LING/C SC/PSYC 438/538

Lecture 4

Sandiway Fong

Today's Lecture

- Text summarization
  - Open Text Summarizer (OTS)
  - macOS Summarize Service
  - ChatGPT
- Beginning programming with Perl
  - focusing on quoting (*very important*!)

App of the Day

- Text summarization service
  - available on Macs (*but buried deeply and turned off by default*)
  - Open Text Summarizer (ots) (on Linux), also web interface

AutoSummarize a document in Microsoft Office Word 2007

Learn how to AutoSummarize a document into key points or an outline using Microsoft Office Word 2007.

Video: AutoSummarize a document

The AutoSummary Tools in Office Word 2007 can highlight and assemble key points of a document. A student can use it to AutoSummarize a long list of science articles or even to quickly create an abstract for a finished history report.

Watch the video

Tweet   Print   Add t

Tips

Students are often asked to write expansi___ papers.

It is a good way to measure their research__ and writing skills. What is often left unmea__ ability to summarize and simplify.

no longer available in MS Office

Open Text Summarizer (ots)

(code: *read about how it works*)
- https://github.com/neopunisher/Open-Text-Summarizer/

- Web interface
  - http://www.splitbrain.org/services/ots

OPEN TEXT SUMMARIZER

This is a webinterface to the Open Text Summarizer tool. The tool automatically analyzes texts in various languages and tries to identify the most important parts of the text.

Just paste your text or load it from an URL to get it summarized.

Input

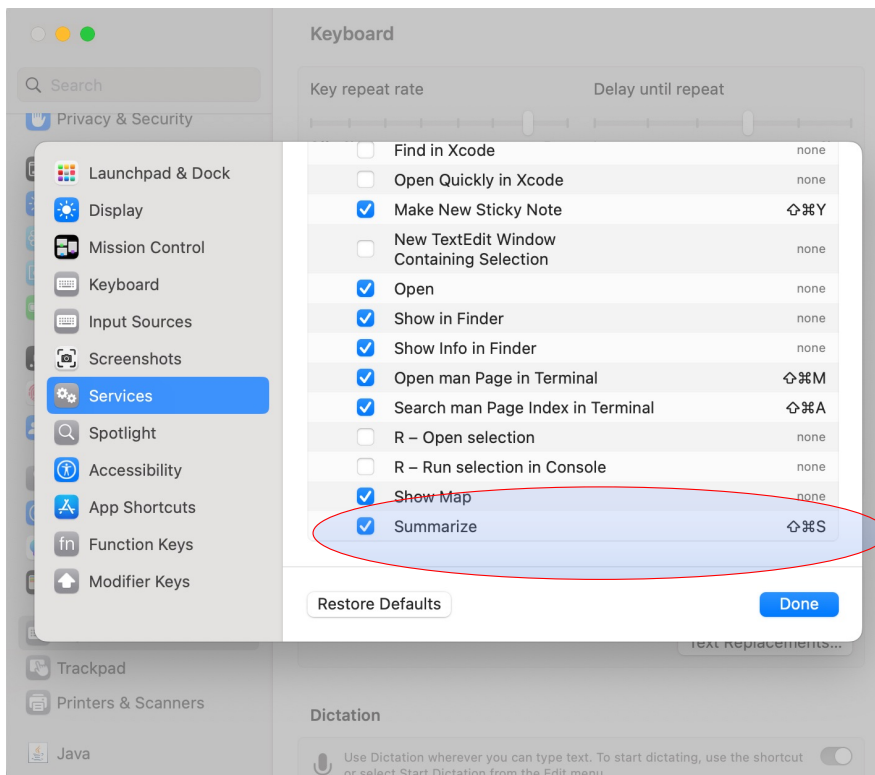_____(or load from URL)

Output
⦿Summary ○Keywords

Summarization Ratio
○5% ○10% ⦿20% ○30% ○40% ○50% ○60% ○70% ○80%

Language
en ⇕

Submit

App of the Day
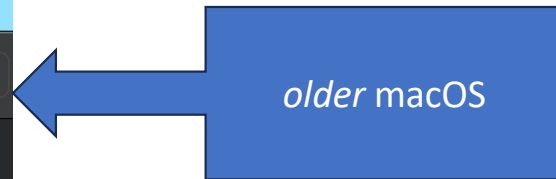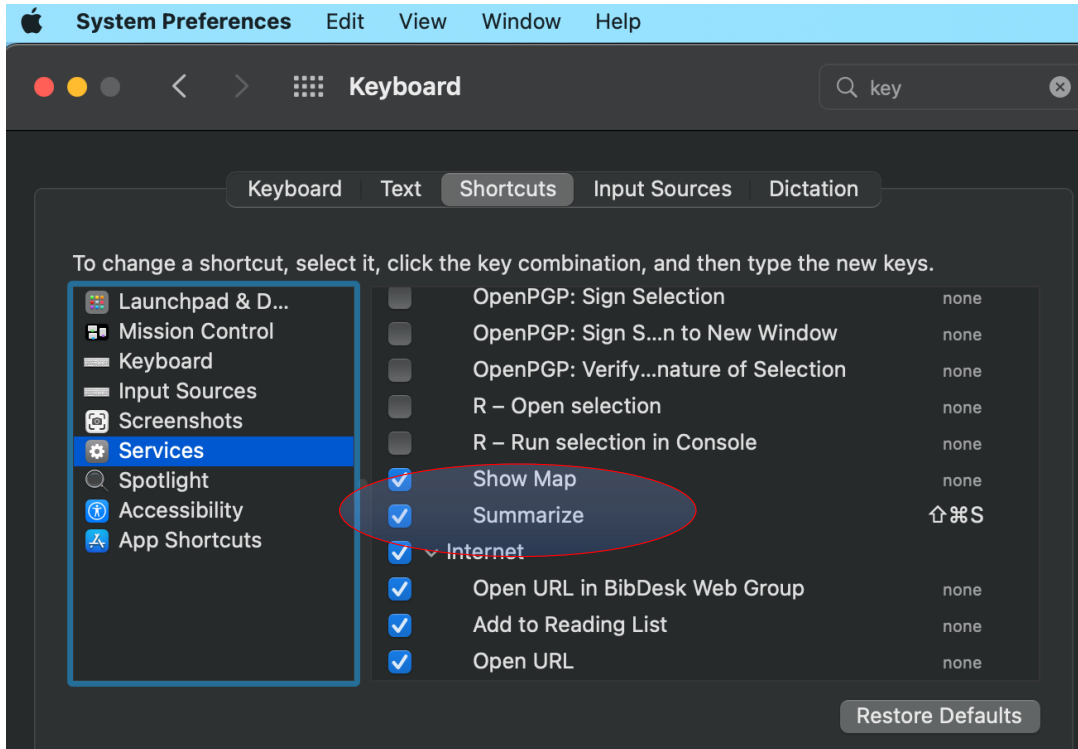
- System Preferences (macOS) > Keyboard > Shortcuts > Services > Summarize

App of the Day

- System Preferences (macOS) > Keyboard > Shortcuts > Services > Summarize

From lecture 2

- Obama: "*At a certain point, I've just concluded that for me personally it is important for me to go ahead and affirm that I think same-sex couples should be able to get married.*"

Executive Summarization

May 9, 2012 2:57pm

Obama: 'I Think Same-Sex Couples Should Be Able to Get Married'

f Like 60k   Tweet 845   +1 80   ✉ + 💬 178 Text +

Summary

Obama: "At a certain point, I've just concluded that for me personally it is important for me to go ahead and affirm that I think same-sex couples should be able to get married."

○ Sentences
○ Paragraphs   1   Summary Size   100%

Clear all

From lecture 2

https://abcnews.go.com/blogs/politics/2012/05/obama-comes-out-i-think-same-sex-couples-should-be-able-to-get-married

From lecture 2

OTS output

Obama: 'I Think Same-Sex Couples Should Be Able to Get Married'
By Rick Klein
May 9, 2012
President Obama today announced that he now supports same-sex marriage, reversing his longstanding opposition amid growing pressure from the Democratic base and even his own vice president. At the same time, the president has opposed efforts to ban gay marriage at the state level, saying that he did not favor attempts to strip rights away from gay and lesbian couples.
President Obama's Evolution on Same-Sex Marriage
The announcement completes a turnabout for the president, who has opposed gay marriage throughout his career in national politics.

From lecture 2

Default (GPT-3.5)

SA   Can you summarize "At a certain point, I've just concluded that for me personally it is important for me to go ahead and affirm that I think same-sex couples should be able to get married"?

Certainly! The statement can be summarized as follows: "I've reached a personal conclusion that it's important for me to express my belief in the right for same-sex couples to marry."

Perl

- Learn Perl
  - Books…
  - Online resources
    - http://learn.perl.org/, we begin with http://perldoc.perl.org/perlintro.html

Basic syntax overview

A Perl script or program consists of one or more statements. These statements are simply written in the script in a straightforward fashion. There is no need to have a `main()` function or anything of that kind.

Perl statements end in a semi-colon:

```
1.        print "Hello, world";
```

Quotes: single ('), double ("), backquote (`)

QWERTY
keyboard layout

- *problematic for programming*

- backquote or backtick (`) , orig. grave accent *è*, used in Bash shell for command substitution.

Quotes: single ('), double ("), backquote (`)

Sholes Glidden (1874)
original QWERTY keyboard
- *had no quotes at all*!

Unicode quotes:

Perl

- http://perldoc.perl.org/perlintro.html
- **One-liner:**
  - `perl –e 'PerlCode'`
  - three different runs:

```
[~$ perl –e 'print "Hello class!"'
[Hello class!~$
[~$ perl –e 'print "Hello class!\n"'
 Hello class!
[~$ perl –e 'print "Hello class!\n"
[> '
 Hello class!
 ~$
```

```
[~$ python3 –c 'print("hello")'
hello
~$
```

\n = newline character (if **interpolated**)

Quoting is tricky:
- Single quotes go to the Bash shell
- says don't touch anything inside, pass the contents to Perl
- double quotes says do escape characters +
        Bash shell variable interpolation

Perl

- http://perldoc.perl.org/perlintro.html
- **One-liner:**
  - perl –e "*PerlCode*" ← *notice double quotes here, single quotes in previous slide*
  - three different runs:

```
[~$ word='class'
[~$ perl –e "print 'Hello\n'"
[Hello\n~$
[~$ perl –e "print \"Hello\n\""
 Hello
[~$ perl –e "print \"Hello $word\n\""
Hello class
~$ ▮
```

```
[~$ word="class\n"
[~$ perl –e "print \"Hello $word\n\""
Hello class

~$ ▮
```

Quoting is tricky:

- double quotes says do Bash shell variable interpolation
- single quotes to Perl says don't interpolate (\n is not newline)
- Backslash \" needed for the Bash shell to not end the string, send " to Perl
- $word is a Bash shell variable, substitutes its value before sending it to Perl

Windows 10: command prompt

- Windows 10 command prompt
  - single quotes not recognized as quotes
  - double quotes can be used
- inside the (initial) double quotes, everything will be passed to Perl:
  - you can use single quotes
  - you can escape double quotes, i.e. \" to get a double quote.

```
Command Prompt
Microsoft Windows [Version 10.0.17134.285]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\sandiway>perl

C:\Users\sandiway>

C:\Users\sandiway>perl -e 'print 42'
Can't find string terminator "'" anywhere before EOF at -e line 1.

C:\Users\sandiway>perl -e "print 42"
42
C:\Users\sandiway>perl -le "print 42"
42

C:\Users\sandiway>perl -e "print \"42\n\""
42

C:\Users\sandiway>perl -e "print '42\n'"
42\n
C:\Users\sandiway>
```

Windows 10/11: PowerShell

- You can also use PowerShell in Windows 10/11:
    - both single and double quotes can be used as quoting characters in the shell.
    - Inside the initial quotes, you can escape any quotes that need to be passed to Perl.


```
Windows PowerShell
PS C:\Users\sandiway> perl -e "print 42"
42
PS C:\Users\sandiway> perl -e 'print 42'
42
PS C:\Users\sandiway> perl -e 'print \"Hello, world\"'
Hello, world
PS C:\Users\sandiway> perl -e 'print "Hello, world"'
No comma allowed after filehandle at -e line 1.
PS C:\Users\sandiway>
```

Programming Pitfalls

Can be maddening!


```
[~$ perl -e 'print "hello\n"'
hello
[~$ perl -e 'print "It's hello\n"'
[> '
[> "
Can't find string terminator '"' anywhere before EOF at -e line 1.
[~$ perl -e 'print "''hello\n"'
hello
[~$ perl -e 'print "hello\n"'
hello
[~$ perl -e 'print "' 'hello\n"'
Can't find string terminator '"' anywhere before EOF at -e line 1.
[~$ perl -e 'print "'"'"'hello\n"'
'hello
[~$ perl -e 'print "'"It's "'hello\n"'
It's hello
[~$ perl -e 'print "@ARGV\n"' 3 1 4 1 5
3 1 4 1 5
[~$ perl -e 'print @ARGV\n' 3 1 4 1 5
Backslash found where operator expected at -e line 1, near "@ARGV\"
        (Missing operator before \?)
syntax error at -e line 1, near "@ARGV\"
Execution of -e aborted due to compilation errors.
[~$ perl -e 'print @ARGV' 3 1 4 1 5
[31415~$ perl -e 'print "@ARGV\n"' 3 1 4 1 5
3 1 4 1 5
[~$ perl -e '@s = sort @ARGV; print "@s\n"' 3 1 4 1 5
1 1 3 4 5
[~$ perl -e '@s = sort @ARGV; print "@s\n"' 10 2 9 20 100
10 100 2 20 9
[~$ perl -e '@s = sort {$a <=> $b} @ARGV; print "@s\n"' 10 2 9 20 100
2 9 10 20 100
~$ ▌
```

18

Perl

- **One-liner:**
  - `perl –e 'PerlCode'`
- **Notes from the tutorial:**

```
2.   use strict;
3.   use warnings;
```

"Nanny mode": helps with debugging: makes you declare variables with `my`

- whitespace not always necessary, e.g.
  - `print"Hello class!\n";`

  is fine, but good idea to consistently use spacing (*not just for readability*)
- variable names must not begin with a number (*use a letter*), so
  - `$538students` is out
  - `$students538` is ok
- error messages are frequently uninformative (*and sometimes misleading*), e.g.

  `Bareword found where operator expected at example.prl line 3, near "$538students"`

  `(Missing operator before students?)`

  error not associated with the variable starting with a number

`print` as a statement vs. function

Perl

`world.perl`

```
1 print "Hello World";¶
2 print "Goodbye World\n"¶
```

Python3

`world.py`

```
1 print("Hello World",end='')¶
2 print("Goodbye World")¶
```

change my prompt →

```
● ● ●                    📁 ling538
(base) ling538-23$ export PS1='$ '
[$ perl world.perl
Hello WorldGoodbye World
$ ▮
```

```
● ● ●                    📁 ling538-
[$ python world.py
Hello WorldGoodbye World
$ ▮
```

- Program in a **plain text file**, don't have to worry about the Shell
- Same behavior, **but** different assumptions about end of line (\n)

  escape character

- Note also the differences in syntax:
  - (…) in Python3, no parentheses (*needed*) in Perl
  - **;** separate statements in Perl, no **;** in Python (*newline separator*)

`print` as a statement vs. function

- Well, in Python, you might need ; sometimes.

```
[~$ python3 -c 'print("hello")'
hello
[~$ python3 -c 'print("hello") print("goodbye")'
  File "<string>", line 1
    print("hello") print("goodbye")
                   ^
SyntaxError: invalid syntax
[~$ python3 -c 'print("hello");print("goodbye")'
hello
goodbye
~$
```

Perl vs. Python3

- **Scalars**: (basically things that take up one 32/64 bit word of memory)
  - variables begin with ($) - no such type naming requirement in Python
- Numbers (integer, floating point)
  - Python includes complex numbers   (`cmath` library)
    - user-definable precision floats   (`decimal` library)
- Strings   (double ".." or single quoted '..')
- References   (pointers to (non-)scalars)

Perl vs. Python3

Variable interpolation

- Perl:
  - `print "My name is $name\n";`
  - `print "My name is \$name\n";`
  - `print 'My name is \$name\n';`
- Python3:
  - `print("My name is", name)`
- Python2.7:
  - `print("My name is", name)`

```
>>> 538students = 'list'
  File "<stdin>", line 1
    538students = 'list'
        ^
SyntaxError: invalid syntax
>>>
```

Note: white spacing

Ungraded Exercises

- If you're new to programming, practice using Perl
  - **read** the intro and **run all** the examples!
  - if you're already a programmer, just learn the syntactic differences