. “</play>” tags. Wumpus could filter a request returning all references to Caesar in plays in the following manner:

(“<play>”..“</play>”) > “Caesar”

Executing this command on my sample server gave the following output:

```
wumpus: wumpus — Konsole
File Edit View Bookmarks Settings Help

@1-Empty query. (0 ms)
@clear
@1-Invalid command. (0 ms)

@1-Empty query. (0 ms)
@gcl [get] ("<play>","</play>") > "Caesar"
39963 80445 "<PLAY> <TITLE>The Tragedy of Antony and Cleopatra</TITLE> <FM> <P>Text placed in the public domain by Moby Lexical Tools, 1992.</P> <P>SGML markup by Jon Bosak, 1992-1994.</P> <P>XML version by Jon Bosak, 1996-1998.</P> <P>This work may be freely copied a"
80459 115584 "<PLAY> <TITLE>All's Well That Ends Well</TITLE> <FM> <P>Text placed in the public domain by Moby Lexical Tools, 1992.</P> <P>SGML mark
up by Jon Bosak, 1992-1994.</P> <P>XML version by Jon Bosak, 1996-1998.</P> <P>This work may be freely copied and distrib"
115531 147863 "<PLAY> <TITLE>As You Like It</TITLE> <FM> <P>Text placed in the public domain by Moby Lexical Tools, 1992.</P> <P>SGML markup by Jon
Bosak, 1992-1994.</P> <P>XML version by Jon Bosak, 1996-1998.</P> <P>This work may be freely copied and distributed worldw"
213403 254924 "<PLAY> <TITLE>Cymbeline</TITLE> <FM> <P>Text placed in the public domain by Moby Lexical Tools, 1992.</P> <P>SGML markup by Jon Bosak
, 1992-1994.</P> <P>XML version by Jon Bosak, 1996-1998.</P> <P>This work may be freely copied and distributed worldwide.<"
279291 325575 "<PLAY> <TITLE>The Tragedy of Hamlet, Prince of Denmark</TITLE> <FM> <P>Text placed in the public domain by Moby Lexical Tools, 1992.<
/P> <P>SGML markup by Jon Bosak, 1992-1994.</P> <P>XML version by Jon Bosak, 1996-1998.</P> <P>This work may be freely cop"
362011 401099 "<PLAY> <TITLE>The Second Part of Henry the Fourth</TITLE> <FM> <P>Text placed in the public domain by Moby Lexical Tools, 1992.</P> <
P>SGML markup by Jon Bosak, 1992-1994.</P> <P>XML version by Jon Bosak, 1996-1998.</P> <P>This work may be freely copied a"
461115 439280 "<PLAY> <TITLE>The Life of Henry the Fifth</TITLE> <FM> <P>Text placed in the public domain by Moby Lexical Tools, 1992.</P> <P>SGML m
arkup by Jon Bosak, 1992-1994.</P> <P>XML version by Jon Bosak, 1996-1998.</P> <P>This work may be freely copied and distr"
439227 471527 "<PLAY> <TITLE>The First Part of Henry the Sixth</TITLE> <FM> <P>Text placed in the public domain by Moby Lexical Tools, 1992.</P> <P>
SGML markup by Jon Bosak, 1992-1994.</P> <P>XML version by Jon Bosak, 1996-1998.</P> <P>This work may be freely copied and"
471547 509223 "<PLAY> <TITLE>The Second Part of Henry the Sixth</TITLE> <FM> <P>Text placed in the public domain by Moby Lexical Tools, 1992.</P> <P>
SGML markup by Jon Bosak, 1992-1994.</P> <P>XML version by Jon Bosak, 1996-1998.</P> <P>This work may be freely copied an"
509243 545980 "<PLAY> <TITLE>The Third Part of Henry the Sixth</TITLE> <FM> <P>Text placed in the public domain by Moby Lexical Tools, 1992.</P> <P>
SGML markup by Jon Bosak, 1992-1994.</P> <P>XML version by Jon Bosak, 1996-1998.</P> <P>This work may be freely copied and"
582491 612760 "<PLAY> <TITLE>The Tragedy of Julius Caesar</TITLE> <FM> <P>Text placed in the public domain by Moby Lexical Tools, 1992.</P> <P>SGML
markup by Jon Bosak, 1992-1994.</P> <P>XML version by Jon Bosak, 1996-1998.</P> <P>This work may be freely copied and dist"
683467 717117 "<PLAY> <TITLE>Love's Labor's Lost</TITLE> <FM> <P>Text placed in the public domain by Moby Lexical Tools, 1992.</P> <P>SGML markup by
Jon Bosak, 1992-1994.</P> <P>XML version by Jon Bosak, 1996-1998.</P> <P>This work may be freely copied and distributed w"
717131 756505 "<PLAY> <TITLE>Measure for Measure</TITLE> <FM> <P>Text placed in the public domain by Moby Lexical Tools, 1992.</P> <P>SGML markup by
Jon Bosak, 1992-1994.</P> <P>XML version by Jon Bosak, 1996-1998.</P> <P>This work may be freely copied and distributed w"
756587 784762 "<PLAY> <TITLE>The Merry Wives of Windsor</TITLE> <FM> <P>Text placed in the public domain by Moby Lexical Tools, 1992.</P> <P>SGML ma
rkup by Jon Bosak, 1992-1994.</P> <P>XML version by Jon Bosak, 1996-1998.</P> <P>This work may be freely copied and distri"
784779 811559 "<PLAY> <TITLE>The Tragedy of Macbeth</TITLE> <FM> <P>Text placed in the public domain by Moby Lexical Tools, 1992.</P> <P>SGML markup
by Jon Bosak, 1992-1994.</P> <P>XML version by Jon Bosak, 1996-1998.</P> <P>This work may be freely copied and distribute"
```

Help on many commands is given through the @help libraries. In the following, the result of “@help count”:

```
wumpus: wumpus — Konsole
File Edit View Bookmarks Settings Help

prints an XPath expression for each given index position returned; only
works if the ENABLE_XPATH configuration variable was set when building
the index
For further modifiers, see "@help query".
-----
@0-Ok. (0 ms)

@1-Empty query. (0 ms)
@
@1-Invalid command. (0 ms)
@help count
-----
count - Returns the number of matches for a given GCL expression.
[Aliases: estimate]

Examples:

@count (((("mother"^^"father")+"parents").."children")<[10]
30
@0-Ok. (2 ms)
@count[size] (((("mother"^^"father")+"parents").."children")<[10]
156
@0-Ok. (2 ms)
@count[avgsz] (((("mother"^^"father")+"parents").."children")<[10]
5.2
@0-Ok. (2 ms)
@count "this", "and", "that"
10879, 81435, 41362
@0-Ok. (6 ms)

Query modifiers supported:
boolean size (default: false)
if set, the search engine returns the total size of all matches
boolean avgsz (default: false)
if set, the search engine returns the average size of all matches
-----
@0-Ok. (0 ms)
```

Information Retrieval

Using the GCL syntax, we can then query the indices and extract information from our documents. We first begin by obtaining a count showing the number of times the tokens “mother” or “father” are found:

```
@count ("mother"+"father")
```

```
1468
```

```
@0-Ok. (2 ms)
```

This indicates that the tokens occur 1468 times. The token “death” occurs 871 times:

```
@count "death"
```

```
871
```

```
@0-Ok. (1 ms)
```

Multiple tokens can be retrieved on a single command line:

```
@count "death", "die", "sleep"
```

```
871, 470, 269
```

```
@0-Ok. (1 ms)
```

Tokens may be stemmed through use of the “\$” symbol. Stemming is done at search time.

```
@count "$death", "die$", "$sleep"
```

```
1471, 470, 398
```

```
@0-Ok
```

Simply entering the token will give the indices of documents matching the tokens.

```
“mother”
```

```
3846 3846
```

```
5268 5268
```

```
6146 6146
```

```
10640 10640
```

```
10964 10964
```

```
13874 13874
```

Information Retrieval

16649 16649

@0-Ok. (0 ms)

Using the token, text may be retrieved:

@get 3840 3852

machinist at Phoenix Armour. Parker's mother died in 1898. Jacob married in

@0-Ok. (20 ms)

Combining tokens with GCL we can then extract relevant text, with filtering. The command shown below extracts text with 6 tokens before the word "mother", which is then followed by 6 more tokens.

@gcl [get] ([6].."mother"..)[6]

3840 3852 "machinist at Phoenix Armour. Parker's mother died in 1898. Jacob married in"

6140 6152 "all are proud to know As mother, wife, and authoress- Thank God, I"

10634 10646 "light my pipe, an' reach fer Mother's hand, An' I wouldn't"

13868 13880 "have him back!' I hope Her mother washed her mouth with soap.
Dorothy"

16643 16655 "You'd know he was his mother's son. 'It's queer that"

49020 49032 "Thou hast a sister by the mother's side,</LINE> <LINE>Admired Octavia:"

53527 53539 "were at blows,</LINE> <LINE>Your mother came to Sicily and did find"

.

Edited for space

.

85914 85926 "<LINE>I say, I am your mother.</LINE> </SPEECH> <SPEECH>
<SPEAKER>HELENA</SPEAKER>"

85993 86005 "COUNTESS</SPEAKER> <LINE>Nor I your mother?</LINE> </SPEECH>
<SPEECH> <SPEAKER>HELENA</SPEAKER>"

@0-Ok. (8 ms)

Ranking

The Wumpus Search engine is capable of returning queries ranked on relevance to the query. Results are ranked using Okapi BM25. TREC evaluation is also available.

TELNET/TCP

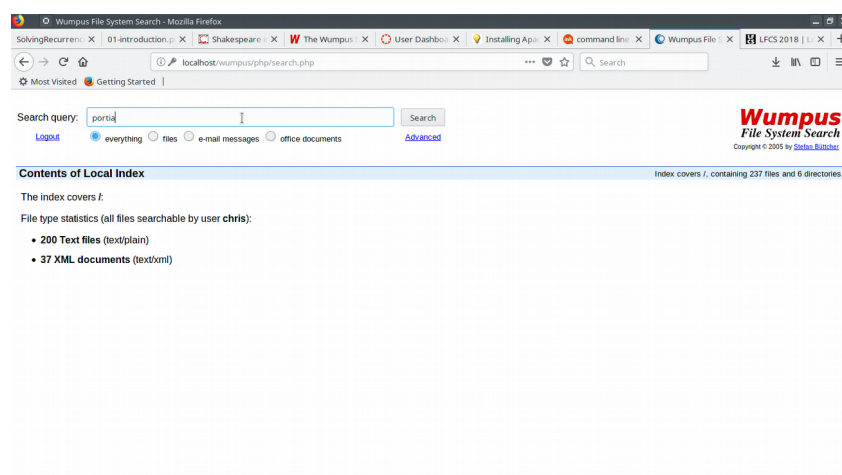
Executing Wumpus with a command line option `TCP_PORT=12345`, or a change to the `Wumpus.cfg` file will direct Wumpus to listen on the specified port, allowing remote connections. A Telnet connection to the port provides full command line access.

WEB

A recent (2011) release of the Wumpus Search engine, contains a web interface written in php which permits queries. The search engine requires the Wumpus password file to be updated with permitted users and their **unencrypted** passwords. The initial login appears as follows:



After a successful login, the initial stats are displayed. As of 5-Dec the query interface is not working for me.



It should be noted that 200 text files are indexed (primarily poems of Dorothy Parker) and 37 XML documents are indexed (Shakespeare).

5 Appendices

GCL Command Language

The GCL command language referred to above is based upon the work of Clarke, Cormack and Burkowski, as noted above. A summary of the command language is built into Wumpus and may be displayed as part of the @help libraries by the command “@help gcl” as shown below:

```
@help gcl
```

```
-----
```

```
gcl - Runs a standard GCL query against the data in the index.
```

```
For a thorough description of the GCL query language, have a look at
```

```
Clarke et al., "An Algebra for Structured Text Search and a Framework for  
its Implementation". The Computer Journal, 38(1):43-56, 1995.
```

```
@gcl is the standard query type. That is, if unspecified, @gcl is assumed.
```

Examples:

```
@gcl[get][count=3] ("because"^"of")<[5]
```

```
1158 1161 "because the window of"
```

```
1569 1573 "of R.H. Macy because"
```

```
1573 1574 "because of"
```

```
@0-0k. (124 ms)
```

```
"later that day"
```

```
2880204 2880206
```

```
3560135 3560137
```

```
3897696 3897698
```

```
@0-0k. (3 ms)
```

Operators supported:

```
"^" (Boolean AND), "+" (Boolean OR), ">" (CONTAINS),
```

```
">" (DOES-NOT-CONTAIN), "<" (CONTAINED-IN), "<" (NOT-CONTAINED-IN),
```

```
".." (FOLLOWED-BY), [N] (window of N char's), N (absolute index address)
```

Information Retrieval

In addition to the canonical GCL operators, Wumpus also understands extended restrictions based on file-related meta-data, for example:

```
{filetype=text/xml} matches all files of type text/xml
{filesize > 100000} matches all files bigger than 100,000 bytes
{filepath=/home/wumpus/*} matches all files below the given directory
"<file!>" returns the start offset of all visible files
"</file!>" returns the end offset of all visible files
```

Query modifiers supported:

```
boolean get (default: false)
    returns the text at each matching index position
boolean filtered (default: false)
    to be used in conjunction with [get]: does not return the original text,
    but the text after being run through Wumpus' input tokenizer
boolean getxpath (default: false)
    prints an XPath expression for each given index position returned; only
    works if the ENABLE_XPATH configuration variable was set when building
    the index
For further modifiers, see "@help query".
```

@@-Ok. (0 ms)

Help Text Output

@help

List of available commands:

```
about - Prints copyright information.
addfile - Adds the contents of the given file to the index.
bm25 - Performs Okapi BM25 relevance ranking.
bm25f - Performs ranked retrieval step according to BM25F (BM25 + weighted fields).
bm25tera - Performs Okapi BM25 relevance ranking on a frequency index or a standard
    schema-independent index.
cdr - Cover density ranking.
chapter6 - Performs BM15 relevance ranking on a frequency index.
```

Information Retrieval

count - Returns the number of matches for a given GCL expression.

desktop - Used to realize desktop search queries.

dfr - Performs a ranked retrieval step based on divergence from randomness.

dictionarysize - Prints the size of the internal dictionary (# of terms).

documents - Returns the number of doc's in a given TREC-formatted collection.

documentsContaining - Prints the number of doc's matching a given GCL expr'n.

experimental - Experimental relevance ranking.

fileinfo - Prints type and name of the file corresponding to an index offset.

files - Prints the number of visible files in the collection.

filestats - Prints a summary of files in the index, split up by file type.

gcl - Runs a standard GCL query against the data in the index.

get - Prints the text stored at a given index range.

get_synonyms - Gets a list of synonyms for the given term in the given context.

help - Prints help information about various query types.

histogram - Prints statistical info about passages matching a GCL expression.

lm - Performs a ranked retrieval step based on language modeling.

qap - Performs MultiText QAP passage-based relevance ranking.

query - Has no functionality, but provides modifiers to other query commands.

rank - Runs a general ranked query on the current index.

removefile - Removes a previously indexed file from the index.

rename - Informs Wumpus that the name or path of the given file has changed.

size - Prints the size of the collection.

stem - Prints the stemmed version of the given token sequence.

summary - Prints a summary of file systems managed by the index.

sync - Syncs the in-memory index with the on-disk index.

system - Executes a given command line via system(3).

updateattr - Makes Wumpus update its internal information about a given file.

vectorspace - Performs ranked retrieval based on the vector space model.

xpath - Executes an XPath query against the index.

For information about a specific command, type "@help command-name".

@@-Ok. (1 ms)

6 Outstanding Issues

- Login queries
- Different types of documents (docx, pdf, odt...)
- Possible simplified text input substituting algebra for text