SEMANTIC SEARCH 3000

LSA-BASED RESEARCH TOOL FOR INFORMATION AND DOCUMENT RETRIEVAL

TERRANCE TAUBES (T14)

PROFESSOR BENAJIBA

CSCI 3907/6907 INTRO TO STATISTICAL NLP

GOAL:

To design a research tool that allows students to find the relevant documents within a large collection of text in order to facilitate a quick and direct approach to finding the appropriate information and sources for their research.

Designed to:

- Reduce the time needed for research
- Quickly find the documents most relevant to the information that the user seeks
- Allow students to organize their documents, discern which documents are of interest, and promptly access the text via the user interface

SEMANTIC SEARCH 3000: OVERVIEW

- The Semantic Search 3000 application is a tool that utilizes an array of natural language processing techniques to compute a similarity score between a user's search query and each document within a specified document group.
- Users are able to organize collections of text files into Document Groups, which are directories containing the text files that are to be grouped together.
- Users are then able to enter search queries into the application and find the most relevant documents within the current Document Group.
- Users interact with the Semantic Search 3000 application using its graphical user interface.

API & MODULES

LSA Model Generation

- Gensim

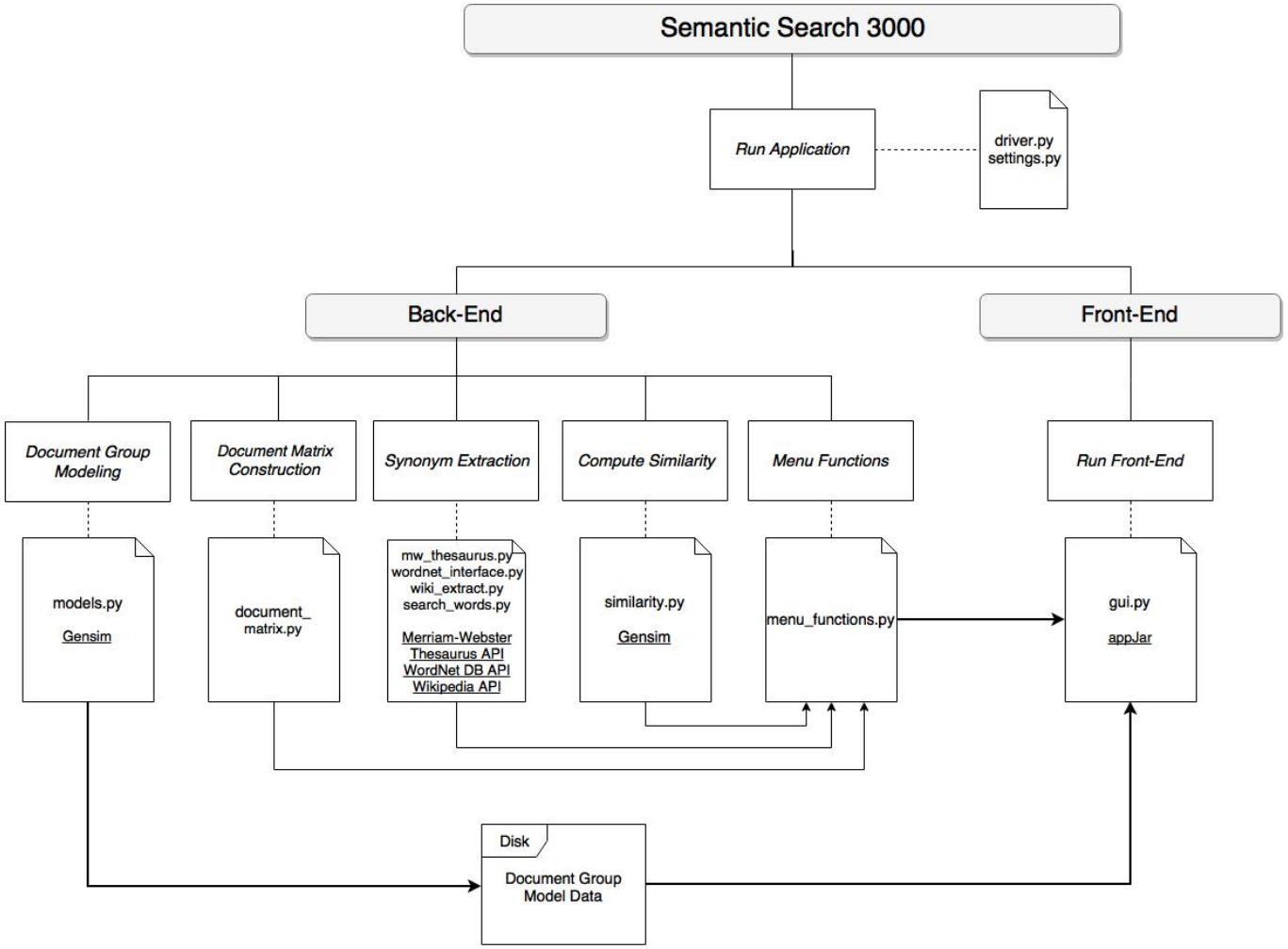
Synonym Extraction

- Merriam-Webster Thesaurus API
- WordNet Database API
- Wikipedia API

Graphical User Interface

- appJar

SEMANTIC SEARCH 3000: DESIGN



APPLICATIONS OF NLP

- Regular Expressions
- Text Normalization (Data Wrangling, Tokenization, Segmentation, Lemmatization)
- External Lexicon and Thesaurus APIs
- Information Extraction and Retrieval
- Latent Semantic Analysis
- Term-Document and Word-Word Matrices
- Term Frequency-Inverse Document Frequency

SEMANTIC SEARCH 3000: FUNCTIONS

Semantic Search 3000 has 3 Major Functions:

1. Search
2. Select Documents
3. Upload Documents

UPLOAD DOCUMENTS

- The ‘Upload Documents’ function allows users to specify a directory of text files as input, which is then used to build a LSA model for the directory and form the Document Group.

SELECT DOCUMENTS

- The ‘Select Documents’ function allows users to specify which Document Group they would like to use, and then loads the Document Group’s model data.

ISSUES WITH INFORMATION RETRIEVAL

- Synonymy – Multiple ways to express or describe the meaning of the same thing
  - Search query words may not be found within a document even though the document is relevant
  - Need a way to include relevant search terms

- Compound Search Terms – "New York", "Shake Shack", "Machine Learning"
  - Search results find matches based on individual word matches and not matches of the whole concept
  - Query = "candy apple", Doc1 = {"candy" : 9, "apple" : 0, "candy apple": 0}, Doc2 = {"candy" : 1, "apple" : 1, "candy apple" : 1}
  - Need a way to add to the similarity scores for documents that contain compound matches

SEARCH

Search Pipeline:

1. Get user query
2. Preprocess query (lowercase, RegEx to remove punctuation and clitics, lemmatization)
3. Get list of synonyms for query words from APIs.
4. Build Term-Document Matrices for query words and API synonyms.
5. Get Word-Word Matrix counts
6. [Similarity Scoring Function]
7. Sort Documents by Relevancy

LATENT SEMANTIC ANALYSIS

- Latent Semantic Analysis (LSA) is a language processing technique that is able to find the semantic, or underlying meanings of text and to represent these semantic values in the form of vectors.
- Similar words and topics will be represented by the LSA model with similar vectors.
- Words and topics appearing within similar contexts will also be represented with similar vectors.
  - *Synonymy*

LATENT SEMANTIC ANALYSIS

- The similarity between two vector representations can be computed by taking the cosine of the vectors, returning a value between (-1, +1), with a value of +1 meaning the vectors are identical.
- The **foundation** of the similarity scores computed between queries and documents is based on the cosine similarity value taken between the vector representation of the search query and the vector representation of the Document Group.
- Using LSA as the foundation of the similarity scoring helps give high weightings towards documents found to be semantically similar and low weightings to documents that are not, essentially eliminating the irrelevant documents from the search at the beginning.

QUERY WORD-DOCUMENT TITLE MATCHING

- The Query Word-Document Title matching function adds positive weighting to documents that have query words within their titles.

TERM-DOCUMENT MATRIX & TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY

- A Term-Document Matrix is composed of search words as row entries and documents as column entries.
- The cells for each corresponding (word, document) pair contain the frequency for that word in that particular document.
- The frequencies of query word matches is summed for each document (the values in each document column are summed up) and then divided by that document's total number of tokens, returning the TF-IDF values that are to be added to each document's similarity score.

WORD-WORD MATRIX

- A Word-Word matrix is a matrix with both the rows and columns represented by the query words.
- A Word-Word matrix is constructed for each document in the Document Group.
- The cells for each corresponding (word, word) pair contain the number of times each pair of query words appears within a document.
  - *Compound Search Terms*
- All the values within a document's Word-Word matrix are summed and divided by the total number of query word combinations, returning the Word-Word frequency values that are to be added to the document's similarity score.

SIMILARITY SCORING FUNCTION (HIGH-LEVEL DESCRIPTION)

```
for (doc in Document Group):
    doc_score =    LSA_Similarity(query words, Document Group model) / float(2)
    doc_score +=  Title_Similarity(query words) / float(2)
    doc_score +=  TFIDF_Similarity(query word  Term-Document matrix) / float(2)
    doc_score +=  TFIDF_API_Similarity(API synonyms  Term-Document matrix) / float(6)
    doc_score +=  WW_Similarity(query word  Word-Word matrix) / float(5)
```

BACK-END OUTPUT

```
Results between Vector Space of the Query and Vector Space of the Document Group
[195, 'Language technology.txt', 0.97754514]
[121, 'Noam Chomsky.txt', 0.97013515]
[234, 'Alan Turing.txt', 0.96761435]
[61, 'IBM.txt', 0.96699286]
[63, 'Natural language understanding.txt', 0.96594411]
[237, 'Corpus linguistics.txt', 0.96334791]
[69, 'Natural language.txt', 0.96215558]
[107, 'Natural speech.txt', 0.96215558]
[148, 'History of natural language processing.txt', 0.96088153]
[172, 'Journal of Machine Learning Research.txt', 0.96061355]
[47, 'Inductive programming.txt', 0.95817387]
[99, 'Computer-assisted translation.txt', 0.95809698]
[97, 'Machine ethics.txt', 0.95332015]
[52, 'Spoken dialogue system.txt', 0.95318896]
[142, 'World Wide Web.txt', 0.9527775]
[178, 'List of important publications in computer science.txt', 0.95178699]
[100, 'Speech synthesis.txt', 0.95168352]
[109, 'Text-to-speech.txt', 0.95168352]
[208, 'List of natural language processing toolkits.txt', 0.95124704]
[238, 'Internet.txt', 0.95053452]
```

```
Results of matching Query words within Document titles:
['Statistical machine translation.txt', 1.0]
['Rule-based machine translation.txt', 0.6666666666666666]
['Example-based machine translation.txt', 0.6666666666666666]
['Machine translation.txt', 0.6666666666666666]
['Brain machine interface.txt', 0.3333333333333333]
['Automatic translation.txt', 0.3333333333333333]
['Journal of Machine Learning Research.txt', 0.3333333333333333]
['Azure machine learning studio.txt', 0.3333333333333333]
['Machine Learning (journal).txt', 0.3333333333333333]
['Machine learning.txt', 0.3333333333333333]
['Statistical models.txt', 0.3333333333333333]
['Statistical inference.txt', 0.3333333333333333]
['Computer-assisted translation.txt', 0.3333333333333333]
['Machine ethics.txt', 0.3333333333333333]
['Amazon Machine Learning.txt', 0.3333333333333333]
['Machine learning control.txt', 0.3333333333333333]
['List of machine learning algorithms.txt', 0.3333333333333333]
['List of datasets for machine learning research.txt', 0.3333333333333333]
['International Conference on Machine Learning.txt', 0.3333333333333333]
['Machine perception.txt', 0.3333333333333333]
```

BACK-END OUTPUT

Results of the term frequency-inverse document frequency calculation for Query words :

[222, 'Example-based machine translation.txt', 0.074509803921568626]
[201, 'Statistical machine translation.txt', 0.061801446416831031]
[17, 'Machine perception.txt', 0.055384615384615386]
[99, 'Computer-assisted translation.txt', 0.054223149113660066]
[191, 'Automatic translation.txt', 0.053216103655714948]
[19, 'Machine translation.txt', 0.053216103655714948]
[16, 'Computational statistics.txt', 0.047619047619047616]
[172, 'Journal of Machine Learning Research.txt', 0.035714285714285712]
[9, 'Stochastic grammar.txt', 0.031531531531531529]
[24, 'International Conference on Machine Learning.txt', 0.029268292682926831]
[148, 'History of natural language processing.txt', 0.029126213592233011]
[125, 'Statistical models.txt', 0.026554856743535988]
[230, 'Rule-based machine translation.txt', 0.025000000000000001]
[147, 'Mlpy.txt', 0.022950819672131147]
[120, 'Statistical inference.txt', 0.022288261515601784]
[151, 'Machine Learning (journal).txt', 0.022034996759559299]
[134, 'Machine learning.txt', 0.022034996759559299]
[58, 'Amazon Machine Learning.txt', 0.022034996759559299]
[39, 'Active learning (machine learning).txt', 0.022034996759559299]
[32, 'List of machine learning algorithms.txt', 0.022034996759559299]

Results of the term frequency-inverse document frequency calculation for API words

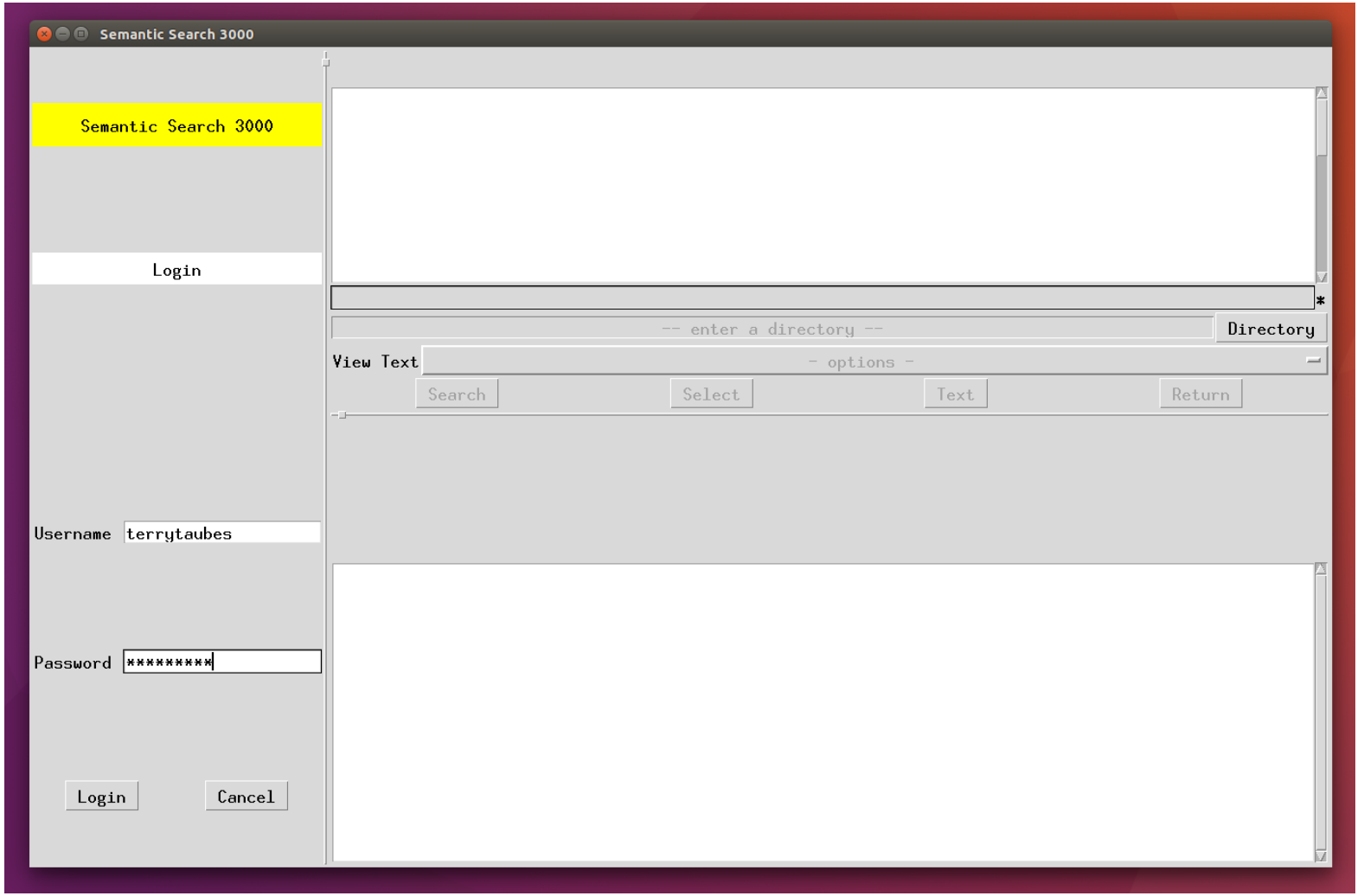
[195, 'Language technology.txt', 0.012903225806451613]
[172, 'Journal of Machine Learning Research.txt', 0.01020408163265306]
[16, 'Computational statistics.txt', 0.0090702947845804991]
[3, 'Computation.txt', 0.0088495575221238937]
[122, 'Handwriting recognition.txt', 0.0078175895765472316]
[231, 'Speech corpus.txt', 0.0071684587813620072]
[177, 'Predictive text.txt', 0.0068352699316472991]
[107, 'Natural speech.txt', 0.0063965884861407248]
[69, 'Natural language.txt', 0.0063965884861407248]
[63, 'Natural language understanding.txt', 0.0056497175141242938]
[98, 'Word2vec.txt', 0.0050403225806451612]
[92, 'Parsing.txt', 0.0050038491147036184]
[223, 'Relationship extraction.txt', 0.0050000000000000001]
[14, 'Automated theorem proving.txt', 0.0048449612403100775]
[139, 'Computational learning theory.txt', 0.0047694753577106515]
[173, 'Evolutionary algorithm.txt', 0.0047483380816714148]
[82, 'Multi-label classification.txt', 0.0046728971962616819]
[159, 'Inductive bias.txt', 0.0046403712296983757]
[9, 'Stochastic grammar.txt', 0.0045045045045045045]
[40, 'Algorithm design.txt', 0.0044843049327354259]

BACK-END OUTPUT

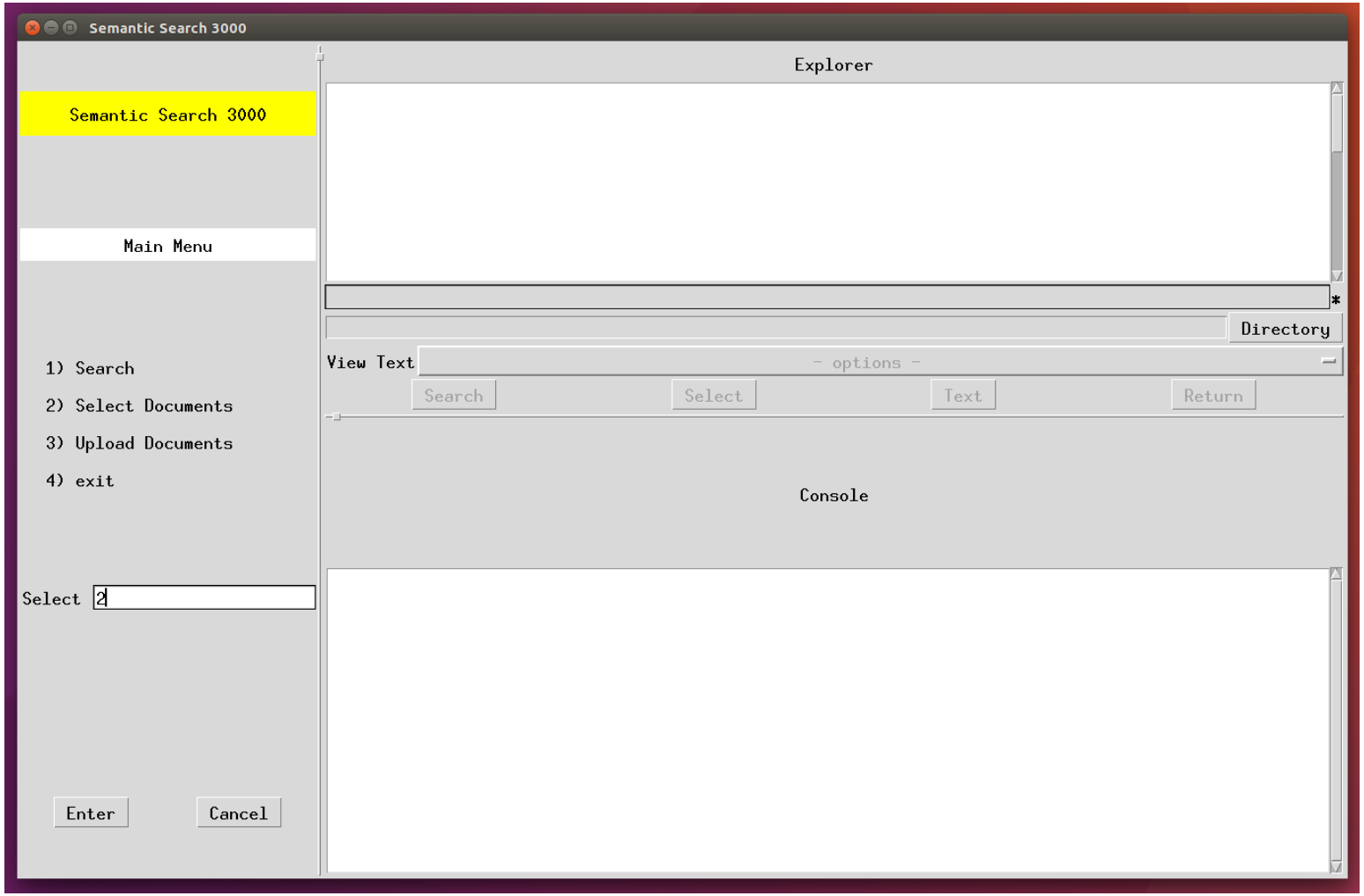
```
Results of the co-occurrence counts per document, normalized by the number of Query
word combinations
[237, 'Corpus linguistics.txt', 1.0]
[230, 'Rule-based machine translation.txt', 1.0]
[227, 'Computer science.txt', 1.0]
[222, 'Example-based machine translation.txt', 1.0]
[216, 'Automatic summarization.txt', 1.0]
[208, 'List of natural language processing toolkits.txt', 1.0]
[204, 'Hidden Markov models.txt', 1.0]
[201, 'Statistical machine translation.txt', 1.0]
[191, 'Automatic translation.txt', 1.0]
[188, 'Information extraction.txt', 1.0]
[178, 'List of important publications in computer science.txt', 1.0]
[171, 'Named entity recognition.txt', 1.0]
[156, 'Artificial intelligence.txt', 1.0]
[148, 'History of natural language processing.txt', 1.0]
[119, 'Latent semantic indexing.txt', 1.0]
[117, 'Deep learning.txt', 1.0]
[93, 'Word-sense disambiguation.txt', 1.0]
[91, 'Agglutination.txt', 1.0]
[88, 'Text corpus.txt', 1.0]
[78, 'Computational linguistics.txt', 1.0]
```

```
Documents by Calculated Score
[201, 'Statistical machine translation.txt', 1.0]
[222, 'Example-based machine translation.txt', 0.98538132480811447]
[19, 'Machine translation.txt', 0.952297410058937742]
[191, 'Automatic translation.txt', 0.9006409214296428]
[230, 'Rule-based machine translation.txt', 0.89966268347519518]
[148, 'History of natural language processing.txt', 0.81463835501932647]
[88, 'Text corpus.txt', 0.790366189452104681]
[208, 'List of natural language processing toolkits.txt', 0.786307319990144471]
[11, 'Terminology extraction.txt', 0.782982981247468681]
[156, 'Artificial intelligence.txt', 0.781405529033805221]
--------------------
```

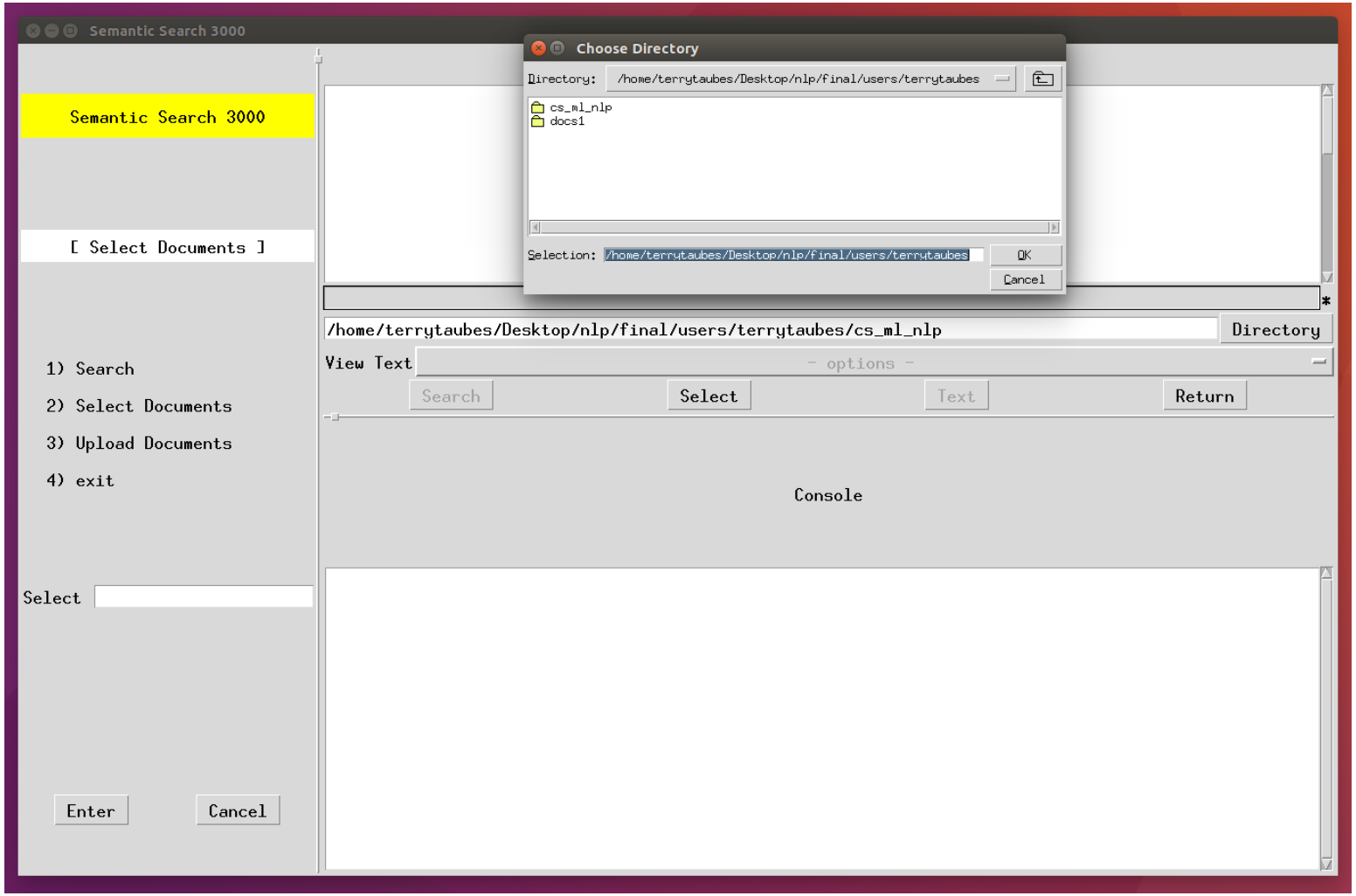
DEMO: LOGIN


Semantic Search 3000
Semantic Search 3000
Login
Username terrytaubes
Password *********
Login
Cancel
-- enter a directory --
Directory
View Text
- options -
Search
Select
Text
Return

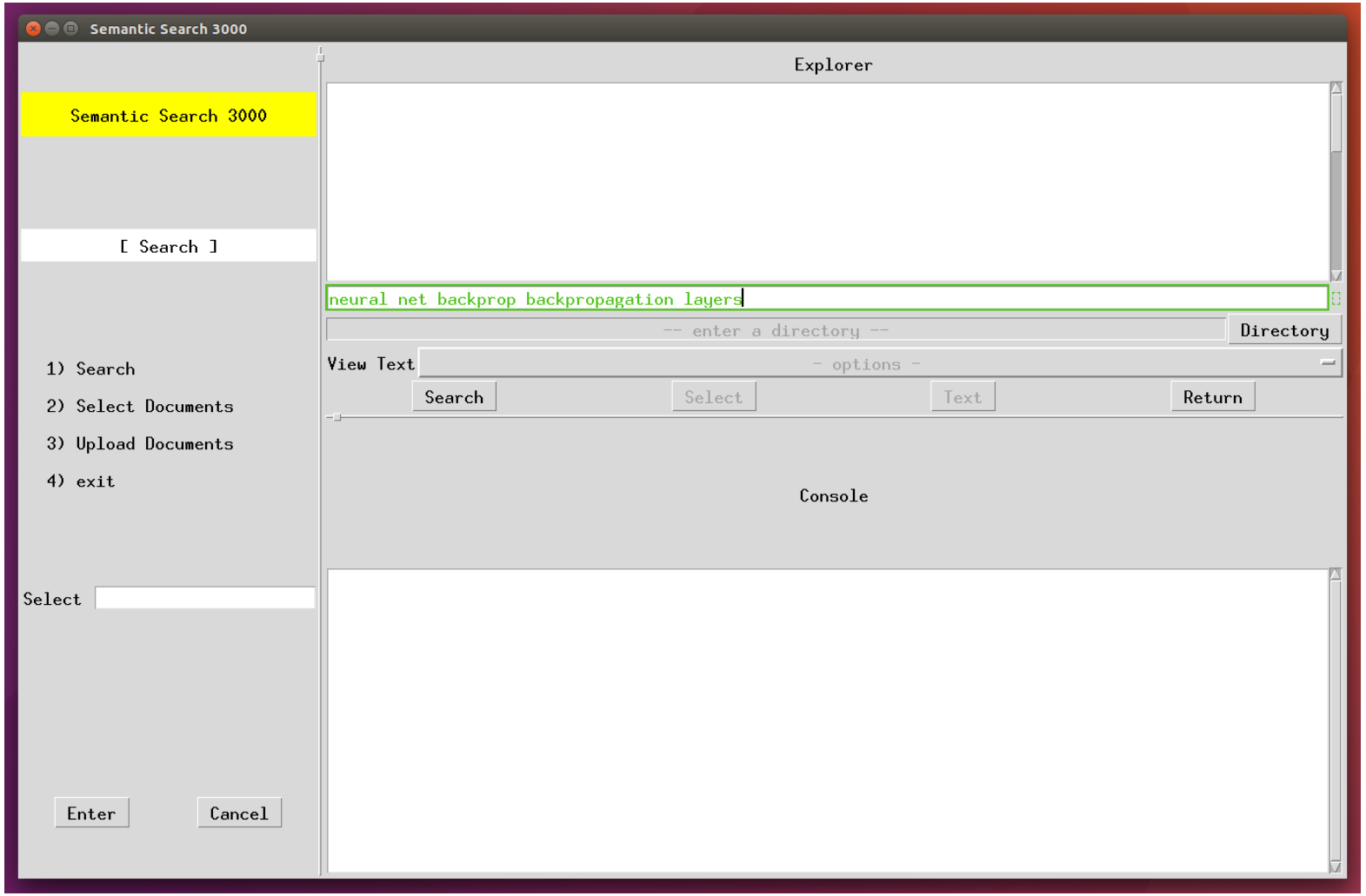
DEMO: MAIN MENU



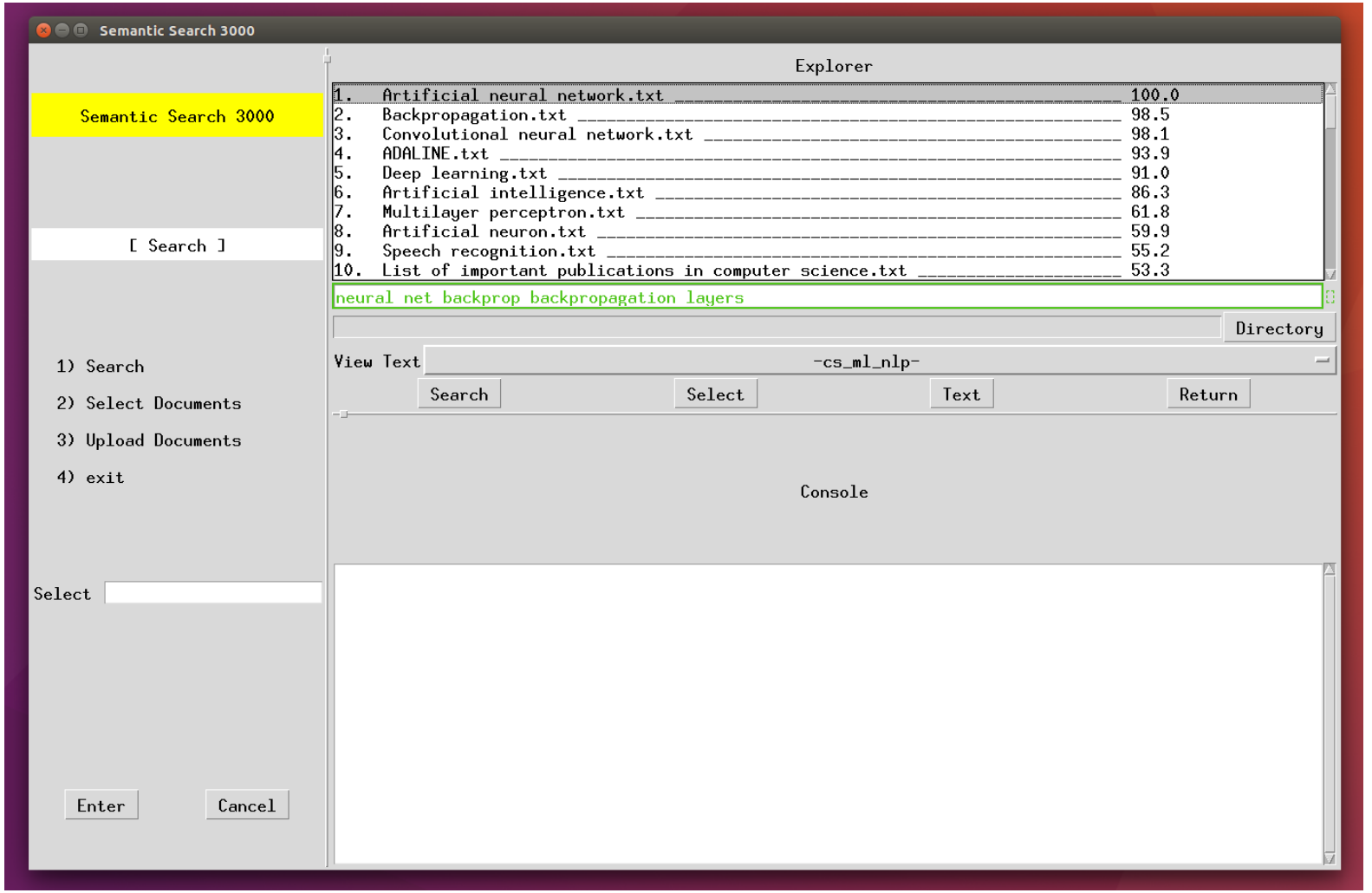
DEMO: SELECT/UPLOAD DOCUMENTS



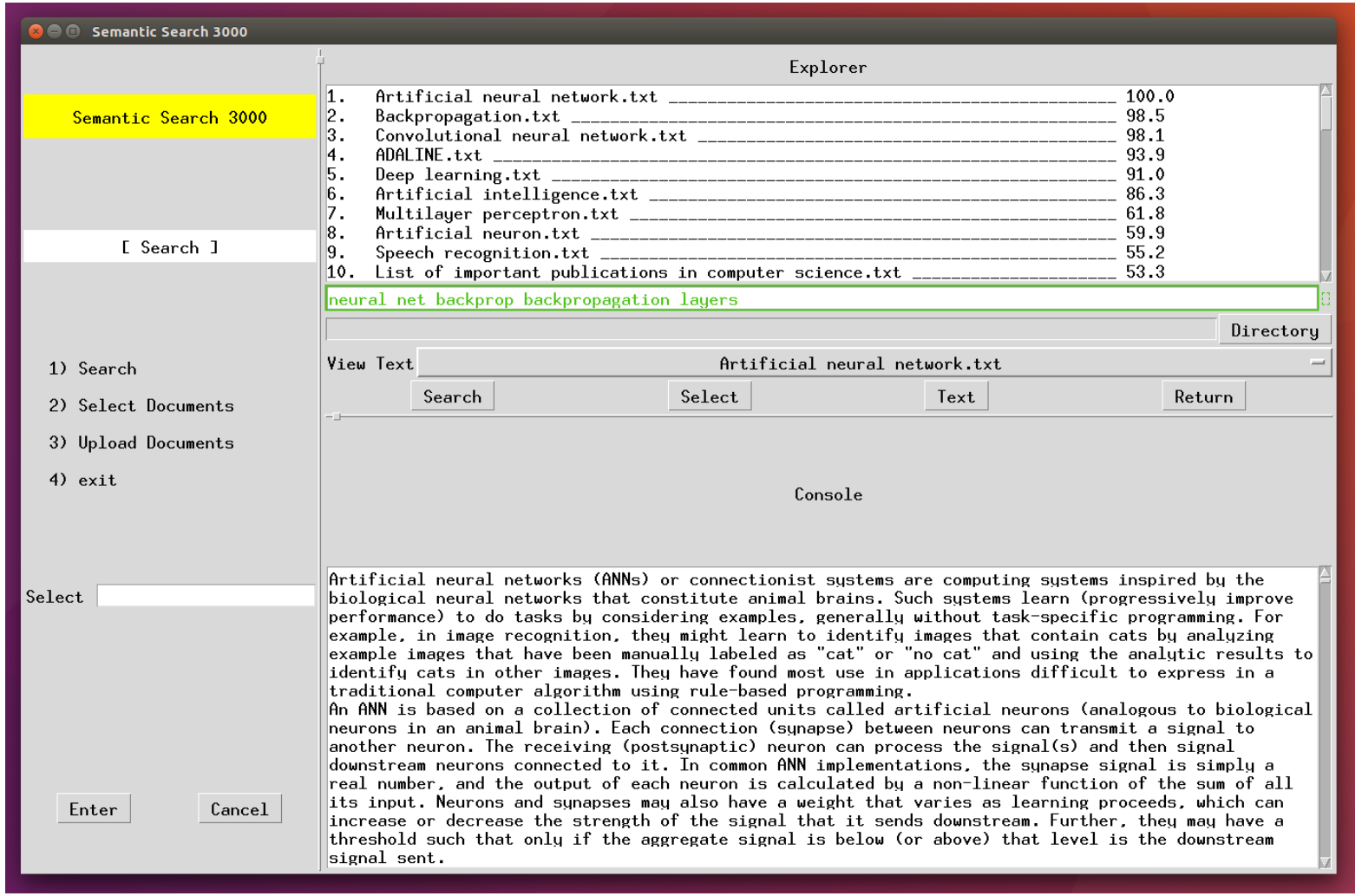
DEMO: SEARCH


Semantic Search 3000
Semantic Search 3000
[Search]
1) Search
2) Select Documents
3) Upload Documents
4) exit
Select
Enter
Cancel
Explorer
neural net backprop backpropagation layers
-- enter a directory --
Directory
View Text
- options -
Search
Select
Text
Return
Console

DEMO: SEARCH RESULTS



DEMO: VIEW TEXT



SEMANTIC SEARCH 3000

Thank You!

SLIDES

- [2 - 3] S
- [4 - 5] T
- [6 - 9] S
- [10 - 18] T