

LING/C SC/PSYC 438/538

Lecture 10

Prof. Sandiway Fong

Today's Topics

- Lingua::EN:CMUDict
- Homework 7: *a fun one*
- *Perl regex*
 - *we're finally here!*

From last time

- Did anyone try the cmudict ungraded homework?

```
$ cpanm Lingua::EN::CMUDict
--> Working on Lingua::EN::CMUDict
Fetching http://www.cpan.org/authors/id/L/LM/LMETCALF/Lingua-
EN-CMUDict-0.07.tar.gz ... OK
Configuring Lingua-EN-CMUDict-0.07 ... OK
Building and testing Lingua-EN-CMUDict-0.07 ... OK
Successfully installed Lingua-EN-CMUDict-0.07
1 distribution installed
```

```
$ cat cmudict.perl
use Lingua::EN::CMUDict;
$o = new Lingua::EN::CMUDict;
print $ARGV[0], ": ", $o->number_of_syllables($ARGV[0]), "\n"
```

Perl warnings

- Seem to be enforced more strictly in recent versions:

```
$ perl cmudict.perl
```

```
Use of uninitialized value $word in pattern match (m//) at  
/Users/sandaway/perl5/lib/perl5/Lingua/EN/CMUDict.pm line 75.
```

```
Use of uninitialized value $word in uc at  
/Users/sandaway/perl5/lib/perl5/Lingua/EN/CMUDict.pm line 81.
```

Google AI suggested fixes

```
AI Mode All Videos Forums Images More ▾

Update the rhymes subroutine (around line 52) to add a guard clause immediately after pop :
```

```
perl

sub rhymes {
    my $self = shift;
    my $word = shift;
    return undef if (!defined $self->{words});

    $word = uc($word);
    return undef if (!defined $self->{words}->{$word});
    my @sylys = split(/ - /,$self->{words}->{$word});
    my $syl = pop(@sylys);

    # --- FIX: Stop early if no valid syllable suffix was found ---
    return wantarray ? () : undef if !defined $syl || $syl eq '';

    my @res;
    foreach my $el (keys %{$self->{words}}) {
        if ($self->{words}->{$el} =~ /$syl$/ && $el ne $word) {
            push @res,$el;
        }
    }
    return @res if wantarray;

    return $res[int(rand(scalar(@res)))];
}
```

```
AI Mode All Videos Forums Images More ▾

Update your number_of_syllables subroutine to look like this:
```

```
perl

sub number_of_syllables {
    my $self = shift;
    my $word = shift;

    # --- FIX 1: Return 0 if the word is undefined or empty ---
    return 0 if !defined $word || $word eq '';
    return undef if (!defined $self->{words});

    if ($word =~ /\s+/) {
        # --- FIX 2: grep out empty elements caused by trailing/\s
        my @words = grep { $_ ne '' } split(/\s+/, $word);
        my $sum = 0;
        foreach my $el (@words) { $sum += $self->number_of_syllab
        return $sum;
    }
    $word = uc($word);

    if (defined $self->{words}->{$word}) {
        return scalar(split(/ - /,$self->{words}->{$word}));
    }
    #check for plurality
    if ($word =~ /s$/i) {
        $word =~ s/s$/i;
        if (defined $self->{words}->{$word}) {
            return scalar(split(/ - /,$self->{words}->{$word}
        }
    }
    return undef;
}
```

Working .pm!

```
$ perl cmudict.perl trump T R AH M P
trump: 1
$ perl cmudict.perl hello HH AH L OW
hello: 2
$ perl cmudict.perl intersection IH N T ER S EH K SH AH N
intersection: 4
$ perl cmudict.perl asoifhaigfsfgdknj
asoifhaigfsfgdknj:
```

Google AI

3. The Code Was Always "Broken" Under the Hood

Even though Perl silently bypassed these issues in the past, the underlying logic was still failing:

- **In `rhymes`** : When `$syl` was uninitialized (`undef`), the regex became `/$syl$/` , evaluating to `//$` (match the end of any string). As a result, the `rhymes` function would return **every single word in the dictionary** instead of actual rhymes.
- **In `number_of_syllables`** : When trailing spaces caused a split into an empty string, `uc(undef)` returned `""` . The dictionary lookup `$self->{words}->{""}` failed, and it silently added `0` to the total syllable count.

The code "worked" only because Perl's permissive nature allowed it to fail gracefully without crashing. Modern Perl installations make sure you know exactly where your variables are slipping through empty.

<https://xkcd.com/208/>

Regular Expressions to the rescue



GroundNews

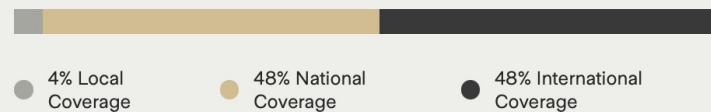
D News about Donald Trump

Follow

Stay current with all the latest and breaking news about **Donald Trump**, compare headlines and perspectives between news sources on stories happening today. In total, **29,647** stories have been published about **Donald Trump** which Ground News has aggregated in the past 3 months.

Location of Coverage About Donald Trump

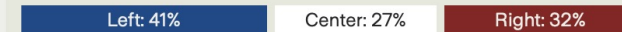
Based on **29,647** recent stories



Media Bias Breakdown

How is the media covering Donald Trump?

Donald Trump often is covered by Left and Right Media sources.



Covered Most by:



Do we expect even coverage?



US Foreign Policy · United States

In U.N. speech, Trump threatens to "annihilate" Iran and defends handling of...

40% Right coverage: 142 sources

[See the Story](#)



US Politics · United States

Trump, Greenland, Denmark sign deal in bid to end Arctic standoff

37% Center coverage: 44 sources

[See the Story](#)



Obamacare · United States

Vance-led task force set to remove 750K Obamacare enrollees, citing fraud

41% Right coverage: 34 sources

[See the Story](#)

GroundNews Blindspot

Blindspots



Blindspot: **Only 19% Left** 80 sources

Trump says Cuba 'will fall' at UN meeting, prompting Cuban delegation to walk...



Blindspot: **Only 3% Right** 40 sources

Trump to address UN General Assembly as Carney becomes voice of...

Blindspots

Blindspot: **0% Left** 11 sources

China's Rare-Earth Magnet Exports To US Plunge As Trump-Xi Meeting Looms

Left	0%
Center	33%
Right	67%

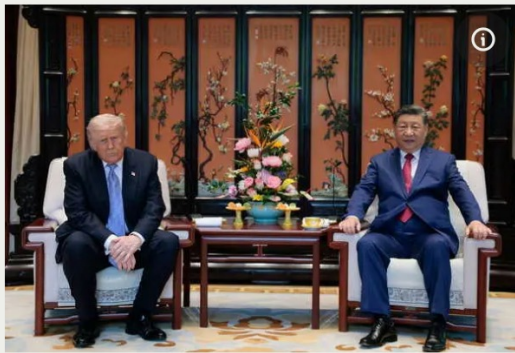
Blindspot: **0% Right** 7 sources

Trump Decries Democrats as 'Communists,' Even as He Looks to Impress...

Left	43%
Center	57%
Right	0%

September 23 to September 25, 2026

going on right now!



China Politics · China

China's Xi expected to press Trump to halt Taiwan arms sales under 1982 agreemen...

38% Right coverage: 29 sources

[See the Story.](#)



First Lady · Washington

First Lady Melania Trump Shares Details Ahead of China's Xi Jinping Visit

87% Right coverage: 15 sources

[See the Story.](#)



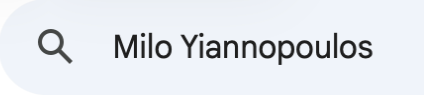
China Politics · China

China says it opposes 'illegal' US sanctions on Iranian airlines

50% Right coverage: 10 sources

[See the Story.](#)

Homework 7

- Here are some headlines about  
- 1. **Milo Yiannopoulos says he was ‘terrified’ by the ‘brutality’ of his ICE deportation**
- 2. Shed no tears for Milo Yiannopoulos, the deported right-wing provocateur
- 3. **I am a persecuted human rights icon**
- 4. **Milo Yiannopoulos, cheerleader of mass deportations, detained by ICE**
- 5. Deported Milo Yiannopoulos says he is ‘ashamed’ of support for Trump
- 6. **Milo Yiannopoulos deported to the U.K. one day after ICE arrest**
- 7. **Milo Yiannopoulos’ Black husband accuses him of shocking racism, “enslavement” & abuse**
- 8. **Milo Yiannopoulos ‘very ashamed’ of support for Trump after deportation**
- 9. ***Milo Yiannopoulos, Right-Wing Agitator, Deported Home to U.K.***
- 10. Far-right commentator Milo Yiannopoulos deported to UK
- 11. **British far-right provocateur Milo Yiannopoulos deported after ICE arrest**
- 12. **Far-right British provocateur Milo Yiannopoulos arrested by US immigration enforcement**

Homework 7

- Question 1
- **without looking ahead**, first use your own judgment and rate each headline (#1-12 *shown previously*) as being from media:
 - Left (L),
 - Center (C), or
 - Right (R)

Homework 7

 **cnn.com**

Milo Yiannopoulos says he was 'terrified' by the 'brutality' of his ICE deportation



20 days ago • By Jack Guy

 **Los Angeles Times**

Column: Shed no tears for Milo Yiannopoulos, the deported right-wing provocateur



9 days ago • By Robin Abcarian

 **The Spectator**

Milo Yiannopoulos: I am a persecuted human rights icon



8 days ago • By Milo Yiannopoulos

 **Courthouse News**

Milo Yiannopoulos, cheerleader of mass deportations, detained by ICE



Aug 28 • By Benjamin S. Weiss

 **The Guardian**

Deported Milo Yiannopoulos says he is 'ashamed' of support for Trump



20 days ago • By Anna Betts

 **NBC News**

Milo Yiannopoulos deported to the U.K. one day after ICE arrest



Aug 29 • By Shayna Jacobs

Homework 7

 LGBTQ Nation

Milo Yiannopoulos' Black husband accuses him of shocking racism, "enslavement" & abuse

4 days ago • By Daniel Villareal



 The Hill

Milo Yiannopoulos 'very ashamed' of support for Trump after deportation

21 days ago • By Ryan Mancini



 The New York Times

Milo Yiannopoulos, Right-Wing Agitator, Deported Home to U.K.

Aug 29 • By Hamed Aleaziz & Mack Liederman



 Reuters

Far-right commentator Milo Yiannopoulos deported to UK

Aug 29 • By Chris Kirkham



 abcnews.com

British far-right provocateur Milo Yiannopoulos deported after ICE arrest

Aug 29 • By Luke Barr



 AP News

Far-right British provocateur Milo Yiannopoulos arrested by US immigration enforcement

Aug 29 • By Safiyah Riddle



Homework 7

- Question 2
- GroundNews uses several sources for rating media sources
- Look up each of the news sources in AllSides (*see next slide*)
- Rate each headline source (#1-12 *shown previously*)
 - Left (L),
 - Center (C), or
 - Right (R)

AllSides

- www.allsides.com/media-bias/ratings

AllSides		
MEDIA BIAS & RATINGS		
Search all ratings FEATURED ALL SOURCES		
News Source	Type	Bias Rating
ABC News (Online)	News Media	
AlterNet	News Media	
Associated Press	News Media	
Associated Press Fact Check	News Media	
Axios	News Media	
BBC News	News Media	
Blaze Media	News Media	
Bloomberg	News Media	
Breitbart News	News Media	

Business Insider	News Media	
BuzzFeed News	News Media	
CBN	News Media	
CBS News (Online)	News Media	
Christian Science Monitor	News Media	
CNBC	News Media	
CNN Digital	News Media	
Daily Beast	News Media	
Daily Mail	News Media	
Democracy Now!	News Media	
Forbes	News Media	
Forgotten Fact Checks (National Review)	News Media	

Homework 7

- Question 3
- Compare your rating and the source? Accurate? Correlated?
- Which keywords and phrases do you think can be used to identify story as being Left (L), Center (C) or Right (R)? Or perhaps none?

Homework 7

- Email to me: sandiway@arizona.edu
- Subject: 438/538 Homework 7: ***YOUR NAME***
- Submit everything in one PDF file
- Submit a real file, not a link
- **Screenshots:** make sure text is not too tiny!
- Due **Sunday** midnight:
 - to be reviewed in class next **Tuesday**

Perl regex

- Python re
 - `import re`
 - slightly complicated string handling: use raw `r'...'` format
 - <https://docs.python.org/3/library/re.html>
 - *(there's also a 3rd party regex module)*

re — Regular expression operations

Source code: [Lib/re.py](#)

This module provides regular expression matching operations similar to those found in Perl.

Both patterns and strings to be searched can be Unicode strings (`str`) as well as 8-bit strings (`bytes`). However, Unicode strings and 8-bit strings cannot be mixed: that is, you cannot match a Unicode string with a byte pattern or vice-versa; similarly, when asking for a substitution, the replacement string must be of the same type as both the pattern and the search string.

Regular expressions use the backslash character (`'\'`) to indicate special forms or to allow special characters to be used without invoking their special meaning. This collides with Python's usage of the same character for the same purpose in string literals; for example, to match a literal backslash, one might have to write `'\\\\'` as the pattern string, because the regular expression must be `\\`, and each backslash must be expressed as `\\` inside a regular Python string literal.

The solution is to use Python's raw string notation for regular expression patterns; backslashes are not handled in any special way in a string literal prefixed with `'r'`. So `r"\n"` is a two-character string containing `'\'` and `'n'`, while `"\n"` is a one-character string containing a newline. Usually patterns will be expressed in Python code using this raw string notation.

Perl regex

- <https://www.pcre.org>

PCRE - Perl Compatible Regular Expressions

The PCRE library is a set of functions that implement regular expression pattern matching using the same syntax and semantics as Perl 5. PCRE has its own native API, as well as a set of wrapper functions that correspond to the POSIX regular expression API. The PCRE library is free, even for building proprietary software.

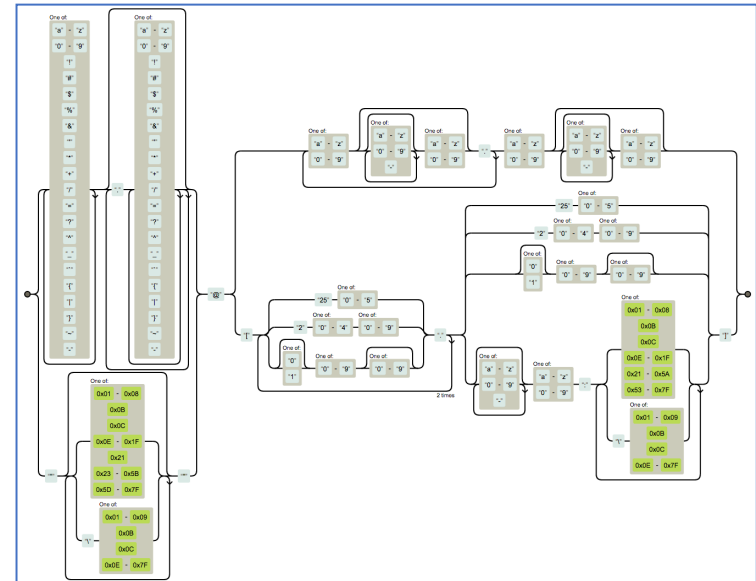
PCRE was originally written for the [Exim MTA](#), but is now used by many high-profile open source projects, including [Apache](#), [PHP](#), [KDE](#), [Postfix](#), and [Nmap](#). PCRE has also found its way into some well known commercial products, like [Apple Safari](#). Some other interesting projects using PCRE include [Chicken](#), [Onyx](#), [Hypermail](#), [Leafnode](#), [Askemos](#), [Wenlin](#), and [8th](#).

A regex from Hell

Email validation: RFC 5322 (Internet Message Format):

[illegible]

<https://www.rfc-editor.org/rfc/rfc5322>



Online regex tester

<https://regex101.com>

regular expressions 101

@regex101 donate contact bug reports & feedback wiki

REGULAR EXPRESSION

1 match, 14 steps (~4ms)

`/ house(cat(s)) /`

TEST STRING

housecats

SWITCH TO UNIT TESTS

EXPLANATION

▼ / house(cat(s)) /

house matches the characters **house** literally (case sensitive)

▼ 1st Capturing Group (cat(s))

▼ 1st Alternative cat(s)

cat matches the characters **cat** literally (case sensitive)

▼ 2nd Capturing Group (s)

▼ 1st Alternative s

s matches the character **s** literally (case sensitive)

MATCH INFORMATION

Match 1

Full match	0-9 `housecats`
Group 1.	5-9 `cats`
Group 2.	8-9 `s`

regex

- Read JM textbook chapter 2: section 1 on regex (*if you have it*)

Imagine that you have become a passionate fan of woodchucks and have recently learned that *groundhog* and *woodchuck* are different names for the same animal. Since you are writing a term paper on woodchucks, you now need to search through your paper for every occurrence of the term *woodchuck* and replace *woodchucks* with *woodchucks (groundhogs)*. But you also need to replace singular *woodchuck* with *woodchuck (groundhog)*. Instead of having to do this search twice, you would prefer to perform a single command for something like *woodchuck with an optional final s*. Or perhaps you might want to search for all the prices in some document; you might want to see all strings that look like \$199 or \$25 or \$24.99 in order to automatically extract a table of prices. In this chapter we introduce the **regular expression**, the standard notation for characterizing text sequences. The regular expression is used for specifying text strings in all sorts of text processing and information extraction applications.

Perl regex

- Read up on the syntax of Perl regular expressions
- Online tutorials
 - <http://perldoc.perl.org/perlrequick.html>
 - <http://perldoc.perl.org/perlretut.html>

Perl regex

- Perl regex matching:

- `$s =~ /foo/` (`/.../` contains a regex)
- can use in a conditional:
 - e.g. `if ($s =~ /foo/) ...`
 - evaluates to true/false depending on what's in `$s`
- can also use as a statement:
 - e.g. `$s =~ /foo/;`
 - global variable `$&` contains the match

- **Examples:**

```
[~$ perl -le 'print $ARGV[0] =~ /[Tt]he/' The man
1
[~$ perl -le 'print $ARGV[0] =~ /[Tt]he/' the man
1
[~$ perl -le 'print $ARGV[0] =~ /[Tt]he/' A man

~$ █
```

Perl regex

- Match is not exact:

```
[~$ perl -le 'print $ARGV[0] =~ /[Tt]he/' Theresa ate the cookies  
1  
~$ █
```

Perl regex

- Perl regex match and substitute:
 - `$s =~ s/foo/bar/`
 - `s/...match... /...substitute... /` contains two expressions
 - will **modify** `$s` by looking for a **single** occurrence of *match* and replacing that with *substitute*
 - `s/...match... /...substitute... /g` global substitution
- **Examples:**

```
~$ perl -le '$s = qq/@ARGV/; $s =~ s/[Tt]he/A/g; print $s' The man ate the cookie
A man ate A cookie
[~$ perl -le '$s = qq/@ARGV/; $s =~ s/[Tt]he/[Aa]/g; print $s' The man ate the cookie
[Aa] man ate [Aa] cookie
~$ █
```

`qq/STRING/`

`"STRING"`

A double-quoted, interpolated string.

<https://perldoc.perl.org/perlop#Quote-Like-Operators>

Perl regex

- Most useful with `template.perl` for reading in a file line-by-line:

```
open($fh, $ARGV[0]) or die "$ARGV[0] not found!\n";  
while ($line = <$fh>) {  
    $line =~ /.../  
}  
close($fh)
```
- Or on the command line in abbreviated form as:

```
perl -le 'open $f, filename; while (<$f>) {while (/regex/g)  
{print $&}}'
```
- Let's practice this after we've introduced the notation ...

Chapter 2: JM

spaces matter!

RE	Example Patterns Matched
/woodchucks/	“interesting links to <u>woodchucks</u> and lemurs”
/a/	“Ma <u>r</u> y Ann stopped by Mona’s”
/Claire_says,/	“ “Dagmar, my gift please,” <u>Claire</u> says,”
/DOROTHY/	“SURRENDER <u>DOROTHY</u> ”
/!/	“You’ve left the burglar behind again <u>!</u> ” said Nori

RE	Match	Example Patterns
/[wW]oodchuck/	Woodchuck or woodchuck	“ <u>Woodchuck</u> ”
/[abc]/	‘a’, ‘b’, or ‘c’	“In uo <u>m</u> ini, in soldat <u>i</u> ”
/[1234567890]/	any digit	“plenty of <u>7</u> to 5”

Figure 2.1 The use of the brackets [] to specify a disjunction of characters.

character class: Perl lingo

Chapter 2: JM

range: in ASCII table

RE	Match	Example Patterns Matched
/[A-Z]/	an upper case letter	"we should call it ' <u>D</u> renched Blossoms' "
/[a-z]/	a lower case letter	" <u>m</u> y beans were impatient to be hoed!"
/[0-9]/	a single digit	"Chapter <u>1</u> : Down the Rabbit Hole"

Figure 2.2 The use of the brackets [] plus the dash - to specify a range.

backslash lowercase letter for class
Uppercase variant for **all but** class

RE	Expansion	Match	Examples
\d	[0-9]	any digit	Party_of_5
\D	[^0-9]	any non-digit	Blue_moon
\w	[a-zA-Z0-9_]	any alphanumeric/underscore	Daiyu
\W	[^\w]	a non-alphanumeric	!!!!
\s	[_\r\t\n\f]	whitespace (space, tab)	
\S	[^\s]	Non-whitespace	in_Concord

Figure 2.6 Aliases for common sets of characters.

Chapter 2: JM

RE	Match (single characters)	Example Patterns Matched
[^A-Z]	not an upper case letter	"Oy <u>f</u> n pripetchik"
[^Ss]	neither 'S' nor 's'	" <u>I</u> have no exquisite reason for't"
[^\\.]	not a period	" <u>o</u> ur resident Djinn"
[e^]	either 'e' or '^'	"look up <u>^</u> now"
a^b	the pattern 'a^b'	"look up <u>a^b</u> now"

Figure 2.3 Uses of the caret ^ for negation or just to mean ^. We discuss below the need to escape the period by a backslash.

RE	Match	Example Patterns
/beg.n/	any character between <i>beg</i> and <i>n</i>	<u>begin</u> , <u>beg'n</u> , <u>begun</u>

Figure 2.5 The use of the period . to specify any character.

Chapter 2: JM

RE	Match	Example Patterns Matched
woodchucks?	woodchuck or woodchucks	<u>“woodchuck”</u>
colou?r	color or colour	<u>“colour”</u>

Figure 2.4 The question mark ? marks optionality of the previous expression.

RE	Match
*	zero or more occurrences of the previous char or expression
+	one or more occurrences of the previous char or expression
?	exactly zero or one occurrence of the previous char or expression
{n}	<i>n</i> occurrences of the previous char or expression
{n,m}	from <i>n</i> to <i>m</i> occurrences of the previous char or expression
{n, }	at least <i>n</i> occurrences of the previous char or expression

Figure 2.7 Regular expression operators for counting.

Can use parentheses (...) to group around a sub-expression if > 1 char
e.g. (ab)* vs. ab*

Perl regex

The regular expression engine provided by the PERL programming language is a powerful tool for defining and locating patterns in unstructured text. Unlike index-based approaches, this strategy does not impose a specific tokenization and thereby a predefined view of the basic entities contained in the corpus. As a consequence, it is possible to formulate patterns based on parts of words and patterns containing optional elements. For example, the expression `\S+ing\b` can be used to retrieve all words ending in *-ing*, or the pattern `\bmusick?\b` can be used to retrieve the spelling variants *music* and *musick*. The bridge version is searched character-by-character. In the search patterns, alpha-numeric characters are interpreted literally, except if they are preceded by a backslash character as in `\b`, which stands for a word-boundary, or `\s`, which stands for any character appearing on screen. Non-alphanumeric characters often have a non-literal interpretation, for example `?`, which, in the pattern `\bmusick?\b`, specifies that the character to its left may be present or

Example: `\S+ing\b`

- `\s` is a whitespace, so `\S` is a non-whitespace
- `+` is repetition (1 or more)
- `\b` is a word boundary, (words are made up of `\w` characters)

`\w` is a character class that matches any single *word* character (letters, digits, Unicode marks, and connector punctuation (like the underscore)).