

Requirements Document

Mission Statement

To provide software to allow users to perform semantic searching of documents in an easy to use and simple manner.

- Easily see semantic relations of search results.
- Provide meaningful feedback to the search.
- Store and reuse past queries.
- Provide friendly and intuitive interface.
- Allow the user to quickly target search results.

Introduction

This project is called **Waikato Semantic Search 2008** (WSS08). We are building upon the Java 1.6 platform and are incorporating the Java software library ART within our software to provide the semantic structure and searching within. Our software searches Wikipedia and uses Wordnet to find relationships between words.

We aim for our software to be able to run on a typical home desktop computer running Java 1.6, our software does not require an internet connection to be able to perform correctly as it contains a subset of Wikipedia and Wordnet locally.

Hardware Requirements

We aim to target the software to be able to perform on the following hardware:

- **Memory:** At least 512MB
- **Hard-drive:** ≥ 100 MB
- **CPU:** ≥ 1.5 Ghz
- **OS:** Windows XP, Mac or Fedora Linux (with Java 1.6 installed).

Expected Performance

At this stage we expect the program to run in a timely manner, with results being returned as quickly as possible. It is expected that ART will run in the background so that more results show up as time goes on.

Nodes

Nodes are a representation of documents, words, or queries which are visualized as coloured circles.

- **Document Nodes:** Correspond to a physical document on disk.
- **Word Nodes:** Correspond to a word within Wordnet.
- **Query Nodes:** Correspond to previous queries that have been performed.

Queries

A query contains two sets of nodes; one set represents a positive weighting (the 'good' set), the other a negative weighting (the 'bad' set). The weighting is a feedback mechanism performed by the user to guide the search; The positively weighted set corresponds to relevant nodes, while the negative one correspond to irrelevant nodes. When a node is put into the positive set the program will use this information to increase the ranking of nodes related to this one, which will result in more relevant nodes being displayed to the user. When a node is put into the negative set the program will use this information to decrease the ranking of nodes related to this one, which will result in more irrelevant nodes not being displayed to the user.

1. Building a query

The system allows the users to create queries from nodes.

- 1.1. The user can create the initial nodes (by entering key words, a Wikipedia page title or a previous query)
- 1.2. The user adds the initial *seeding* nodes to the query. These nodes can be any of the one mentioned in the "Node" category.
- 1.3. The user classifies the *seeding* nodes into either the 'good' set or the 'bad' set.
- 1.4. With these two sets the user will submit this to the program as one query.

2. Browsing the results

There will be multiple ART searches performed by WSS08; The results of a query will be displayed as a structured graph of clustered nodes, the explicit initial query centred in the middle, and related queries branching off the initial query. Each query forms the nucleus of a cluster with relevant results attached.

- 2.1. The user is able to quickly view descriptions of each result.
- 2.2. The user can view the entirety of a node.
 - 2.2.1. **Document node:** Causes the document to be displayed.
 - 2.2.2. **Word node:** Causes the definition of the word to be displayed.
 - 2.2.3. **Query node:** Performs that query in a new window.
- 2.3. The user can remove a node from the display, which will be replaced by the next most relevant node.

3. Giving feedback

The user can give feedback to the semantic search to narrow down the results and increase the precision. The result display is automatically updated with each feedback placed.

- 3.1. The user can put a node into either a 'good' or 'bad' set.

3.1.1. Please see above, in “Queries”, for information about how this affects the program.

3.2. A user can select an option so that a node is simply not displayed, and this will not affect the search.

4. Filtering nodes

The user can eliminate different types of nodes (Document, Word, or Query) from results.

4.1. The user has an option to select types of nodes.

4.2. The selected node types are displayed within the results.

4.3. The unselected node types are eliminated from the results.

5. Saving and abandoning searches

The user will have the ability to decide on when and whether the search is saved.

5.1. The user will have the ability to manually save the state of a search at any time.

5.2. The program will automatically save the state of a search when a search is complete.

5.2.1. A search is complete when the user starts a new search.

5.2.2. A search is complete when the user terminates the program.

5.3. The user will have the ability to abandon a search which will cause the program to not save any results.

6. Help

The user can get help on how to use this program.

6.1. The user will be able to choose an option to display help on how to use the program.