AI INSTITUTE #AIISC
UNIVERSITY OF SOUTH CAROLINA

UNIVERSITY OF SOUTH CAROLINA

CSCE 590-1: Trusted AI

Lecture 13: Unstructured Text – Processing and Representation

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

30TH SEP, 2021

Carolinian Creed: "I will practice personal and academic integrity."

Organization of Lectures 13

- Introduction Segment
  - Review of Quiz 2
  - Review of students course projects – peer discussion

- Main Segment
  - Words
  - Parsing
  - Entity extraction
  - Text representation

- Concluding Segment
  - About next lecture – Lecture 14
  - Ask me anything

Introductory Segment

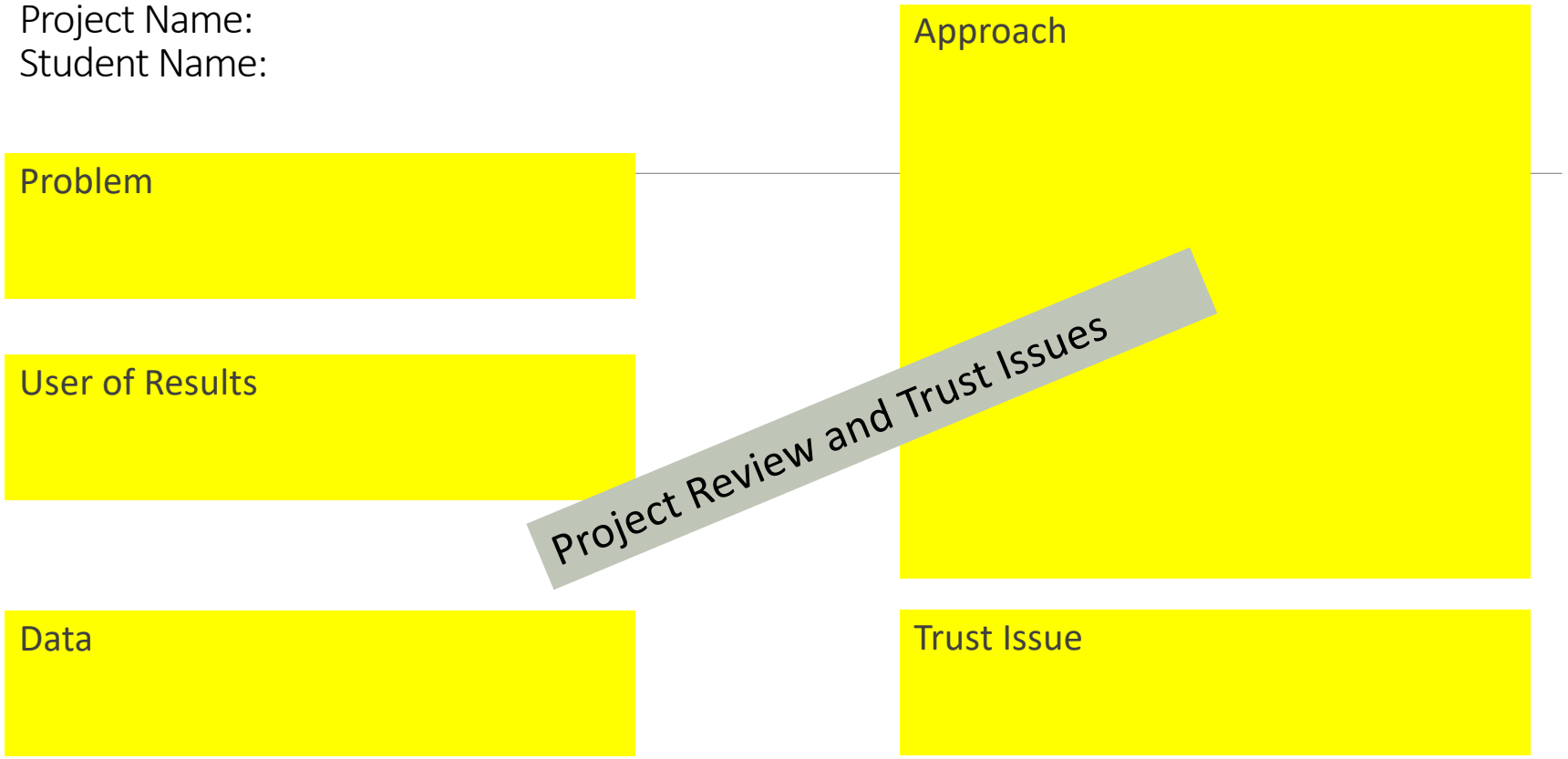
Schedule Snapshot

Sep 28 (Tu)	**Review: AI Fairness, Project presentations, Discussion**	Quiz 2
Sep 30 (Th)	AI - Unstructured (Text): Processing and Representation	
Oct 5 (Tu)	AI - Unstructured (Text): Common NLP Tasks	Mid-sem Project Review
Oct 7 (Th)	FALL BREAK	NO Classes, Course Midpoint
Oct 12 (Tu)	AI – Unstructured (Text): Analysis – Supervised ML	
Oct 14 (Th)	AI – Unstructured (Text): Analysis – Supervised ML	
Oct 19 (Tu)	**Invited Guest** – AI - Supervised ML: External Talk/ AI Explanation Methods (AIX)	10 am EST
Oct 21 (Th)	**Invited Guest** – AI - Supervised ML: External Talk/ Working Session on AIX360	10 am EST
Oct 26 (Tu)	AI - Unstructured (Text): Analysis – Supervised ML – Trust Issues	Quiz 3
Oct 28 (Th)	**Review: project presentations, Discussion**	
Nov 2 (Tu)	AI - Unstructured (Text): Analysis – Supervised ML – Trust Issues	
Nov 4 (Th)	AI - Unstructured (Text): Analysis – Supervised ML – Mitigation Methods	
Nov 9 (Tu)	AI - Unstructured (Text): Analysis – Supervised ML – Explanation Methods	
Nov 11 (Th)	Trust: Data Privacy Trust: AI Testing	

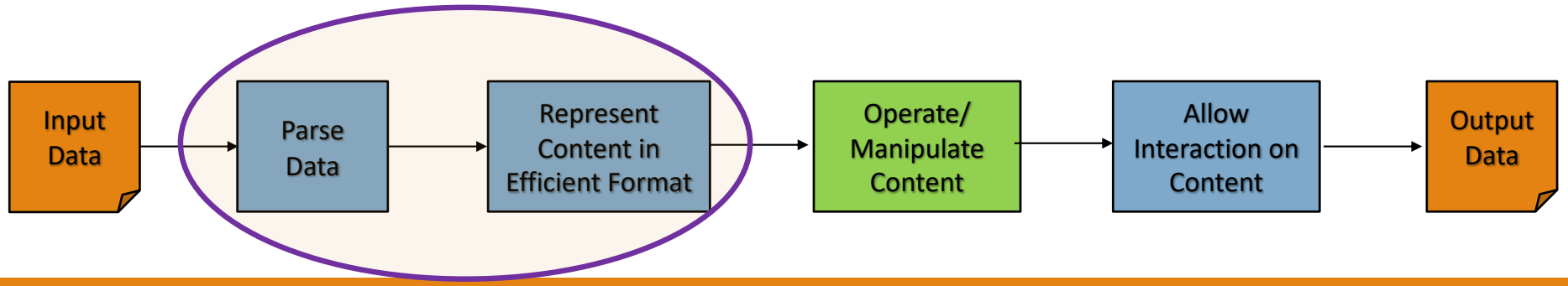
Quiz 2 Recap

- Individual part: Trust issues in projects
- Group part:
  - Water treatment data in Weka
  - Water Atlas data, multiple locations

Project Name:
Student Name:



Main Segment



How Text Complements Quantitative Analysis

- Quantitative data:
  - Captures precise information about well-defined attributes
  - Allows all the tools of mathematics for analysis
  - Example: average student scores in a course
- Textual data
  - Captures multi-dimensional information
  - Needs careful consideration about the context of information
    - Language can be imprecise
  - Example: topics covered in a course
- Both have complementary strengths and weaknesses

Understanding Concepts - Words

What is a Word ?

- Unix command - man wc

"**A word is defined as a string of characters delimited by white space characters**."

- Example
  - Content = "CSCE 771: Computer Processing of Natural Language
    Lecture 3: Words, Morphology, Lexicons
    Prof. Biplav Srivastava, AI Institute
    31st Aug 2020 "
  - Command = "wc -w content.txt"
    Result = "20 content.txt" (stored in file - result.txt)
  - "CSCE 771: Computer Processing of Natural Language (7)
    Lecture 3: Words, Morphology, Lexicons (12)
    Prof. Biplav Srivastava, AI Institute (17)
    31st Aug 2020 " (20)

Types of Words in English

Content words (open-class – i.e., continuously changing):

– **Nouns**: student, university, knowledge,...

– **Verbs**: write, learn, teach,...

– **Adjectives**: difficult, boring, hard, ...

– **Adverbs**: easily, repeatedly,...

Function words (closed-class – fixed):

– **Articles**: a, an, the

– **Prepositions**: in, with, under,...

– **Conjunctions**: and, or,...

– **Determiners**: a, the, every,...

Another Language - Turkish

A Turkish word

uygarlaştıramadıklarımızdanmışsınızcasına

uygar_laş_tır_ama_dık_lar_ımız_dan_mış_sınız_casına

"as if you are among those whom we were not able to civilize (=cause to become civilized)"

uygar: *civilized*
_laş: *become*
_tır: *cause somebody to do something*
_ama: *not able*
_dık: past participle
_lar: plural
_ımız: 1st person plural possessive (our)
_dan: *among* (ablative case)
_mış: past
_sınız: 2nd person plural (you)
_casına: *as if* (forms an adverb from a verb)

K. Oflazer pc to J&M

Chinese: 我开始写小说 = 我 开始 写 小说
I start(ed) writing novel(s)

A strict reliance on spaces will make us miss useful parts of text

Common Definitions

- **Corpus** (plural corpora): a computer-readable corpora collection of text or speech.
- **Lemma**: A lemma is a set of lexical forms having the same stem, the same major part-of-speech, and the same word sense. Example: Cat and cats have same lemma.
- **Word form**: The word form is the full inflected or derived form of the word. Example: Cat and cats have different word forms.
- **Word type**: Types are the number of distinct words in a corpus. If the set of words is V, the number of types is the word token vocabulary size |V|.
- **Word tokens**: The total number N of running words in the sentence / document of interest.
- **Code switching**: use multiple languages in a code switching single communicative act – Example: Hindlish (Hindi English), Spanish (Spanish English)

"They picnicked by the pool, then lay back on the grass and looked at the stars."

- 16 (word) tokens, 14 (word) types

Source: Jurafsky & Martin

Lexical Meaning – Common Terms

- **Synonym**: same/ similar meaning
  - start-begin, finish-end, far-distant
- **Antonym**: opposite meaning
  - Far – near, clever - stupid, high - low, big – small
- **Homonym**: identical in spelling and pronunciation
  - bear, bank, ...
- **Homophones**: sounds identical but are written differently
  - site-sight, piece-peace.
- **Homograph**: written identically but sound differently
  - Potato, tomato, lead, wind, minute
- **Polysemy**: a word or phrase which has two(or more) rated meanings
  - Duck, sharp

Source: Mausam

Knowing About Words

Of course he wants to take the advanced course too.
He already took two beginners' courses.

- Words – set of characters separated by spaces
- Word forms –
  - Spelling differences - specialize v/ specialise
  - Meaning similarity/differences - Take/ took, course/ courses, two/ too
- Word types – distinct words

Pop Quiz:
- Are word tokens and word types same in the example above?
- Which words have different forms in the example above?

Word Variety

- **Inflection:** creates different forms of the same word
  - Verbs: to be, being, I am, you are, he is, I was,
  - Nouns: one book, two books
- **Derivation:** creates different words from the same lemma
  - grace ⇒ disgrace ⇒ disgraceful ⇒ disgracefully
- **Compounding:** new words from combinations

  "ice cream", "website", "web site", "New York-based"
- **Clitics** - *a clitic is a morpheme that has syntactic characteristics of a word, but depends phonologically on another word or phrase. In this sense, it is syntactically independent but phonologically dependent …*

  English: "doesn't" , "I'm" ,
  Italian: "dirglielo" = dir + gli(e) + lo // tell + him + it

New words over time:
Google ⇒ Googler, to google, to ungoogle, to misgoogle, googlification, ungooglification, googlified, Google Maps, Google Maps service, ...

Morphology

Morphemes: The small meaningful units that make up words
Stems: The core meaning-bearing units
Affixes: Bits and pieces that adhere to stems

Morphemes: stems, affixes

dis-grace-ful-ly
prefix-stem-suffix-suffix

Many word forms consist of a stem plus a number of affixes (*prefixes* or *suffixes*)

Infixes are inserted inside the stem.

Circumfixes (German gesehen) surround the stem

Morphemes: the smallest (meaningful/grammatical) parts of words.

Stems (grace) are often free morphemes.

Free morphemes can occur by themselves as words.

Affixes (dis-, -ful, -ly) are usually bound morphemes.

Bound morphemes have to combine with others to form words.

- Plural nouns add -s to singular:
  - book-books,
- but:
  - box-boxes, fly-flies, child-children
- Past tense verbs add -ed to infinitive:
  walk-walked,
- but:
  - like-liked, leap-leapt

Source: Julia Hirschberg

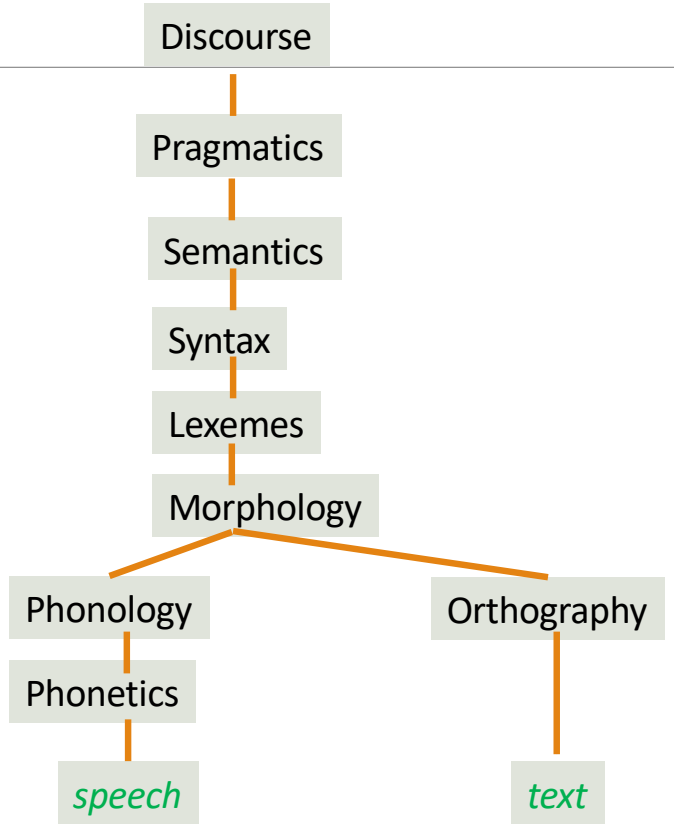
Morphological Generation

- Generate legal variations.
  - For **grace** (**stem**): graceful, gracefully, disgrace, disgraceful, disgracefully, ungraceful, ungracefully, undisgraceful, undisgracefully,…
- But avoid ungrammatical variations
  - *gracelyful, *gracefuly, *disungracefully,...

Source: Julia Hirschberg

Processing Language

Levels of Linguistic Studies



- **Discourse:** study of group of sentences
- **Pragmatics:** how context contributes to meaning of sentences
- **Semantics:** meaning of words and combinations of words
- **Syntax:** rules for combining and using words/ phonemes.
- **Lexemes:** a set of words that are related through inflection (fly: verb, fly: noun)
- **Morphology**—rules that govern morphemes - the minimal meaningful units of language (lemmas and affixes)
- **Orthography**: convention for writing a language. E.g., spelling
- **Phonology:** organization of speech sound (i.e., phoneme)
- **Phonetics**: study of how sound is made and received

Types of Parsing

- **Phrase structure** / **Constituency Parsing**: find phrases and their recursive structure.
  Constituency - groups of words behaving as single units, or constituents.
  - **Shallow Parsing/ Chunking**: identify the flat, non-overlapping segments of a sentence: noun phrases, verb phrases, adjective phrases, and prepositional phrases.
- **Dependency Parsing**: find relations in sentences
- **Probabilistic Parsing**: given a sentence X, predict the most probable parse tree Y

Advanced Topic –Language Formalism

An alphabet $\sum$ is a set of symbols:

e.g. $\sum = \{a, b, c\}$

A string ω is a sequence of symbols, e.g $\omega = abcb$.

The empty string ε consists of zero symbols.

The Kleene closure $\sum^*$ ('sigma star') is the (infinite) set of all strings that can be formed from $\sum$:

$\sum^* = \{\varepsilon, a, b, c, aa, ab, ba, aaa, \ldots\}$

A language $L \subseteq \sum^*$ over $\sum$ is also a set of strings.

Typically we only care about proper subsets of $\sum^*$ $(L \subset \Sigma)$.

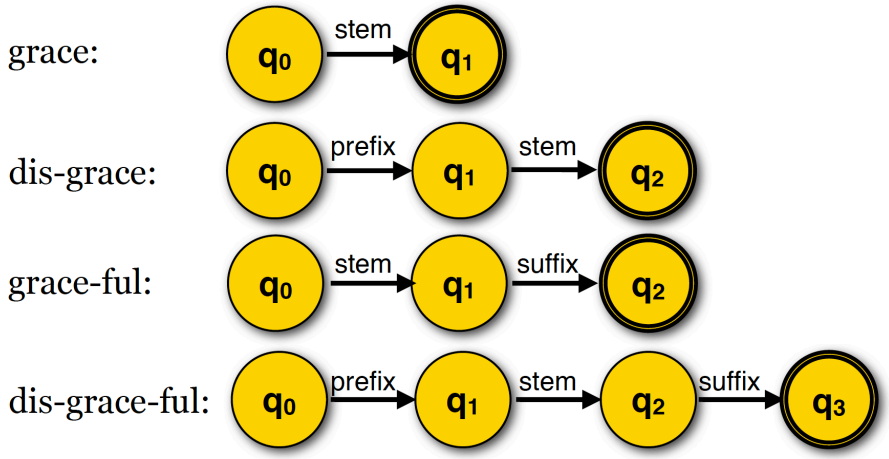
- Automata
- Finite State Automata
- Deterministic Finite State Automata(DFSA)
- Non-Deterministic Finite State Automata(NDFSA)

Source: Julia Hirschberg

Advanced Topics – Recognizing as Automata

- Automata –
  - an abstract model of a computer which reads an input string, and changes its internal state depending on the current input symbol. It can either accept or reject the input string.
  - Hence, an automata defines a language
- Finite State Automata – regular expressions

Source: Julia Hirschberg



Review: Regular Expression

Metacharacter	Explanation
^	Matches the starting position within the string
.	Matches any single character
[ ]	Matches a single character that is contained within the brackets
[^]	Matches a single character that is not contained within the brackets.
$	Matches the ending position of the string
*	Matches the preceding element zero or more times
+	Matches the preceding element one or more times
|	Separates choices

Regex	Matches any string that
hello	contains {hello}
gray|grey	contains {gray, grey}
gr(a|e)y	contains {gray, grey}
gr[ae]y	contains {gray, grey}
b[aeiou]bble	contains {babble, bebble, bibble, bobble, bubble}
[b-chm-pP]at|ot	contains {bat, cat, hat, mat, nat, oat, pat, Pat, ot}
colou?r	contains {color, colour}
rege(x(es)?|xps?)	contains {regex, regexes, regexp, regexps}
go*gle	contains {ggle, gogle, google, gooogle, goooogle, ...}
go+gle	contains {gogle, google, gooogle, goooogle, ...}
g(oog)+le	contains {google, googoogle, googoogoogle, googoogoogoogle, ...}
z{3}	contains {zzz}
z{3,6}	contains {zzz, zzzz, zzzzz, zzzzzz}
z{3,}	contains {zzz, zzzz, zzzzz, ...}

Example Source: https://cs.lmu.edu/~ray/notes/regex/

Implementation: Finding Words in Python

- Python has extended Regex specifications for convenience
- Useful for
  - Matching patterns
  - Information extraction
  - Content manipulation (e.g., substitution)
  - Error (e.g., spelling) correction

```
data = "The CSCE 771 course is taught at
        University this Fall!"
pattern = "[tT]+\w"
m = re.findall(pattern, data)
print(m)
```

['Th', 'ta', 'ty', 'th']

Details: https://docs.python.org/3/library/re.html

Code Examples

- Regular expressions
  - https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l20-text-overview/WordLesson-Examples.ipynb
- Supporting multiple languages – encoding
  - https://github.com/biplav-s/course-d2d-ai/blob/main/sample-code/l20-text-overview/Multiple%20Languages.ipynb

Word Representation

- Words as independent, discrete symbols
- Why
  - To process language efficiently (numeric representation)
  - Find similarity between words efficiently (vector representation)

Credit:
Contextual Word Representations: Putting Words into Computers", by Noah Smith, CACM June 2020

Contextual Word Embeddings Paper

- Words as independent, discrete symbols
- Words with distributional assumptions:
  - Context: given a word, its nearby words or sequences of words
  - Words used in similar ways are likely to have related meanings; i.e., words used in the same (similar) context have related meanings
    - No claim about meaning except relative similarity v/s dis-similarity of words
- Two main strategies
  - Compare with words in a manually-created taxonomy, e.g., Wordnet
  - Learn context and representation from data

Credit:
Contextual Word Representations: Putting Words into Computers", by Noah Smith, CACM June 2020

Concluding Segment

Lecture 13: Concluding Comments

- We completed review of projects for trust issues
- Looked at textual content
  - Words in English
  - Understood types of language processing
  - Explored representation

About Next Lecture – Lecture 14

Lecture 14: Common NLP Tasks

- Text representation
- A subset from common NLP Tasks
  - Text similarity
  - Event Extraction
  - Sentiment detection
  - Question Answering
  - Summarization
  - Machine translation
  - Natural Language Interface to Databases
  - Natural Language Generation