

Semantic Search on Help Files

Naushad UzZaman
naushad@cs.rochester.edu

Introduction

Every time we start using new software we spend lot of time to figure out how to use that software. The most successful applications are those, which are easily usable without taking help from other people or even seeing the manual or searching the help files, e.g. google. Windows operating system can be another example, but an operating system and many other software are so huge and have so many operations that it is not always possible make it understandable through the user interface. A user certainly needs some assistance to figure out how they can use the new software. All software currently comes with "good" (not always actually!) help documentation. So a possible way to access the help information is to browse the help documents or search with some queries.

It is ironic that information retrieval has emerged so rapidly in recent years, that we can retrieve information (matching the keywords mostly!) over the Internet so easily, but for some reason, searching on help file has always been neglected in a software. Actually most people don't even use the help file because of poor results when you search for something. It mostly tries to do basic keyword matching, which sometimes might show what you are looking for, but not always.

You will see in the next section how badly help-file searching works in popular software, used by millions of users!

Problem

We will explain the problem with one of the most used software in this Internet age, a browser, be it Firefox or Internet Explorer! What could be a basic question in a help search engine from a new user? May be, "open a web page", to learn how to open a web page. We tried this question in both of the browsers; you will be very surprised seeing the results!

Internet Explorer returned a page that is on "To disable a browser add-on". In description it has, "Open an Internet Explorer". In notes it has, "Some Web pages, or Internet Explorer...". So you got your all three keywords "open" "a" and "web pages" without getting the answer of the most basic question. Firefox actually doesn't even give any result for this query! You can say its better not to get anything than getting the wrong result!

But it is evident that even though the search technology has improved so much in the recent years but searching

on help file has been neglected in almost all the software. We cannot even find the basic answer from a help search engine! We will show few more examples to understand how intelligent a searching on help file need to be, to serve the users!

First think of who uses the help files of software? Most of the times a person who is new to that software. So the searching on help file is expected to get naïve queries, e.g. in a banking software, for the query of opening a bank account, the sample queries could be:

- i. opening a bank account
 - ii. open a bank account
 - iii. create a bank account
- another example:
- iv. modify loan information

If we implement a Help file Search Engine based on the simple keyword search then from these example we can understand that it is very much possible to get other results or no results; lets consider what mistake a normal keyword search engine can make when we are searching these normal queries. Assume that there are help documents with title, "*Opening a bank account*" and "*Edit loan information*".

- i. opening a bank account

This one should at least match with the instruction on opening a bank account. However, we have even seen that in case of browsers, we were even unable to get results for simple queries like this. The reason could be there weren't any document with the same title. But in this case, we are assuming that there is a document with this title. So a basic keyword matching search engine should be able to retrieve this page.

- ii. open a bank account

If your search engine for help file is not sophisticated enough to handle different morphological form of the same verb then this page might not show up, or may be similar pages like, "*Canceling a bank account*" will show up! So just keyword matching won't work. You need to include some morphology or stemming to get results for these types of queries.

- iii. create a bank account

It is easy for human being to see that "*create a bank account*" and "*opening a bank account*" are same and by searching "*create a bank account*", the title "*opening a bank account*" should come first! But for keyword matching won't even understand *create* and *open* means the same. It won't be any surprise, if the search engine returns "*delete a bank account*" for this query!

- iv. modify loan information

We had a title "*Edit loan information*". So, if we search with "modify loan information", then the user expects the "*edit loan information*" to show up! But in keyword matching search engine this is unlikely to happen.

Semantic Search engine as solution!

From the examples above, it is evident that a keyword based generic search engine is not enough for a help file search engine, because the user can give naïve search query and it is very obvious to assume that the user will write synonyms of some words, rather than exact word for query. We need to

understand the meaning of the query to find out what documents exist with the same meaning, so that we can give better results. Bottom line is we need to understand the meaning of the query, so we need to make it a semantic search engine. In case of web searching it might not be feasible because of extra overhead of representing the documents in semantic form. But in case of this type of small applications, compare to the web, the domain is so small that we can easily implement a semantic search engine for help files of software.

Semantic Search Engine

We will explain a basic concept for semantic search, the Logical Form, before describing how a basic Semantic Search engine works.

Logical Form (LF): is the representation of the context-independent meaning of a sentence.

The logical form consists of a set of terms describing objects and relationships evoked by the utterance. One key term is speech act that was performed. For example, the logical form of

But the man wants to eat it

is as follows:

```
((LF::SPEECHACT V107605
W::SA_TELL :CONTENT V107499
:MODS (V107426))
This is a TELL speech act with content
V107499 and (discourse) modifier
V107426
```

```
(LF::F V107426 (:* LF::CONJUNCT
W::BUT) :OF V107605)
V107605 is related by a "but"
relationship to previous context.
```

```
(LF::F V107499 (:* LF::WANT
W::WANT) :ACTION V107537
:EXPERIENCER V107492 :TMA
((W::TENSE W::PRES)))
V107499 is a wanting relation between
V107492 and V107537, that holds at a
time indicated by the present tense.
```

```
(LF::THE V107492 (:* LF::MALE
W::MAN))
V107492 is some man identifiable in
context.
```

```
(LF::F V107537 (:* LF::CONSUME
W::EAT) :THEME V107545 :AGENT
V107492)
V107537 is an eating relation between
V107545 and V107492.
```

```
(LF::PRO V107545 (:*
LF::REFERENTIAL-SEM W::IT)
:CONTEXT-REL W::IT))
V107545 is some object identifiable in
context by pronoun "it".
```

This is actually how Logical Form (LF) look like and it is the description of the semantic content of a sentence.

To explain how a semantic search engine will work, let's see how a basic search on small domain will work from abstract

level.

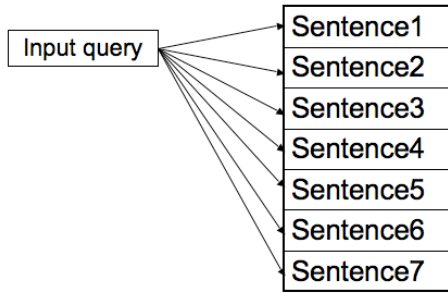


Figure 1: Basic Search

Think that we have few sentences and we give a query to match if we have the given query in the existing system or not. A Basic Search engine will compare with all the sentences, generate some scores (based on some techniques like [Edit-distance] or [LCS]) that represents the difference between the query and all the sentences. So depending on the difference scores, we can sort the sentences and return to the user.

The idea of using semantic knowledge in search is similar. The difference is we compare between the Logical Forms (LF), instead of only the queries. Figure 2 will clarify the concept more.

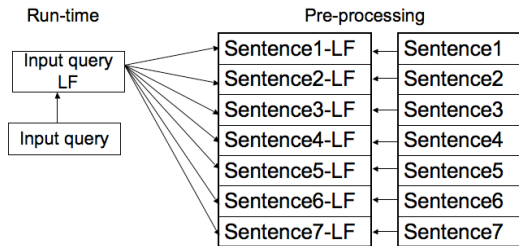


Figure 2: Basic Semantic Search

Here we change all the sentences to its Logical Forms in the preprocessing and in runtime, when user gives a query, then it generates the LF of query and then compare query-LF with all the pre-

generated LFs. It calculates some distance score that represent the semantic difference between two LFs and based on that score it will return the results.

Prototype of our Semantic Search Engine for Mac software Quicken

We implemented a Semantic Search Engine on help-files of Quicken, banking software on Mac operating system [Quicken]. We took the help file folder, made a program to browse the directory and for each files, the program extracts the title of the file and store the title and corresponding absolute path of the file in a new file. We use this file to get the titles for generating LFs of titles and use the path to return the result to the user.

We used the TRIPS Parser [TRIPS] to get the Logical Form of a sentence and also to calculate distance between two LFs. The following figure will explain how it works from abstract level.

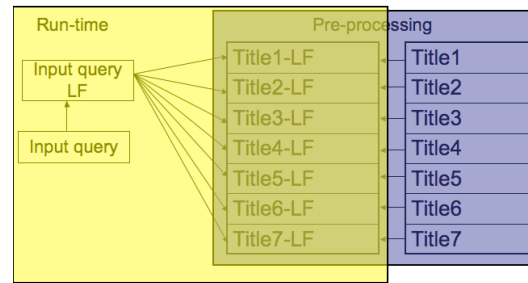


Figure 3: Semantic Search on Help file

From the pre-generated title file, we get all the titles and in preprocessing we generate all the Logical Forms for all these titles. In runtime, when we get the query, we generate the LF for the query

and compare the query-LF with all pre-generated LFs and generate semantic distance scores. We sort the titles based on this scores and return the user top few results as suggestions.

Performance and Evaluation

We experimented our implemented system on 20 queries, added in the APPENDIX A. These queries represents the type of queries the Quicken Help Search engine can get. We tried to make different types of queries rather than similar queries.

We experimented these queries on three different systems.

- i. Our first implemented system
- ii. Our system removing the articles (explained in detail in the next section)
- iii. Quicken 2007 help-file search

The following table explains the performance of three systems.

System	First result (out of 20)	Top 3 (out of 20)
Our first implemented system	8	14
Our system removing the articles	11	13
Quicken 2007 help-file search	7	9

Table 1: Evaluation of different semantic search systems on 20 Quicken help queries

The following figure shows the result in graph.

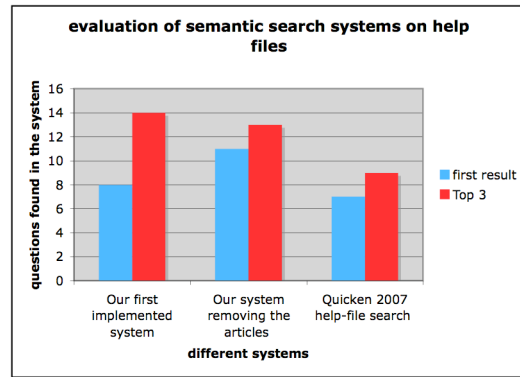


Figure 4: Evaluation of different semantic search systems on 20 Quicken help queries

The following table shows the result in percentage.

System	First result	Top 3
Our first implemented system	40%	70%
Our system removing the articles	55%	65%
Quicken 2007 help-file search	35%	45%

Table 2: Evaluation of Semantic Search systems on 20 Quicken help queries (performance in percentage)

The following figure shows the performance of three systems in percentage.

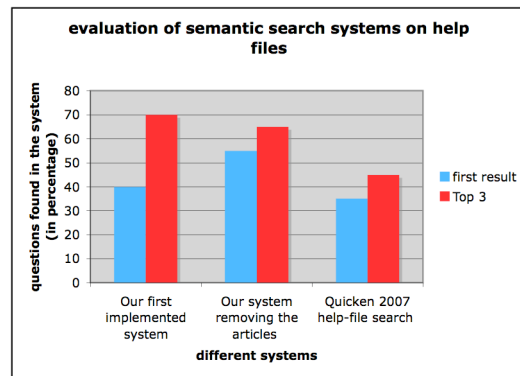


Figure 5: Evaluation of Semantic Search systems on 20 Quicken help queries (performance in percentage)

The figures show that our systems are already performing better than the original Quicken help-file search engine. We can perform very well if it queried in different morphological forms (open instead of opening) or different words in same meaning (*modify* instead of *edit*, *create* instead of *open*, etc). But during evaluations, we found that we have lots of room for improvement. In this section, we will describe in which cases our system is failing, which will be useful to understand where to improve.

Reasoning behind second system without article: Article (a, an, the) was a problem for us. For searching in help file, people usually don't use these articles, but the article cost as another extra word and made the semantic difference value to decrease (more difference), which make it hard to get the expected result in the top (Check results of query 2. *Set up banking account*; 4. *Open account*, of our first system in the APPENDIX A). Hence we implemented our second system, which will remove the articles before generating the LF of titles and queries, which means we are ignoring the articles. Our result showed that the system without article performs better than our first system and can put the expected results in top. But in one case (out of 20), where the first system (with article) was getting the result in top 3, our new system (without article) cannot get that expected result in top 3.

The example query is "*loan payment*". Our expected title is "*Making a loan payment*".

In our first system, it returns following titles with corresponding scores:

0.85714287 "Refinancing a loan"
0.7 "Making a loan payment"

In our next system, without article, it has the same semantic difference 0.7 with the expected result, "Making a loan payment", but we get few other titles with higher scores. Here is what our second system, without article, returns:
0.85714287 "Refinancing a loan"
0.8333333 "Printing a budget"
0.8333333 "Entering a paycheck"

The reason we are getting these results are, we put the words loan, budget and paycheck in the same class in our ontology. So removing the article makes it semantically closer to results we got in top 3. To solve this problem, we should put these words in different hierarchies in our ontology.

We tried to solve the problem of article and showed performance of both systems. But there are other problems, solving those will improve the performance of our system lot better. We are describing these below.

Cannot handle substring: If you search for something that is substring of existing title, our system cannot handle that very properly. For example, we searched for "*edit category*", expecting "*Editing a category or subcategory*". It will have higher similarity scores with titles with smaller length (e.g. *Using Insights, Contents, Trademarks, Copyright*) than title that contains the query (i.e. "*Editing a category or subcategory*").

Different ordering: If the query is in different order than the title then we cannot give good results. For example,

we searched for “*set up web connect account*”, expecting “*Setting up an account for Web Connect*”. Our system results very low semantic difference, like 0.33 for the expected query! But in this case, we had all the keywords in the query.

If the keyword doesn’t match: If the keywords in the query doesn’t match then our system returns titles with smaller length (e.g. *Using Insights, Contents, Trademarks, Copyright*).

These are the problems we found in our current system, and solving these will enable our system to perform better than our existing systems. In the next section, we will try to give directions to solve these problems.

Future Work

Our system is already performing better than the existing Quicken Help File search engine, but to make the system even better we will like to solve the problems found in the current system that has been explained in the previous section.

Problems of different ordering, substring match can be solved very easily in keyword matching search techniques. Our proposed future solution for this search engine will be a hybrid solution, which will take both semantic search and keyword matching into account and will merge two scores to rank the suggestions. In this way, we can very easily solve the problems that we are currently having.

Conclusion

We worked on an easy, but neglected, area of software, search engine for help file! We tried to see the feasibility of semantic search engine as search engine for help files. We implemented a prototype of semantic search engine for help file of Quicken, banking software.

We experimented on 20 different types of queries that represent most of the titles in the help file. We found that 55% times our better system (out of two systems) gives the expected result at top, whereas the Quicken search engine can result the expected result 35% times. 65% times, we get the expected result in Top 3 results, whereas the Quicken get only 45% times.

We also found that there are lot more opportunity to improve our system, we pointed that our system cannot give good suggestions for substrings, query in different ordering. Finally, we proposed solutions to the problems, which will further improve the performance of our system.

Reference

1. [TRIPS] TRIPS: The Rochester Interactive Planning System <<http://www.cs.rochester.edu/research/trips/>>
2. [Quicken] Quicken Software <www.quicken.com>
3. [Edit-distance] Edit distance <http://en.wikipedia.org/wiki/Levenshtein_distance>
4. [LCS] Longest Common Subsequence <http://en.wikipedia.org/wiki/Longest_common_subsequence_problem>

APPENDIX A

1. Query: creating a bank account

Our first implemented system

Returns:

0.9285714 Setting up a bank account"

Expected: same as returned result

Our system removing the articles

Returns:

0.9285714 Setting up a bank account"

Expected: same as returned result

Quicken 2007 help file search results

Tips from Quicken.com

Tracking cash and ATM expenditures

Tracking credit card and ATM

expenditures

2. Query: set up banking account

Our first implemented system

Returns:

0.65 "Printing the register"

0.64285713 "Setting up a bank account"

0.64285713 "Setting up a cash account"

Expected:

0.64285713 "Setting up a bank account"

Our system removing the articles

Returns:

0.71428573 "Setting up a bank account"

0.71428573 "Setting up a cash account"

Expected:

0.71428573 "Setting up a bank account"

Quicken 2007 help file search results

Setting up an account for online banking

Getting started with online banking

Using the PIN Vault

3. Query: edit information

Our first implemented system

Returns:

0.65 "Printing the register"

0.625 "Setting register preferences"

0.5714286 "Editing account information"

Expected:

0.5714286 "Editing account information"

Our system removing the articles

Returns:

0.625 "Setting register preferences"

0.5714286 "Editing account information"

0.5416667 "Printing the register"

Quicken 2007 help file search results

Tips for saving time

Editing a transaction

Editing a class or subclass

4. Query: open account

Our first implemented system

Returns:

0.65 "Printing the register"

0.625 "Setting register preferences"

0.6 "Opening an account"

Expected:

0.6 "Opening an account"

Our system removing the articles

Returns:

0.8 "Opening an account"

0.7 "Deleting an account"

0.7 "Hiding an account"

Expected:

0.8 "Opening an account"

Quicken 2007 help file search results

Starting a new file for a new year

Opening an account

Tips for saving time

5. Query: set up web connect account

Our first implemented system

Returns:

0.8 "Printing the register"

0.75 "Setting register preferences"

0.6 "Setting up a password"

Expected:
0.333 "Setting up an account for Web Connect"

Our system removing the articles
0.75 "Setting register preferences"
0.6818182 "Entering a buy or sell transaction"
0.6 "Setting up a password"

Expected:
0.42857143 "Setting up an account for Web Connect"

Quicken 2007 help file search results
Setting up an account for Web connect
Downloading your account statement
What's new in Quicken 2007

6. Query: cancel account

Our first implemented system

Returns:

0.6666667 "Using Insights"
0.6666667 "Contents"
0.6666667 "Trademarks"
0.6666667 "Copyright"

Expected:

0.4 Deleting an account

Our system removing the articles

Returns:

0.6666667 "Using Insights"
0.6666667 "Contents"
0.6666667 "Trademarks"
0.6666667 "Copyright"

Expected:

0.4 Deleting an account

Quicken 2007 help file search results

Canceling an online service
Getting started with online banking
Performing data file maintenance

7. Query: asset liability account

Our first implemented system

Returns:

0.6666667 "Using Insights"

0.6666667 "Contents"
0.6666667 "Trademarks"
0.6666667 "Copyright"

Expected:

0.42857143 "Setting up an asset or liability account"

Our system removing the articles

0.9 "Editing a transaction"
0.71428573 "Reconciling a cash account"

0.6666667 "Trademarks"

Expected:

0.3 "Setting up an asset or liability account"

Quicken 2007 help file search results

Closing an asset account
Reconciling an asset or liability account
Setting up and asset or liability account

8. Query: close asset account

Our first implemented system

Returns:

0.78571427 "Closing an asset account"
Expected: same as returned result

Our system removing the articles

Returns:

0.85714287 "Closing an asset account"
0.7 "Deleting an account"
Expected: same as returned result

Quicken 2007 help file search results

Closing an asset account

9. Query: loan payment

Our first implemented system

Returns:

0.85714287 "Refinancing a loan"
0.7 "Making a loan payment"
Expected:
0.7 "Making a loan payment"

Our system removing the articles

0.85714287 "Refinancing a loan"

0.8333333 "Printing a budget"
0.8333333 "Entering a paycheck"
Expected:
0.7 "Making a loan payment"

Quicken 2007 help file search results
Viewing a loan's payment schedule
Making a loan payment
Setting up a loan

10. Query: modify loan information

Our first implemented system
Returns:
0.8636364 "Editing loan information"
Expected: same as returned result

Our system removing the articles
Returns:
0.8636364 "Editing loan information"
Expected: same as returned result

Quicken 2007 help file search results
Viewing a loan's payment schedule
Making a loan payment
Setting up a loan

11. Query: append images in checks

Our first implemented system
Returns:
0.65 "Adding images to checks"
Expected: same as returned result

Our system removing the articles
Returns:
0.65 "Adding images to checks"
Expected: same as returned result

Quicken 2007 help file search results
No pages with your search words were found

12. Query: change NSF check

Our first implemented system
Returns:
0.77272725 "Editing loan information"

0.75 "Investment tips"
0.75 "Tax tips"
0.75 "Changing graph preferences"
Expected:
0.5555556 Replacing an NSF check

Our system removing the articles
0.77272725 "Creating a budget report"
0.77272725 "Requesting a payment investigation"
0.77272725 "Editing loan information"
Expected:
0.667 Replacing an NSF check

Quicken 2007 help file search results
No pages with your search words were found

13. Query: print budget

Our first implemented system
Returns:
0.85714287 "Refinancing a loan"
0.72727275 "Printing a budget"
0.6666667 "Using Insights"
Expected:
0.72727275 "Printing a budget"

Our system removing the articles
0.8888889 "Printing a budget"
0.85714287 "Refinancing a loan"
0.8333333 "Entering a paycheck"
Expected:
0.8888889 "Printing a budget"

Quicken 2007 help file search results
Printing a budget
Setting register preferences
Planning tips

14. Query: print register

Our first implemented system
Returns:
0.85 "Printing the register"
Expected: same as returned result

Our system removing the articles

Returns:
0.6666667 "Printing the register"
0.6666667 "Setting register preferences"
0.59090906 "Entering a buy or sell transaction"

Quicken 2007 help file search results
Printing a register
Reviewing check's you've written
Adding a note to a check

15. Query: edit category

Our first implemented system

Returns:
0.7777778 "Assigning a category"
0.7222222 "Customizing a graph"
0.7222222 "Deleting a budget"
Expected:
0.46153846 Editing a category or subcategory

Our system removing the articles

Returns:
0.8333333 "Assigning a category"
0.7777778 "Creating a graph"
0.7777778 "Printing a graph"
Expected:
0.40625 Editing a category or subcategory

Quicken 2007 help file search results

Editing category or subcategory
Tips for saving time
Editing a memorized transaction

16. Query: how to use quicken help

Our first implemented system

Returns:
0.96428573 "Using Quicken Help"
Expected: same as returned result

Returns:
0.96428573 "Using Quicken Help"
Expected: same as returned result

Quicken 2007 help file search results

What's new in Quicken 2007
Miscellaneous Tips
Products and Customer Support

17. Query: input credit card transaction

Our first implemented system

Returns:
0.8125 "Investment tips"
0.8125 "Tax tips"
0.71428573 "Editing a transaction"
Expected:
0.59090906 Entering a credit card transaction

Returns:
1.0 "Editing a transaction"
0.8125 "Investment tips"
0.8125 "Tax tips"
Expected:
0.59090906 Entering a credit card transaction

Quicken 2007 help file search results

No pages with your search words were found

18. Query: customize quicken

Our first implemented system

Returns:
0.8 "Customizing Quicken"
Expected: same as returned result

Our system removing the articles

Returns:
0.8 "Customizing Quicken"
Expected: same as returned result

Quicken 2007 help file search results

What's new in Quicken 2007
Creating a memorized graph
Creating a memorized report

19. Query: create account for online banking

Our first implemented system

Returns:

0.7916667 "Setting up an account for
online banking"

0.78571427 "Changing report
preferences"

0.71428573 "Performing data file
maintenance"

Expected: same as returned result

Our system removing the articles

0.875 "Setting up an account for online
banking"

0.8333333 "Printing the register"

0.78571427 "Changing report
preferences"

Expected: same as returned result

Quicken 2007 help file search results

Setting up an account for online banking

Setting up an account for Web Connect

20. Query: how to register quicken in online

Our first implemented system

Returns:

0.75 "Setting up Insights"

0.625 "Registering Quicken online"

0.60714287 "Estimating year-to-date
interest"

Expected:

0.625 Registering Quicken online

Our system removing the articles

Returns:

0.75 "Setting up Insights"

0.625 "Registering Quicken online"

0.60714287 "Estimating year-to-date
interest"

Expected:

0.625 Registering Quicken online

Quicken 2007 help file search results

What's new in Quicken 2007

Exporting your portfolio to Quicken.com

Reconciling an account