SensPrecOptimizer

Technical Specification

Title A software tool for combining search queries to design efficient search strategies

Version 2.0.X

Depends None

Author Mohsen Mesgarpour <mohsen.mesgarpour@gmail.com>

Contributors Bita Mesgarpour, Harald Herkner

Date 2013-10-28

Description Development of Highly Sensitive Search Strategies, Sensitivity and precision analysis for "OR" combination of search queries

Supported OSs Windows (6.0 or above) or Mac (10.4 or above) or Linux (Fedora 17 or above)

License Apache License 2.0

Input and Output Encoding UTF-8

3rd Party Software Needs MS Office, Apache OpenOffice

Repository Google Code

Available for download at: https://code.google.com/p/sens-prec-optimizer/

Table of Contents

1. Overview

SensPrecOptimizer is a freeware software to implement profiling of *Highly Sensitive Search Strategies* (HSSS) by combining the search queries. HSSS refers to the two strategies recommended by the Cochrane Handbook for Systematic Reviews of Interventions: a sensitivity-maximizing version and a sensitivity- and precision-maximizing version [1]. This software may also be used for developing search filters or hedges.

1.1. Why it was important to develop this software?

Before answering this question, there are two terms that must be defined: search strategy, and search query. A search strategy is consisted of a number of search queries connected with Boolean operators. A search query is a text string that usually contains the exact sequence of words and/or characters, and may include Boolean or proximity operators.

To develop a HSSS, particularly a sensitivity-maximizing version, a comprehensive set of possible search queries should be constructed. However, obtaining the combination of queries with the best retrieval performance is troublesome in practice, particularly when the number of queries is getting larger (Fig. 1) or when the retrieval records of queries have considerable overlap.

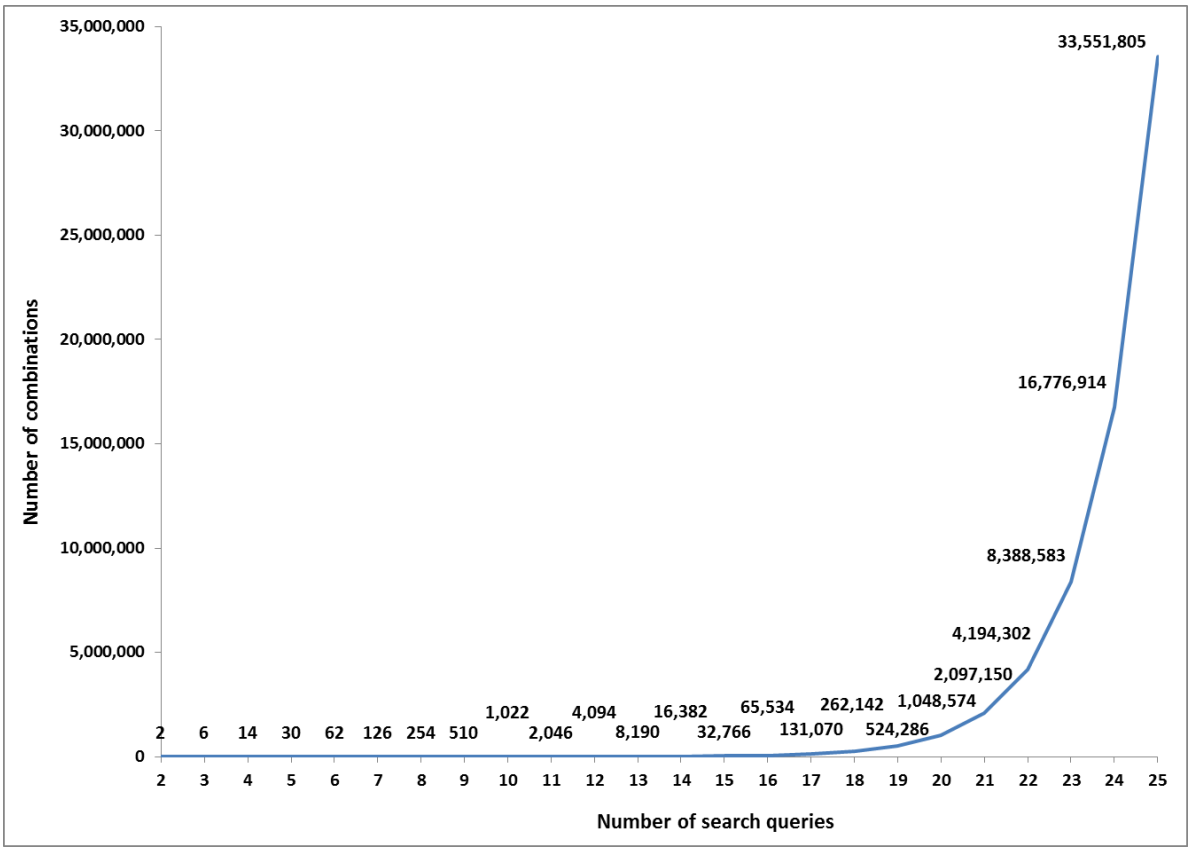

Fig. 1. The exponential growth of the number of combinations as search queries are being added to the search strategies

SensPrecOptimizer analyses the “OR” combination of search queries to construct the best performing search strategies or HSSS.

1.2. History

A need for such a software has been realized during the study on developing search strategies for identifying off-label drug use in MEDLINE and EMBASE [2, 3]. The first version of SensPrecOptimizer presented as a poster in the 19th Cochrane Colloquium in Madrid in 2011 [4].

2. Simple Explanation of Analysis Process

The programme initially read and validates configurations, and then read the input file. Afterward, it calculates, filter and output the statistics for the combinations with different length. There are four main iterations in calculating the statistics, based on the combination length of search queries, and each iteration can consist of one or more steps. The step calculates the retrieval performance for search strategies of one particular length. The derived 'maximum sensitivity' in each step is the main condition that can interrupt the further progress of the iteration. The iterations are presented in below:

- **Iteration 1**: Calculate the performance of the Individual search queries
- **Iteration 2:** Calculate the performance of the combination length that is defined in the configuration as the starting point (*Combinations Iteration - The Starting Point*).
- **Iteration 3:** Calculate the larger combinations from the starting point. The combination length increases stepwise up to the point that the sensitivity does not improve.
- **Iteration 4:** Calculate the smaller combinations from the starting point. The combination length decreases up to the point that the sensitivity becomes less than the maximum sensitivity.
- **Iteration 5:** Calculate additional smaller combinations from the stopping point.

Finally, the programme filters and outputs the statistics.

For example, if the input file consists of 15 search queries and SensPrecOptimizer is set with *The Starting Point* 80% and *The Number of Additional Smaller Neighbours* 3:

- **Iteration 1**: Calculate the performance of each 15 search queries
- **Iteration 2:** Calculate the performance of the combination of search queries of length 12, resulted in calculation of 455 combinations (COMB(12,15))
- **Iteration 3:** Calculate the performance of the combinations of length 13 and higher up to the point that the sensitivity does not improve (assume the highest sensitivity in combinations of length 14 is 90.05% and do not increase comparing to combinations of length 13)
- **Iteration 4:** Calculate the combinations from length 11 down to the point that the sensitivity becomes less than the maximum sensitivity (90.05%).

- **Iteration 5:** Calculate three additional smaller combinations from the stopping point with no other stopping condition[1]. For instance, if the stopping point was length 10, then it is going to calculate down to length 8.

2.1 Internal Flow of the Programme

Below figure represents the internal workflow of SensPrecOptimizer using the sequence diagram UML.

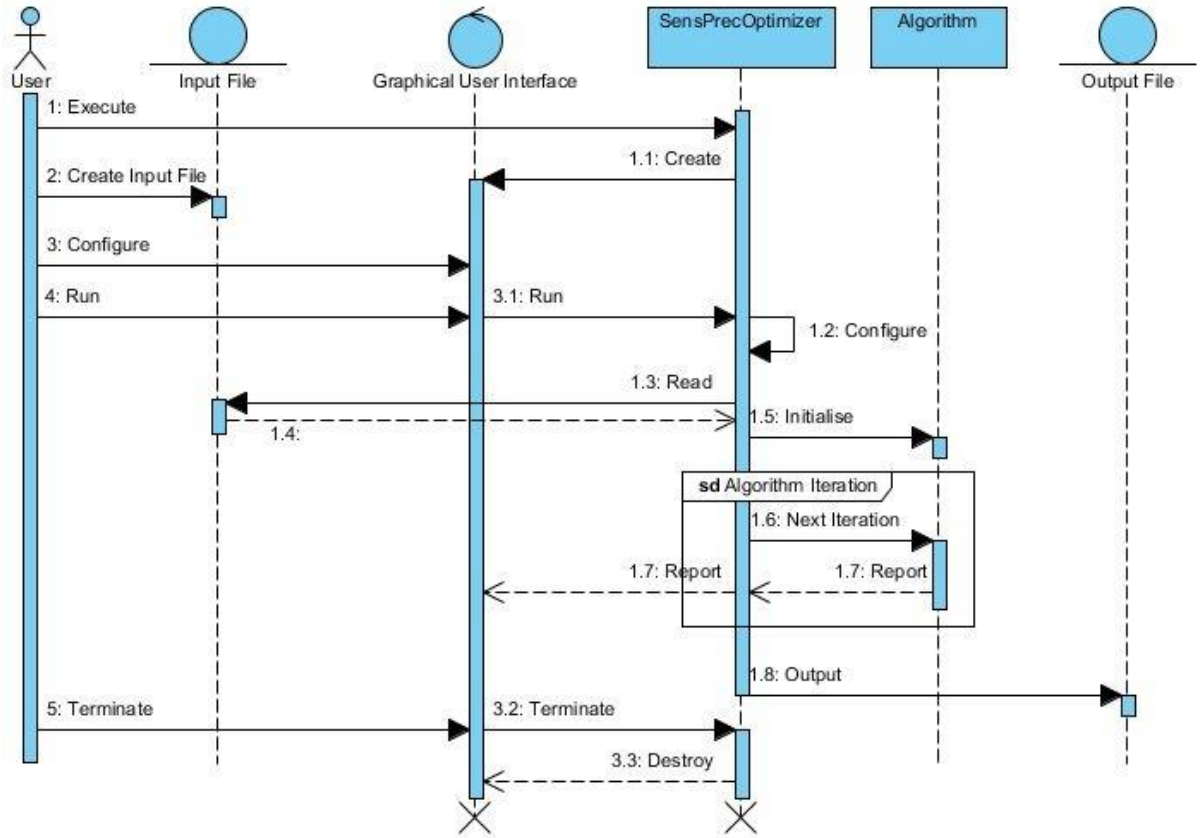

Fig. 2. An abstract sequence-diagram UML of the SensPrecOptimizer

[1] It is obvious that combining all search queries achieve the highest sensitivity, but it is desirable to obtain the highest sensitivity with as less and shortest combination of queries as possible.

3. Development

SensPrecOptimizer is available under a free software license, allowing free access, distribution, modification and distribution of its modified versions, under the terms of the license. It can be installed on the major Operating Systems (see the user manual) and the source code and installation versions (binaries) are available for download at Google Code.

3.1. Setting up a Development Environment

The supported Operating Systems (OS) for developments are:

- MS Windows OS (6.0 and above)
- Apple Mac OS (10.4 and above)
- Linux OS (Fedora 17 and above)

3.1.1. MS Windows OS (6.0 and above)

On a PC with Windows OS, the following packages and libraries must be installed and any other versions or service pack might interfere with the compilation of the programme:

- MS Windows .Net Framework 4.0 (32bit)
- MS Windows SDK (32bit) + SP1
- MS Visual Studio Express, C++ 2010 (32bit) + SP1
- QT 5.x (library and creator), the MSVS 2010 binary (32bit)
- C++ Boost libraries, the MSVS 2010 binary (32bit)

3.1.2. Apple Mac OS (10.4 and above)

On an Apple machine, the following packages and libraries must be installed:

- Install XCode with command line tools from the Mac App Store.
- Install MacPorts, which is a package management system for Mac OS:
    1. Download the source files (not the binary versions).
    2. Compile, install and update using the following commands:

        $ ./configure && make && sudo make install && sudo port -v selfupdate
    3. Add the folders to the path[2]:

[2] To modify the path environment variables globally, you may use the following scripts:

$ touch ~/.bash_profile #create the bash_profile if does not exist
$ open ~/.bash_profile #open it and edit
$ source ~/.bash_profile #reload it
$ echo $PATH #print the path env-variable

$ *export PATH=/opt/local/bin:/opt/local/sbin:$PATH*

4. Update its database:

$ *sudo port -v selfupdate*

- Install boost and boost-build using the macports.

  $ sudo port install boost boost-build

- QT 5.x (library and creator)

3.1.3. Linux OS (Fedora 17 and above)

In a Fedora Linux the packages and libraries that must be installed are as follows:

- C++ Boost and C++ Boost-dev libraries using 'yum' tool
- QT 5.x (library and creator)

3.2. Documentation

The documentation of the code available as part of the source code package. It includes documentation of the code and the UML diagrams. The code documentation is generated from the comments using the DoxyGen document generator, and the UMLs are designed using the community version of the Visual Paradigm for UML.

3.3. Configuration of the Makefile

The makefile has been used for the setting compile time parameters and settings. The below parameters has to be configured for each development environment:

- CONFIG
- INCLUDEPATH
- LIBS

In order to include the external libraries using the QT, please refer to the official documentation of the QT-Creator (Mac Version, Windows version, Linux Version).

References

[1] Lefebvre C, Manheimer E, Glanville J. Chapter 6: Searching for studies. In: Higgins JPT, Green S (editors). Cochrane Handbook for Systematic Reviews of Interventions Version 5.1.0 (updated March 2011). The Cochrane Collaboration, 2011. Available from www.cochrane-handbook.org.

[2] Mesgarpour B, Müller M, Herkner H. Search strategies-identified reports on ‘‘off-label’’ drug use in MEDLINE. J Clin Epidemiol 2012; 65: 827-834.

[3] Mesgarpour B, Müller M, Herkner H. Search strategies to identify reports on “off-label” drug use in EMBASE. BMC Med Res Methodol 2012; 12:190.

[4] Mesgarpour B, Mesgarpour M, Müller M, Herkner H. Developing software for combining search queries to design efficient search strategies. The Cochrane Library: Abstracts of the 19th Cochrane Colloquium 2011(supplement): 58.