Hi,
I am SUMAIYAH

Email : sumaiyah@nu.edu.pk

Office : In front of CS Secretariat

Faculty offices

Marks Distribution

Mid-1	:	15
Mid-2	:	15
Assignment:	05	
Quizzes	:	05
Project	:	10
Final	:	50

This is AI4001

GCR : t37g47w

Speech and Language Processing

An Introduction to Natural Language Processing,
Computational Linguistics, and Speech Recognition

Third Edition draft

Daniel Jurafsky
Stanford University

James H. Martin
University of Colorado at Boulder

Language

Language

Language is an instrument through which we communicate. It can be symbolic, spoken, and in written form.

Language – is an exclusively human property.

Language Components

Symbols set ={ 0,1}
Words = sequence of symbols (form & meaning)
Vocabulary = sequence of symbols or set of words
Text = composed of sequence of words from the vocabulary
Language = a language is constructed from sets all possible text

A language family is a group of languages with a common origin.

Natural Language Processing

Natural Language

Natural language or ordinary language is any language that has evolved naturally in humans through use and repetition without conscious planning or premeditation.

Natural languages can take different forms, such as speech or signing.

They are distinguished from constructed and formal languages such as those used to program computers.

Artificial Language

Artificial languages are languages of a typically very limited size which emerge either in computer simulations between artificial agents, robot interactions or controlled psychological experiments with humans.

It is different from formal language.

Formal language - A formal language is a set of strings of symbols together with a set of rules that are specific to it.

Natural Language Vs Artificial Language

There are four major reasons why Natural Language (NL) is very much more difficult to process than an Artificial Language(AL)

- NL contains a great deal of ambiguity which is controlled in AL
- NL generally has more complex structure than is to be found in AL
- There appears no simple universal way of representing the meaning of sentences in NL
- Structure and meaning are necessarily interconnected in NL but not in AL

Automatic Summarization

Information Extraction

Language Identification

Machine translation

Named Entity Recognition

Speech Recognition

Text Classification

Word Sense Disambiguation

Syntax

Syntax of a particular language is its grammar

It refers to the structure of sentences and the rules governing how words are organized to form grammatically correct phrases and sentences.

Horse Bag Tomato Teeth (N N N N)

Semantics

Semantics is particularly about the meaning of words used.

It refers to the meaning of words, phrases, sentences, and how they convey information. It deals with the interpretation of language in terms of its underlying meaning.

Colorless green ideas sleep furiously.

Semantics Analysis Vs Syntactic Analysis

What’s an Orange?

I am wearing an Orange shirt.

I am eating an Orange.

Think about Google Search. Then Vs Now.

Name Entity recognition.

Which type of conversation we do in our daily life???

"The cake, it's delicious!"

"The new restaurant, I heard it's really good."

I saw the man on the hill with a telescope.

They watched the movie with excitement in the crowded theater.

Lexical Ambiguity

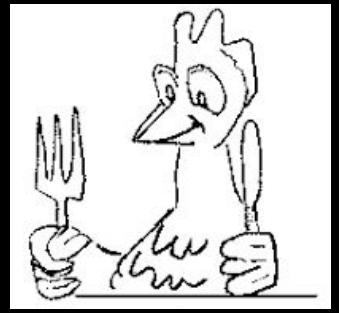
The presence of two or more possible meanings within a single word.

The banker decided to deposit the money in the bank.

She's reading a book about bats.

Syntactic Ambiguity

The presence of two or more possible meanings within a single sentence or sequence of words.



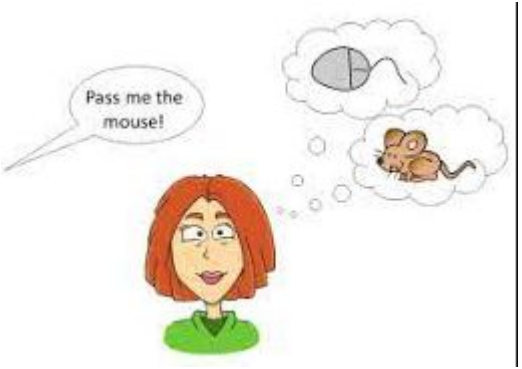
The chicken is ready to eat.

She left the university.

He's painting the house..

Class Activity

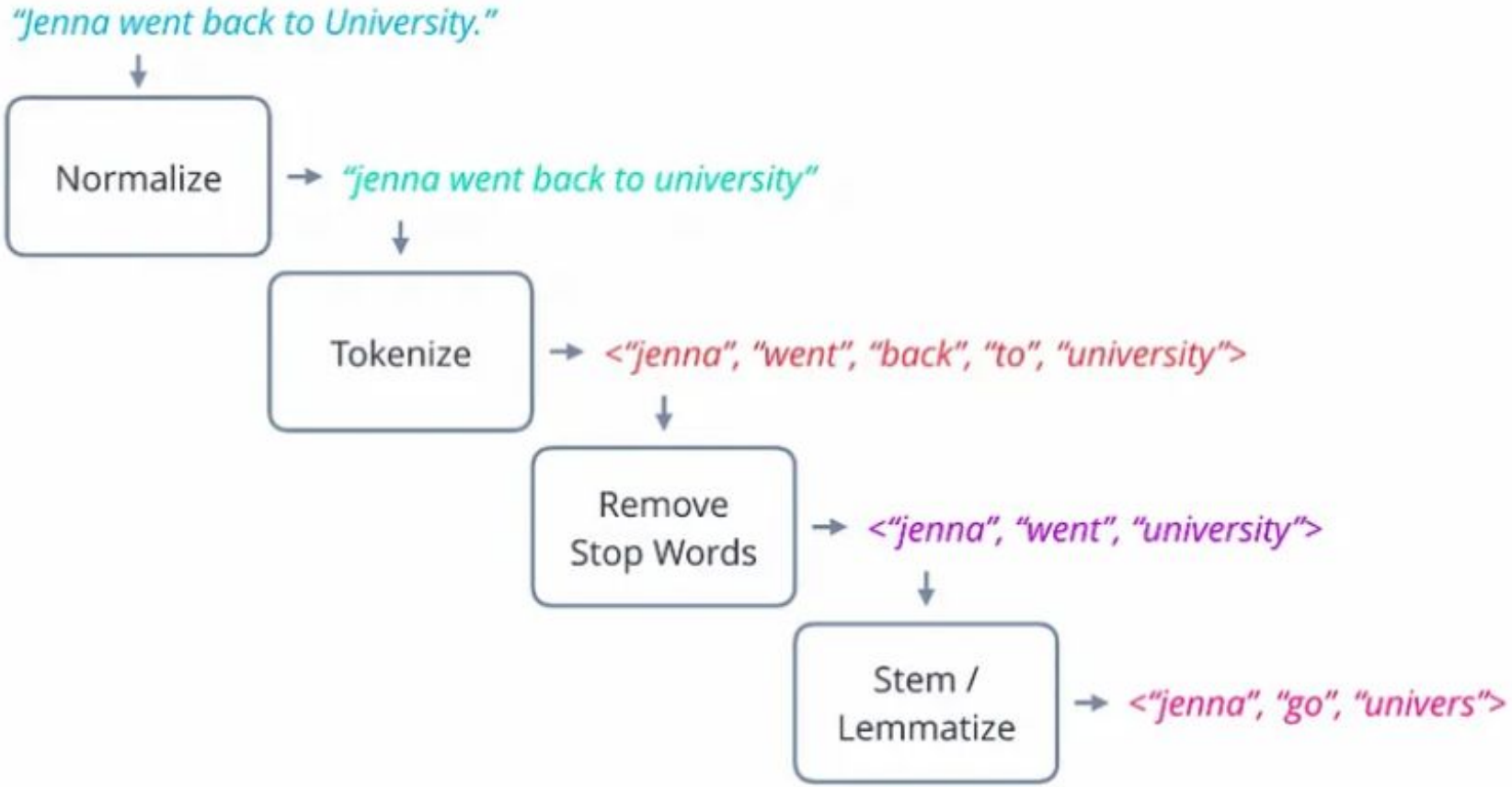
Write 3 sentence which shows lexical ambiguity and 3 sentence which shows syntactic ambiguity.



context=food

context=hardware

Did you say you were looking for **mixed nuts**?


"Jenna went back to University."
Normalize
"jenna went back to university"
Tokenize
<"jenna", "went", "back", "to", "university">
Remove Stop Words
<"jenna", "went", "university">
Stem / Lemmatize
<"jenna", "go", "univers">

Regular Expression

Regular Expressions

Useful for searching in texts, when we have a pattern to search for and a corpus of texts to search through.

The corpus can be a single document or a collection

Basic Regular Expressions

RE	Example Patterns Matched
/woodchucks/	"interesting links to woodchucks and lemurs"
/a/	"Mary Ann stopped by Mona's"
/!/	"You've left the burglar behind again!" said Nori

Figure 2.1 Some simple regex searches.

Basic Regular Expressions

RE	Match	Example Patterns
`/[wW]oodchuck/`	Woodchuck or woodchuck	"Woodchuck"
`/[abc]/`	'a', 'b', *or* 'c'	"In uomini, in soldati"
`/[1234567890]/`	any digit	"plenty of 7 to 5"

Figure 2.2 The use of the brackets `[]` to specify a disjunction of characters.

Basic Regular Expressions

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

Figure 2.3 The use of the brackets [] plus the dash - to specify a range.

Basic Regular Expressions

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

Figure 2.4 The caret ^ for negation or just to mean ^. See below re: the backslash for escaping the period.

Basic Regular Expressions

`?` as meaning “zero or one instances of the previous character”

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

Figure 2.5 The question mark ? marks optionality of the previous expression.

RE	Match	Example Matches
`/beg.n/`	any character between *beg* and *n*	begin, beg’n, begun

Figure 2.6 The use of the period `.` to specify any character.

Basic Regular Expressions

* commonly called Kleene * , generally pronounced “cleany star”

The Kleene star means “zero or more occurrences of the immediately previous character or regular expression”.

/a*/ means “any string of zero or more as”.

Matches a or aaaaaa or Off etc

/aa*/, meaning one a followed by zero or more as.

/[ab]*/ matches what?????

Basic Regular Expressions

Write a regex expression to find all these:

baa! Or baaa! Or baaaa! Or baaaaa!

Write a regex expression to find an integer?

Which strings will be match by this /fast.*fast/??

Activity : Explore `Kleen+`

Basic Regular Expressions - Anchors

Anchors are special characters that anchor regular expressions to particular places in a string.

`^` matches the start of a line.

`$` matches the end of a line.

`/^The dog\.$/` finds?????

Class Activity

Write regular expressions for the following languages.

1. the set of all alphabetic strings;
2. the set of all lower case alphabetic strings ending in a b;
3. the set of all strings from the alphabet a,b such that each a is immediately preceded by and immediately followed by a b;

Class Activity

Write a regular expression to match dates in the format MM/DD/YYYY.

Given a text containing hashtags (words starting with '#'), extract all the hashtags.

Basic Regular Expressions - Anchors

A “word” for the purposes of a regular expression is defined as any sequence of digits, underscores, or letters;

`\b` matches a word boundary, and `\B` matches a non-boundary.

`/\bthe\b/` matches `the` but not `other`.

`/\b99\b/` matches `99, $99` but not `299`

Class Activity

Given a text containing times in the format HH:MM AM/PM, write a regular expression to extract all times.

Example:

Text: "Meetings at 2:30 PM and 10:00 AM."

Class Activity

Given a text containing times in the format HH:MM AM/PM, write a regular expression to extract all times.

Example:

Text: "Meetings at 2:30 PM and 10:00 AM."

Solution:

Regular Expression: **(1[012]|0?[1-9]):[0-5][0-9] [APap][Mm]**

Class Activity

Give an RE that targets any string starting with an arbitrary number of \ followed by any number of *

Design an ERE that accepts phrases that fulfill the following criteria:

- The first word must start with a capital letter
- The phrase must end with a full stop .
- The phrase must be made of one or more words (made of the characters a...z and A...Z) separated by a single space

Class Activity

Give an RE that targets any string starting with an arbitrary number of \ followed by any number of * `\\*\**`

Design an ERE that accepts phrases that fulfill the following criteria:

- The first word must start with a capital letter
- The phrase must end with a full stop .
- The phrase must be made of one or more words (made of the characters a...z and A...Z) separated by a single space

```
[A-Z][A-Za-z]*(\ [A-Za-z]+)*\.$
```

Disjunction , Grouping and Precedence

/cat|dog/

/gupp(y|ies)/

/the*/ matches theeeee but not thethe.

Parenthesis	()
Counters	* + ? {}
Sequences and anchors	the ^my end$
Disjunction	|

/the|any/ matches the or any but not theny.

/[a-z]*/ this expression could match nothing, or just the first letter o, on, onc, or once. In these cases regular expressions always match the largest string they can; greedy we say that patterns are greedy, expanding to cover as much of a string as they can.

non-greedy matching, The operator *? And +? matches as little text as possible.

Regular Expressions

RE	Expansion	Match	First Matches
\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 2.7 Aliases for common sets of characters.

Regular Expressions

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}	*n* occurrences of the previous char or expression
{n,m}	from *n* to *m* occurrences of the previous char or expression
{n,}	at least *n* occurrences of the previous char or expression
{,m}	up to *m* occurrences of the previous char or expression

Figure 2.8 Regular expression operators for counting.

Regular Expressions

RE	Match	First Patterns Matched
*	an asterisk "*"	"K*A*P*L*A*N"
\.	a period "."	"Dr. Livingston, I presume"
\?	a question mark	"Why don't they come and lend a hand?"
\n	a newline	
\t	a tab	

Figure 2.9 Some characters that need to be backslashed.