# How to connect to the Aviationknowledge LSA Tools

Richard Brown
richard.uoc@gmail.com
6/4/2010 (Version 2)
For use on http://aviationknowledge.colorado.edu

# Table of Contents

## 1. What you should already know

This tutorial assumes the reader is familiar with Latent Semantic Analysis (LSA) and the Cognitive Walkthrough as it applies to Aviation. If not, a review of the LSA papers at http://lsa.colorado.edu/, the AutoCWW Tutorial http://autocww2.colorado.edu/~blackmon/Tutorials/AutoCWWTutorialA.pdf, and the papers on the Cognitive Walkthrough http://autocww2.colorado.edu/~blackmon/Papers.html are suggested reading before continuing.

This tutorial will walk the reader through the process of creating a valid GET Uniform Resource Locator (URL) for use on http://aviantionknowledge.colorado.edu and give details and examples for each LSA tool. The appendices contain an ASCII chart for easy reference, the filtering process performed on the server, and finally examples of how to connect to the server in various programming languages.

## 2. URL Query String

The method to connect to the LSA tools from your program is through a HTTP request using the GET method using a query string. A Query string is simply the part of the URL that contains data to be passed to the tools. A typical URL contains the server name and path (http://server/path) such as http://aviationknowledge.colorado.edu/~pilotlsa/. A URL with a query string contains the name of the program to run as well as the data to be passed (http://server/path/program?query_string) such as http://aviationknowledge.colorado.edu/cgi-bin/termVectors.cgi?Web=1&Space=ExpertPilot_v3&Links=Navigation%20Waypoint%0D%0ADirection

The question mark is used to separate the program name and the query string and is not passed to the program. The query string is composed of field-data pairs separated by an equal sign and each pair is separated by the ampersand.

Some characters cannot be part of the URL and will need to be converted. It is important to only convert (encode) the data portion of the field-data pairs. Specifically, letters (A-Z and a-z), numbers (0-9), period, dash, tilde, and underscore can sent without being encoded. The space is encoded as a '+' or %20 and all other characters must be encoded as %FF hex representation. It is suggested that you filter out some characters before creating your query string. The most important character to remove is the apostrophe as it can cause issues resulting in your program not working properly. Appending B covers the filtering process used by the LSA tools and it is suggested you

perform similar filtering. Section 6.a covers in more detail how to perform the required encoding on the data portion of the field-data pairs.

For our purposes, a valid URL and query string will contain the server name aviationknowledge.colorado.edu, the path /cgi-bin/, the program name, the web parameter, the Space parameter, and other program specific parameters. The next sections will cover the available programs and spaces as well as the options for the web and other parameters.

## 3. LSA Tools program names

Table 1 contains the program names for each of the LSA Tools. The program names are case sensitive. Each program requires a specific set of parameters, which is covered in section 7.

| LSA Tool | Program name |
|---|---|
| Unfamiliar headings/links analysis | termVectors.cgi |
| Confusable headings/links analysis | nph-matrix.cgi |
| Goal-specific competing headings/links analysis | nph-elaborate.cgi |
| Frequency Analysis | nph-findfrequency.cgi |
| Low Frequency Words Analysis | nph-findParaFreq.cgi |
| Elaborate Links | nph-elaborate1.cgi |
| One to Many Comparison | nph-one2many.cgi |
| Coherence Tool | nph-coherenceCheck.cgi |
| Paragraph to Paragraph Coherence Tool | nph-PCoherenceCheck.cgi |

Table 1: List of available programs

## 4. Web Parameter

Each of the LSA tools requires the usage of the Web parameter. This parameter takes only two values, 0 or 1. When the Web parameter is set as 1, the script outputs the results in HTML. The HTML output is in the same format as seen when using the LSA tools from the http://aviationknowledge.colorado.edu/HomePage.html website. If the Web parameter is set to 0, only the results are sent with no formatting other than a "\n" (newline) or a "\t" (tab) to separate the results.

## 5. Spaces Parameter

The aviationknowledge LSA tools currently support 6 semantic spaces. The table below contains the names of all the spaces. Please note, the names are case sensitive and must be added to the URL as shown.

| Space | Description |
|---|---|
| CDUskills_v3 | CDUskills corpus |
| CDUskills_v4 | CDUskills_v3 with improved multi-word support |
| CDUskills_v5 | CDUskils_v4 with single character letters and numbers removed |
| ExpertPilot_v3 | ExpertPilot corpus |
| ExpertPilot_v4 | ExpertPilot_v3 with improved multi-word support |
| ExpertPilot_v5 | ExpertPilot_v4 with single character letters and numbers removed |

Table 2: List of Semantic Spaces

For more information on the spaces, visit http://aviationknowledge.colorado.edu/~pilotlsa/.

# 6. Other Parameters

For all other parameters, the LSA tools require the links or words to be separated by a blank line (CR/LF) and are encoded as %0D%0A. Two distinct links, Love and Hope, are encoded as "Love%0D%0AHope". In some cases, such as when submitting sentences or paragraphs, two CR/LFs (%0D%0A) are needed to perform the separation. These cases are individually identified in section 7. As noted in section 2, the data in the field-data pair must be encoded.

## a. URL Encoding

### What is URL Encoding?

URL Encoding allows for the conversion of certain characters in an URL into a triplet representation of the converted character. The triplet consists of a percent character "%" followed by two hexadecimal digits.

The character is replaced with the triplet representation when the character is outside the set of characters allowed to be in a URL. The character is also replaced when the character corresponds to a reserved character used by the system to for some other purpose.

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | - | _ | . | ~ | | | | | | | | | | | | |

Table 3: List of Unreserved Characters

Table 3: List of Unreserved Characters represents the list of acceptable unreserved characters for submission in an URL While RFC 3986 [1] allows for more characters

not included in this list, some systems can modify the characters resulting in unpredictable behavior. All characters not represented in the table will have to be encoded.

### How are characters URL encoded?

The encoding uses the two-digit hexadecimal representation of the code point of a character set preceded by a percent character "%". For example, the character "Š" in the ISO 859-15 character set has a code point of 8A and is encoded as %8A.

More information on characters sets and the ISO 8859 character sets can be found at:

> The ISO 8859 Alphabet Soup -http://czyborra.com/charsets/iso8859.html
>
> Character Set Tables – http://www.columbiauniversity.org/kermit/csettables.html

Appendix A contains an ASCII chart with the hexadecimal value assigned to each character and symbol.

### Example

Take for instance the test string "This is a % * simple & short + test." Notice the string contains several characters, which must be encoded. After encoding the test string converts to "This%20is%20a%20%25%20*%20simple%20%26%20short%20%2B%20test."

To test your own strings, you can use an online encoder/decoder such as http://meyerweb.com/eric/tools/dencoder/.

### Language Support

Most languages with Web support contain built-in functions to encode and decode a String. Below is a brief list of the languages and their supported functions.

| Language | Encoding | Decoding |
|---|---|---|
| .Net | HttpUtility.UrlEncode http://msdn.microsoft.com/en-us/library/system.web.httputility.urlencode%28VS.71%29.aspx | HttpUtility.UrlDecode http://msdn.microsoft.com/en-us/library/system.web.httputility.urldecode%28VS.71%29.aspx |
| JavaScript | escape(String) http://www.w3schools.com/jsref/jsref_escape.asp | unescape(String) http://www.w3schools.com/jsref/jsref_unescape.asp |
| Perl | uri_escape http://search.cpan.org/~mrjc/cvswebedit-v2.0b1/cvs-web/lib/URI/Escape.pm | uri_unescape http://search.cpan.org/~mrjc/cvswebedit-v2.0b1/cvs-web/lib/URI/Escape.pm |
| PHP | urlencode(String) http://php.net/manual/en/function.urlencode.php | urldecode(String) http://php.net/manual/en/function.urlencode.php |

## 7. Putting it all together – URL examples for each LSA Tool

From the previous section, a valid URL with a query string contains the server name, path, program name, web parameter, space parameter, and other parameters in the format below.
http://servername/path/program_name?Web_paramter&Space_parameter&other_parameters. All parameters are in the field-data pair format separated by an ampersand. Each of the LSA programs requires a specific set of parameters and are listed below.

### Unfamiliar headings/links analysis

Script: termVectors.cgi

Parameters:

Space – A valid space name from section 5
Web – 0 (no HTML output) or 1 (HTML output)
Links – separated by %0D%0A

Example:

http://aviationknowledge.colorado.edu/cgi-bin/termVectors.cgi?Web=1&Space=ExpertPilot_v3&Links=Navigation%20Waypoint%0D%0ADirection

### Frequency Analysis

Script: nph-findfrequency.cgi

Parameters:

Space – A valid space name from section 5
Web – 0 (no HTML output) or 1 (HTML output)
Words – separated by %0D%0A

Example:

http://aviationknowledge.colorado.edu/cgi-bin/nph-findfrequency.cgi?Web=1&Space=ExpertPilot_v3&Words=Navigation%0D%0AWaypoint

### Confusable headings/links Analysis

Script: nph-matrix.cgi

Parameters:

Space – A valid space name from section 5
Web – 0 (no HTML output) or 1 (HTML output)
Links – separated by %0D%0A%0D%0A

Example:

http://aviationknowledge.colorado.edu/cgi-bin/nph-matrix.cgi?Web=1&Space=ExpertPilot_v3&Links=Navigation%20Waypoint%0D%0ADirection

### Goal-specific competing headings/links analysis

Script: nph-elaborate.cgi

Parameters:

Space – A valid space name from section 5
Web – 0 (no HTML output) or 1 (HTML output)

Links – separated by %0D%0A
Frequency – Typically 50
Cosine – i.e. .25
Goal

Example:

http://aviationknowledge.colorado.edu/cgi-bin/nph-elaborate.cgi?Space=ExpertPilot_v3&Frequency=50&Cosine=.25&Goal=The+website+offers+information+and+documentation+to+support+your+analysis.&Links=Documents%0D%0A%0D%0AQuestions&Web=1

### Low Frequency Words Analysis

Script: nph-findParaFreq.cgi

Parameters:
Space – A valid space name from section 5
Web – 0 (no HTML output) or 1 (HTML output)
FreqRange – Flags words with frequency less than or equal to set value
Para – Paragraphs separated by %0D%0A%0D%0A

Example:

http://aviationknowledge.colorado.edu/cgi-bin/nph-findParaFreq.cgi?FreqRange=15&Space=ExpertPilot_v3&Web=1&Para=Unfortunately%2C+for+the+most+part%2C+newspapers+seem+to+look+down+on+%22curating%22+as+if+it%27s+some+sort+of+lesser+form+of+journalism%2C+and+this+is+a+sticking+point+that+they%27re+going+to+need+to+get+past+if+they+want+to+understand+how+people+engage+with+the+news+today.+These+days%2C+everyone+is+a+curator+of+the+news+in+some+fashion%3A+they+share+news%2C+comment+on+it%2C+post+about+it%2C+etc.%0D%0A%0D%0AI%27m+not+denying+that+there+is+some+resentment+out+there+of+successful+people.+There+are+always+some+people+who+are+resentful+of+others%2C+but+I+just+don%27t+see+that+as+a+driving+force+in+the+criticism+of+content+creators+who+choose+a+path+that+is+anti-fan.

### Elaborating Links

Script: nph-elaborate1.cgi

Parameters:
Space – A valid space name from section 5

Web – 0 (no HTML output) or 1 (HTML output)
Links – separated by %0D%0A%0D%0A
Frequency – Typically 50
Cosine – i.e. .50

Example:

http://aviationknowledge.colorado.edu/cgi-bin/nph-elaborate1.cgi?Space=ExpertPilot_v3&Frequency=50&Cosine=.50&Links=Directions%0D%0A%0D%0AUnknown+Words&Web=1

### One to Many Comparison

Script: nph-one2many.cgi

Parameters:
Space – A valid space name from section 5
Web – 0 (no HTML output) or 1 (HTML output)
Links – separated by %0D%0A%0D%0A
Goal

Example:

http://aviationknowledge.colorado.edu/cgi-bin/nph-one2many.cgi?Space=ExpertPilot_v3&Goal=I%27m+not+denying+that+there+is+some+resentment+out+there+of+successful+people.+There+are+always+some+people+who+are+resentful+of+others%2C+but+I+just+don%27t+see+that+as+a+driving+force+in+the+criticism+of+content+creators+who+choose+a+path+that+is+anti-fan.&Links=Fans%0D%0A%0D%0ALoyal%0D%0A%0D%0AMoney&Web=1

### Coherence Tool

Script: nph-coherenceCheck.cgi

Parameters:
Space – A valid space name from section 5
Web – 0 (no HTML output) or 1 (HTML output)
Links – No separation required.

Example:

http://aviationknowledge.colorado.edu/cgi-bin/nph-coherenceCheck.cgi?Space=ExpertPilot_v3&Links=I%27m+not+denying+that+there

+is+some+resentment+out+there+of+successful+people.+There+are+always+some+people+who+are+resentful+of+others%2C+but+I+just+don%27t+see+that+as+a+driving+force+in+the+criticism+of+content+creators+who+choose+a+path+that+is+anti-fan.&Web=1

### Paragraph to Paragraph Coherence Tool

Script: nph-PCoherenceCheck.cgi

Parameters:

Space – A valid space name from section 5
Web – 0 (no HTML output) or 1 (HTML output)
Links – Paragraphs separated by %0D%0A%0D%0A

Example:

http://aviationknowledge.colorado.edu/cgi-bin/nph-PCoherenceCheck.cgi?Space=ExpertPilot_v3&Web=1&Links=I%27m+not+denying+that+there+is+some+resentment+out+there+of+successful+people.+There+are+always+some+people+who+are+resentful+of+others%2C+but+I+just+don%27t+see+that+as+a+driving+force+in+the+criticism+of+content+creators+who+choose+a+path+that+is+anti-fan.+%0D%0A%0D%0AComing+at+the+same+question+from+the+other+direction%2C+again%2C+I+have+trouble+seeing+%22resentment%22+as+the+issue+at+all.+When+we+look+at+the+success+stories%2C+the+one+thing+that+comes+through+loud+and+clear+is+that+respecting+fans+results+in+those+fans+becoming+incredibly+loyal.+They%27re+loyal+to+a+fault%2C+in+fact.+There%27s+no+resentment+there+at+all.+If+anything%2C+at+times%2C+it+seems+to+border+on+hero+worship.

## References

[1] Berners-Lee, T. January 2005. Uniform Resource Identifier (URI): Generic Syntax. Retrieved February 15, 2010. http://tools.ietf.org/html/rfc3986

## Appendix A: ASCII Chart

| Ctrl | Dec | Hex | Char | Code |
|---|---|---|---|---|
| ^@ | 0 | 00 | | NUL |
| ^A | 1 | 01 | | SOH |
| ^B | 2 | 02 | | STX |
| ^C | 3 | 03 | | ETX |
| ^D | 4 | 04 | | EOT |
| ^E | 5 | 05 | | ENQ |
| ^F | 6 | 06 | | ACK |
| ^G | 7 | 07 | | BEL |
| ^H | 8 | 08 | | BS |
| ^I | 9 | 09 | | HT |
| ^J | 10 | 0A | | LF |
| ^K | 11 | 0B | | VT |
| ^L | 12 | 0C | | FF |
| ^M | 13 | 0D | | CR |
| ^N | 14 | 0E | | SO |
| ^O | 15 | 0F | | SI |
| ^P | 16 | 10 | | DLE |
| ^Q | 17 | 11 | | DC1 |
| ^R | 18 | 12 | | DC2 |
| ^S | 19 | 13 | | DC3 |
| ^T | 20 | 14 | | DC4 |
| ^U | 21 | 15 | | NAK |
| ^V | 22 | 16 | | SYN |
| ^W | 23 | 17 | | ETB |
| ^X | 24 | 18 | | CAN |
| ^Y | 25 | 19 | | EM |
| ^Z | 26 | 1A | | SUB |
| ^[ | 27 | 1B | | ESC |
| ^\ | 28 | 1C | | FS |
| ^] | 29 | 1D | | GS |
| ^^ | 30 | 1E | ▲ | RS |
| ^- | 31 | 1F | ▼ | US |

| Dec | Hex | Char |
|---|---|---|
| 32 | 20 | |
| 33 | 21 | ! |
| 34 | 22 | " |
| 35 | 23 | # |
| 36 | 24 | $ |
| 37 | 25 | % |
| 38 | 26 | & |
| 39 | 27 | ' |
| 40 | 28 | ( |
| 41 | 29 | ) |
| 42 | 2A | * |
| 43 | 2B | + |
| 44 | 2C | , |
| 45 | 2D | - |
| 46 | 2E | . |
| 47 | 2F | / |
| 48 | 30 | 0 |
| 49 | 31 | 1 |
| 50 | 32 | 2 |
| 51 | 33 | 3 |
| 52 | 34 | 4 |
| 53 | 35 | 5 |
| 54 | 36 | 6 |
| 55 | 37 | 7 |
| 56 | 38 | 8 |
| 57 | 39 | 9 |
| 58 | 3A | : |
| 59 | 3B | ; |
| 60 | 3C | < |
| 61 | 3D | = |
| 62 | 3E | > |
| 63 | 3F | ? |

| Dec | Hex | Char |
|---|---|---|
| 64 | 40 | @ |
| 65 | 41 | A |
| 66 | 42 | B |
| 67 | 43 | C |
| 68 | 44 | D |
| 69 | 45 | E |
| 70 | 46 | F |
| 71 | 47 | G |
| 72 | 48 | H |
| 73 | 49 | I |
| 74 | 4A | J |
| 75 | 4B | K |
| 76 | 4C | L |
| 77 | 4D | M |
| 78 | 4E | N |
| 79 | 4F | O |
| 80 | 50 | P |
| 81 | 51 | Q |
| 82 | 52 | R |
| 83 | 53 | S |
| 84 | 54 | T |
| 85 | 55 | U |
| 86 | 56 | V |
| 87 | 57 | W |
| 88 | 58 | X |
| 89 | 59 | Y |
| 90 | 5A | Z |
| 91 | 5B | [ |
| 92 | 5C | \ |
| 93 | 5D | ] |
| 94 | 5E | ^ |
| 95 | 5F | _ |

| Dec | Hex | Char |
|---|---|---|
| 96 | 60 | ` |
| 97 | 61 | a |
| 98 | 62 | b |
| 99 | 63 | c |
| 100 | 64 | d |
| 101 | 65 | e |
| 102 | 66 | f |
| 103 | 67 | g |
| 104 | 68 | h |
| 105 | 69 | i |
| 106 | 6A | j |
| 107 | 6B | k |
| 108 | 6C | l |
| 109 | 6D | m |
| 110 | 6E | n |
| 111 | 6F | o |
| 112 | 70 | p |
| 113 | 71 | q |
| 114 | 72 | r |
| 115 | 73 | s |
| 116 | 74 | t |
| 117 | 75 | u |
| 118 | 76 | v |
| 119 | 77 | w |
| 120 | 78 | x |
| 121 | 79 | y |
| 122 | 7A | z |
| 123 | 7B | { |
| 124 | 7C | \| |
| 125 | 7D | } |
| 126 | 7E | ~ |
| 127 | 7F | ⌂* |

* ASCII code 127 has the code DEL. Under MS-DOS, this code has the same effect as ASCII 8 (BS). The DEL code can be generated by the CTRL + BKSP key.

From: http://msdn.microsoft.com/en-us/library/60ecse8t(v=VS.71).aspx

# Appendix B: Characters filtered by the LSA tools on aviationknowledge.colorado.edu

The user input is pre-processed through a set of filters before submitting the requests to the LSA tools. This ensures the security of the system as well as aligns the user submitted data to the same standards as the space. The filtering process is performed in this order:

- All characters are converted to lower case
- All commas are removed from numbers such that 40,000 is transformed into 40000
- All periods are removed and replaced by a space. Please note that numbers like “4.5” will be transformed to “4 5” with a space between the number 4 and 5.
- All commas are removed and replaced by a space. This is performed after commas are removed from numbers.
- All dashes ‘-‘ and underscores ‘_’ are removed with no substitution. A word containing one of these characters, such as “vertical_flight” or “vertical-flight” becomes “verticalflight”.
- All instances of the basic possessive apostrophe, ’s, is replaced by a space. e.g. “cat’s” is replaced by “cat”
- All other instances of the apostrophe is replaced by a space
- These characters are replaced by a space in the order shown:

  ` ! @ # $ % ^ & * + = / \ ( ) [ ] { } | : ; “ < > ?

In almost all of the programs, multiple spaces are replaced by one space during the filtering process.

# Appendix C: Programming Examples

Below are two snippets of code in two different program languages, which show how to connect to the server using a valid URL.

## C++ .Net 2008

```
// This is a snippet of code from a small test application.

// Prepare the webpage
 HttpWebRequest request =
(HttpWebRequest)WebRequest.Create("http://aviationknowledge.colorado.edu/cgi-
bin/nph-
findfrequency.cgi?Web=1&Space=ExpertPilot_v3&Words=Navigation%0D%0AWaypoint")
;
// execute the request
HttpWebResponse response = (HttpWebResponse)
    request.GetResponse();

Stream responseStream = response.GetResponseStream();

MemoryStream memStream = new MemoryStream();
byte[] buffer = new byte[2048];
int bytesRead = 0;

do
{
    bytesRead = responseStream.Read(buffer, 0, buffer.Length);
    memStream.Write(buffer, 0, bytesRead);

} while (bytesRead != 0);

responseStream.Close();

buffer = memStream.ToArray();

string html = System.Text.Encoding.ASCII.GetString(buffer);

// Do something with returned HTML
```

## PERL

```
#!/usr/bin/perl -w

use strict;
use LWP::Simple;
use URI::Escape;

my $site = 'http://aviationknowledge.colorado.edu/cgi-bin/';
my $script = 'nph-findfrequency.cgi';
my $web = '0'; # 0=no HTML output, 1=HTML ouput
my $space = 'ExpertPilot_v3';

#Using a set of words, create string with \r\n separating words.  Encode
string for the URL
my @words= ("Navigation" ,"Waypoint");
my $unencoded_words = $words[0] . "\r\n". $words[1];
my $filtered_words = filter_symbols($unencoded_words);
my $encoded_words = uri_escape($filtered_words);

#create URL and send for processing
```

```
my $url = $site . $script . "?Web=" . $web . "&Space=" . $space . "&Words=" .
$encoded_words;
my $content = get $url; die "Couldn't get $url" unless defined $content;

#All the information returned from the website is returned in $content
print $content;

sub filter_symbols
{
      my ($orig_text) = @_;

      $orig_text =~ tr/A-Z/a-z/; # Make everything lower-case

      $orig_text =~ s/([0-9])(,)([0-9])/$1$3/; #removes comma from numbers
i.e 40,000 -> 40000
      $orig_text =~ s/\./ /g;  #subs all periods with space
      $orig_text =~ s/\,/ /g;  #subs all commas with space

      $orig_text =~ s/`/ /g;
      $orig_text =~ s/!/ /g;
      $orig_text =~ s/@/ /g;
      $orig_text =~ s/#/ /g;
      $orig_text =~ s/\$/ /g;
      $orig_text =~ s/%/ /g;
      $orig_text =~ s/\^/ /g;
      $orig_text =~ s/&/ /g;
      $orig_text =~ s/\*/ /g;

      $orig_text =~ s/-//g;
      $orig_text =~ s/_//g;
      $orig_text =~ s/\+/ /g;
      $orig_text =~ s/=/ /g;

      $orig_text =~ s/\// /g;
      $orig_text =~ s/\\/ /g;

      $orig_text =~ s/\(/ /g;
      $orig_text =~ s/\)/ /g;
      $orig_text =~ s/\[/ /g;
      $orig_text =~ s/\]/ /g;
      $orig_text =~ s/\{/ /g;
      $orig_text =~ s/\}/ /g;
      $orig_text =~ s/\|/ /g;

      $orig_text =~ s/:/ /g;
      $orig_text =~ s/;/ /g;
      $orig_text =~ s/'s/ /g;
      $orig_text =~ s/'/ /g;
      $orig_text =~ s/"/ /g;
      $orig_text =~ s/</ /g;
      $orig_text =~ s/>/ /g;
      $orig_text =~ s/\?/ /g;


      return $orig_text;
}
```