CSCE 771: Computer Processing of Natural Language
Lecture 5: Representation (Paper), Parsing, Projects

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

3RD SEPTEMBER, 2024

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

Acknowledgement: Used materials by Profs. Mausam, Jurafsky & Martin, Robert C. Berwick, Graham Neubig

Organization of Lecture 5

- Opening Segment
  - Announcements
- Main Lecture
- Concluding Segment
  - Course Project – review topics
  - About Next Lecture – Lecture 6

Main Section
- Paper discussion – Word Representation
- Parsing - introduction

Recap of Lecture 4

- We looked at a variety of NLP basic tasks
  - Tokenization – getting tokens for processing
  - Normalization - making into canonical form
  - Case folding – handling cases
  - Lemmatization – handling variants (shallow)
  - Stemming – handling variants (deep)

- NLP for business – sentiments for market intelligence

See and try out tools added on Github reading page:
https://github.com/biplav-s/course-nl-f24/blob/main/reading-list/Readme-AI-NLP.md

Review of Resume Exercise

Main Lecture

Paper Discussion

Contextual Word Representations: Putting Words into Computers",

by Noah Smith, CACM June 2020

https://cacm.acm.org/research/contextual-word-representations/

Problem

- How to represent words ?
- How to measure similarity, e.g., between words, and texts?
- How to determine different contexts (senses) in which words are used?
- How to handle noise, typos?

S1 - This is an apple
S2 - These are apples

S3 - This is an apples
S4 - There are apply

Option 1 - Characters

- How to represent words?
  - Characters / Unicode / …
- How to measure similarity between words, and texts?
  - Edit distance***: actions to convert one string to another***
  - Hamming distance: ***difference considering substitution***
- How to determine different contexts (senses) in which words are used?
  - Neighborhood of words: Bi-, tri-, N-gram representations

Distance between: Kitten, Sitting

Edit Distance

Algorithm	Operations Allowed			
	Insertions	Deletions	Substitutions	Transposition
Levenshtein Distance	✓	✓	✓	
Longest Common Subsequence (LCS)	✓	✓		
Hamming Distance			✓	
Damerau–Levenshtein Distance	✓	✓	✓	✓
Jaro distance				✓

Levenshtein distance:
1.**k**itten → **s**itten (substitute "s" for "k")
2.sitt**e**n → sitt**i**n (substitute "i" for "e")
3.sittin → sittin**g** (insert "g" at the end)

LCS distance (insertions and deletions only):
1.**k**itten → itten (delete "k" at 0)
2.itten → **s**itten (insert "s" at 0)
3.sitt**e**n → sittn (delete "e" at 4)
4.sittn → sitt**i**n (insert "i" at 4)
5.sittin → sittin**g** (insert "g" at 6)

Source: https://en.wikipedia.org/wiki/Edit_distance

Option 2 - Vectors

- How to represent words? Vectors
  - But, what scheme in vectors
    - One-hot encoding
    - Arbitrary, principled, …

- How to measure similarity between words, and texts?
  - Cosine similarity

- How to determine different contexts in which words are used?
  - Neighborhood of words: Bi-, tri-, N-gram representations
  - Contextual word vectors

Cosine Similarity

$$\text{cosine similarity} = S_C(A,B) := \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\cdot\sqrt{\sum\limits_{i=1}^{n} B_i^2}},$$

Property: two proportional vectors have a cosine similarity of 1, two orthogonal vectors have a similarity of 0, and two opposite vectors have a similarity of -1.

Usually used for [0,1]

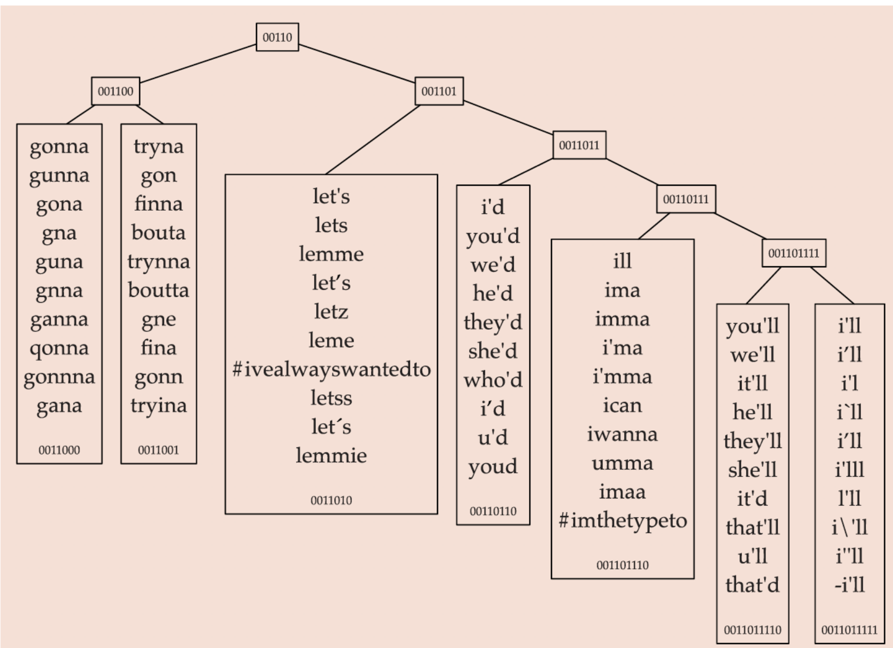
Sci-kit method python: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html

Source: https://en.wikipedia.org/wiki/Cosine_similarity

Contextual Word Embeddings

- Words as discrete
- 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

Contextual Representation by Clustering



Main steps

- Cluster words by context (i.e., neighborhood of the word)
- Compare with words in a manually-created taxonomy, e.g., Wordnet

The 10 most frequent words in clusters in the section of the hierarchy with prefix bit string 00110.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N.A. Improved part-ofspeech tagging for online conversational text with word clusters. In Proceedings of 2013 NAACL.

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

Contextual Representation by Dimensionality Reduction

- Creating word vectors in which each dimension corresponds to the frequency the word type occurred in some context (here, two words on either side of *astronomers, bodies, objects*)
- Strategy 1: select contexts
  - Examples
    - Words in the neighborhood
    - Words of specific types
  - Build vectors
  - Use vector operations to derive meaning

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

context words	**v**(astronomers)	**v**(bodies)	**v**(objects)
’t			1
,		2	1
.	1		1
1			1
And			1
Belt			1
But	1		
Given			1
Kuiper			1
So	1		
and		1	
are		2	1
between			1
beyond		1	
can			1
contains		1	
from	1		
hypothetical			1
ice		1	
including		1	
is	1		
larger		1	
now	1		
of	1		

$$\text{cosine\_similarity}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|}$$

	astronomers	bodies	objects
astronomers	$\frac{14}{\sqrt{14} \cdot \sqrt{14}} = 1$	$\frac{0}{\sqrt{24} \cdot \sqrt{14}} = 0$	$\frac{1+1}{\sqrt{14} \cdot \sqrt{16}} \approx 0.134$
bodies		$\frac{24}{\sqrt{24} \cdot \sqrt{24}} = 1$	$\frac{2+2+2}{\sqrt{24} \cdot \sqrt{16}} \approx 0.306$
objects			$\frac{16}{\sqrt{16} \cdot \sqrt{16}} = 1$

Bodies and **objects** are **most similar** (0.306) than
- **Bodies** and **astronomers** (0)
- **Objects** and **astronomers** (0.134)

Outside Paper – TF-IDF

TF-IDF based Word Representation -1

- Given N documents

- **Term frequency (TF):** for term (word) t in document d = tf(t, d)

Variants to reduce bias due to document length

Variants of term frequency (tf) weight

weighting scheme	tf weight
binary	$0, 1$
raw count	$f_{t,d}$
term frequency	$f_{t,d} \Big/ \sum_{t' \in d} f_{t',d}$
log normalization	$\log(1 + f_{t,d})$
double normalization 0.5	$0.5 + 0.5 \cdot \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$
double normalization K	$K + (1 - K) \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$

Sources:
(a) sci-kit documentation
(b) Wikipedia: https://en.wikipedia.org/wiki/Tf%E2%80%93idf

TF-IDF based Word Representation -2

- Given N documents
- Term frequency (TF): for term (word) t in document d
  = tf(t, d)

- **Inverse document frequency IDF**(t)

  = log [N / **DF**(t)] + 1

DF(t) = **document frequency**, the number of documents in the document set that contain the term t.

- **TF-IDF**(t, d) = TF(t, d) * IDF(t),

Variants of inverse document frequency (idf) weight

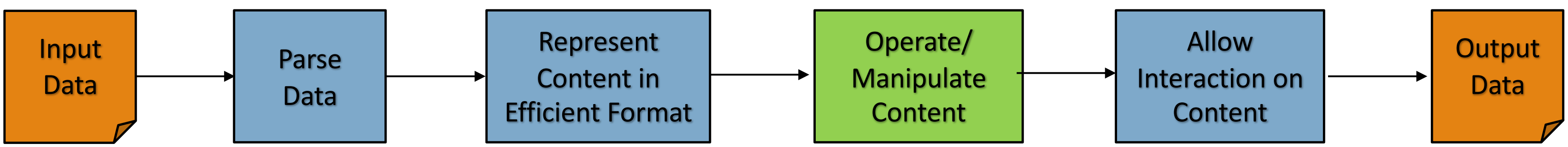
weighting scheme	idf weight ($n_t = \|\{d \in D : t \in d\}\|$)
unary	1
inverse document frequency	$\log \frac{N}{n_t} = -\log \frac{n_t}{N}$
inverse document frequency smooth	$\log\left(\frac{N}{1+n_t}\right) + 1$
inverse document frequency max	$\log\left(\frac{\max_{\{t' \in d\}} n_{t'}}{1+n_t}\right)$
probabilistic inverse document frequency	$\log \frac{N-n_t}{n_t}$

Sources:
(a) sci-kit documentation
(b) Wikipedia: https://en.wikipedia.org/wiki/Tf%E2%80%93idf

TF-IDF Example Calculation

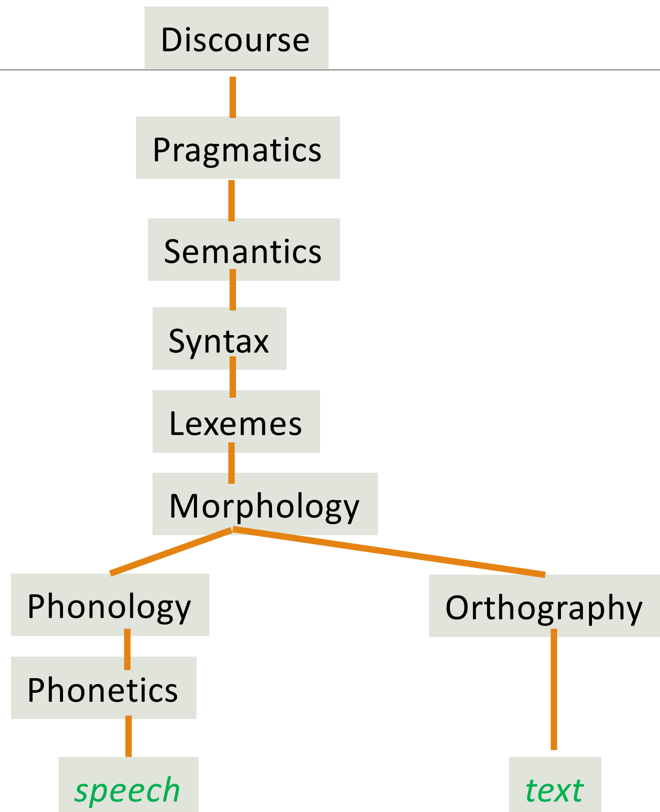
Github: https://github.com/biplav-s/course-nl-f22/blob/main/sample-code/l5-wordrepresent/Word%20Representations%20-%20Vectors.ipynb

Parsing



Input Data
Parse Data
Represent Content in Efficient Format
Operate/ Manipulate Content
Allow Interaction on Content
Output Data

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

Why Parsing

- Recognizing legal inputs from illegal
- Usage of parse representation - parse tree
  - Grammar checking
  - Semantic analysis
  - Machine translation
  - Question answering
  - Information extraction
  - Speech recognition
  - …

Adapted from material by Robert C. Berwick

Background: Context Free Grammar (CFG)

N	a set of **non-terminal symbols** (or **variables**)
Σ	a set of **terminal symbols** (disjoint from N)
R	a set of **rules** or productions, each of the form $A \rightarrow \beta$,
	where A is a non-terminal,
	β is a string of symbols from the infinite set of strings $(\Sigma \cup N)*$
S	a designated **start symbol** and a member of N

From Jurafsky & Martin

Simple Example Using CFGs

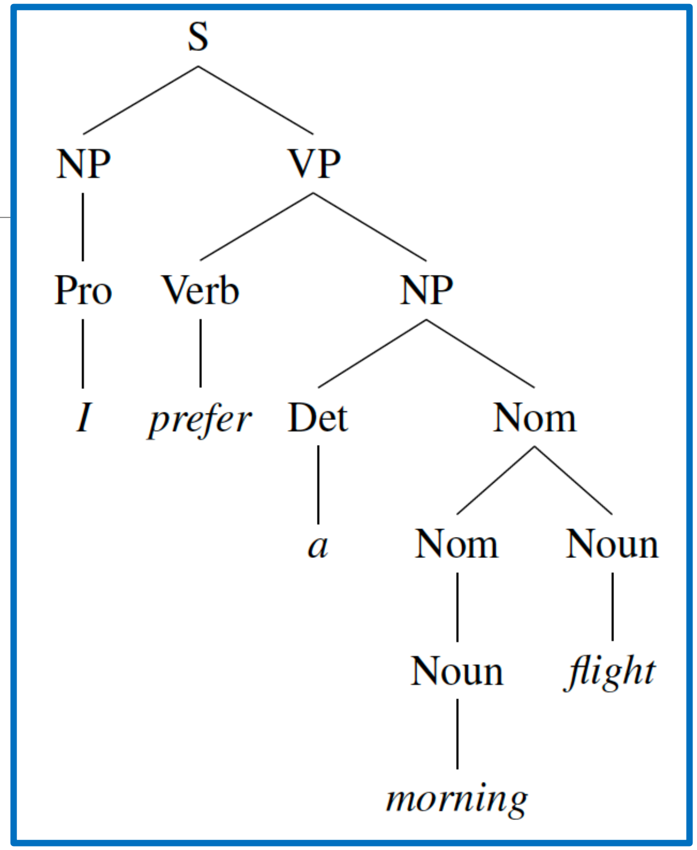
Noun	$\rightarrow$	*flights* | *breeze* | *trip* | *morning*
Verb	$\rightarrow$	*is* | *prefer* | *like* | *need* | *want* | *fly*
Adjective	$\rightarrow$	*cheapest* | *non-stop* | *first* | *latest* | *other* | *direct*
Pronoun	$\rightarrow$	*me* | *I* | *you* | *it*
Proper-Noun	$\rightarrow$	*Alaska* | *Baltimore* | *Los Angeles* | *Chicago* | *United* | *American*
Determiner	$\rightarrow$	*the* | *a* | *an* | *this* | *these* | *that*
Preposition	$\rightarrow$	*from* | *to* | *on* | *near*
Conjunction	$\rightarrow$	*and* | *or* | *but*

Grammar Rules			Examples
S	$\rightarrow$	*NP VP*	I + want a morning flight
NP	$\rightarrow$	*Pronoun*	I
	|	*Proper-Noun*	Los Angeles
	|	*Det Nominal*	a + flight
Nominal	$\rightarrow$	*Nominal Noun*	morning + flight
	|	*Noun*	flights
VP	$\rightarrow$	*Verb*	do
	|	*Verb NP*	want + a flight
	|	*Verb NP PP*	leave + Boston + in the morning
	|	*Verb PP*	leaving + on Thursday
PP	$\rightarrow$	*Preposition NP*	from + Los Angeles

From Jurafsky & Martin

An Example Using CFGs

Grammar Rules			Examples
S	→	*NP VP*	I + want a morning flight
NP	→	*Pronoun*	I
	|	*Proper-Noun*	Los Angeles
	|	*Det Nominal*	a + flight
Nominal	→	*Nominal Noun*	morning + flight
	|	*Noun*	flights
VP	→	*Verb*	do
	|	*Verb NP*	want + a flight
	|	*Verb NP PP*	leave + Boston + in the morning
	|	*Verb PP*	leaving + on Thursday
PP	→	*Preposition NP*	from + Los Angeles



From Jurafsky & Martin

$[_S\ [_{NP}\ [_{Pro}\ \text{I}]]\ [_{VP}\ [_V\ \text{prefer}]\ [_{NP}\ [_{Det}\ \text{a}]\ [_{Nom}\ [_N\ \text{morning}]\ [_{Nom}\ [_N\ \text{flight}]]]]]]$

Bracketed Notation

Example: Larger English CFG

Grammar
$S \rightarrow NP\ VP\ .$
$S \rightarrow NP\ VP$
$S \rightarrow$ " S " , $NP\ VP\ .$
$S \rightarrow$ -NONE-
$NP \rightarrow DT\ NN$
$NP \rightarrow DT\ NNS$
$NP \rightarrow NN\ CC\ NN$
$NP \rightarrow CD\ RB$
$NP \rightarrow DT\ JJ\ ,\ JJ\ NN$
$NP \rightarrow PRP$
$NP \rightarrow$ -NONE-
$VP \rightarrow MD\ VP$
$VP \rightarrow VBD\ ADJP$
$VP \rightarrow VBD\ S$
$VP \rightarrow VBN\ PP$
$VP \rightarrow VB\ S$
$VP \rightarrow VB\ SBAR$
$VP \rightarrow VBP\ VP$
$VP \rightarrow VBN\ PP$
$VP \rightarrow TO\ VP$
$SBAR \rightarrow IN\ S$
$ADJP \rightarrow JJ\ PP$
$PP \rightarrow IN\ NP$

Number	Tag	Description
1.	CC	Coordinating conjunction
2.	CD	Cardinal number
3.	DT	Determiner
4.	EX	Existential there
5.	FW	Foreign word
6.	IN	Preposition or subordinating conjunction
7.	JJ	Adjective
8.	JJR	Adjective, comparative
9.	JJS	Adjective, superlative
10.	LS	List item marker
11.	MD	Modal
12.	NN	Noun, singular or mass
13.	NNS	Noun, plural
14.	NNP	Proper noun, singular
15.	NNPS	Proper noun, plural
16.	PDT	Predeterminer
17.	POS	Possessive ending
18.	PRP	Personal pronoun
19.	PRP$	Possessive pronoun
20.	RB	Adverb
21.	RBR	Adverb, comparative
22.	RBS	Adverb, superlative
23.	RP	Particle
24.	SYM	Symbol
25.	TO	to
26.	UH	Interjection
27.	VB	Verb, base form
28.	VBD	Verb, past tense
29.	VBG	Verb, gerund or present participle
30.	VBN	Verb, past participle
31.	VBP	Verb, non-3rd person singular present
32.	VBZ	Verb, 3rd person singular present
33.	WDT	Wh-determiner
34.	WP	Wh-pronoun
35.	WP$	Possessive wh-pronoun
36.	WRB	Wh-adverb

Table Source:
https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

Interpretation of Parsing Rules

- generation (production): $S \rightarrow NP\ VP$
- parsing (comprehension): $S \leftarrow NP\ VP$
- verification (checking): $S = NP\ VP$
- CFGs are <u>declarative</u> – tell us *what* the well-formed structures & strings are
- Parsers are <u>procedural</u> – tell us *how* to compute the structure(s) for a given string

From Robert C. Berwick

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

Lecture 5: Concluding Comments

- Resume exercise: completed
- We looked at word representation
  - As characters
  - As vectors
    - Structured representation
    - Statistical representation
- We looked at parsing and roles it plays: verifying , generating, recognizing
  - Many types of parsing
  - Shallow parsing for quick NLP tasks
  - Phrase structure parsing
  - Dependency parsing

Concluding Segment

Discussion: Course Project

Theme: Analyze quality of official information available for elections in 2024 [in a state]

- Take information available from
  - Official site: State Election Commissions
  - Respected non-profits: League of Women Voters
- Analyze information
  - State-level: Analyze quality of questions, answers, answers-to-questions
  - Comparatively: above along all states (being done by students)
- Benchmark and report
  - Compare analysis with LLM
  - Prepare report

- Process and analyze using NLP
  - Extract entities
  - Assess quality – metrics
    - Content – *Englishness*
    - Content – *Domain* -- election
  - … other NLP tasks
  - Analyze and communicate overall

Major dates for project check

- Sep 10: written – project outline
- Oct 8: in class
- Oct 31: in class // LLM
- Dec 5: in class // Comparative

Review current states chosen by others

Project Discussion

1. Go to Google spreadsheet against your name
2. Enter the **state** you will focus on for course project

1. Create a private Github repository called "CSCE771-Fall2024-<studentname>-Repo". Share with Instructor (biplav-s) and TA (vr25)
2. Create Google folder called "CSCE771-Fall2024-<studentname>-SharedInfo". Share with Instructor (prof.biplav@gmail.com) and TA (rawtevipula25@gmail.com)
3. Create a Google doc in your Google repo called "Project Plan" and have the following by Friday (Aug 30, 2024)

Timeline

1. Title: Analyze quality of official information available for elections in 2024 in <state>
2. Data need:
   1. Official: state's election commission
   2. LWV: https://www.vote411.org/
3. Methods:
4. Evaluation:
5. Milestones
   - Sep 10: written and feedback
   - Oct 8: in class
   - Oct 31: in class
   - Dec 5: in class

Discussion: Course Project

- Expectations
  - Apply methods learned in class or of interest to a problem of interest
  - Be goal oriented: aim to finish, be proactive, be innovative
  - Do top-class work: code, writeup, presentation
- Typical pitfalls
  - Not detailing out the project, assuming data
  - Not spending enough time
- What will be awarded
  - Results and efforts (balance)
  - Challenge level of problem

Review current states chosen by others

Course Project – Deadlines and Penalty Rubric

- Penalty
  - Missing milestones: **[-10%]**
  - Maximum: **[-40%]**
- Bonus possible
  - if two or more states considered

Timeline

1. Title: Analyze quality of official information available for elections in 2024 in <state>
2. Data need:
   1. Official: state's election commission
   2. LWV: https://www.vote411.org/
3. Methods:
4. Evaluation:
5. Milestones
   - Sep 10: written and feedback
   - Oct 8: in class
   - Oct 31: in class
   - Dec 5: in class

About Next Lecture – Lecture 6

Lecture 6:

- Shallow/ Deep parsing
- Statistical Parsing

4	Aug 29 (Th)	NLP Tasks, Case Study – Business Application		
5	Sep 3 (Tu)	Parsing, Paper 1 discussion; project topics review		Practice exercise
6	Sep 5 (Th)	Project topics review, statistic Parsing		
7	Sep 10 (Tu)	Statistical parsing, QUIZ		Quiz 1, **Project Check**
8	Sep 12 (Th)	Evaluation, Semantics		Coding running example
9	Sep 17 (Tu)	Semantics Machine Learning for NLP, Evaluation - Metrics		Code: scikit f1 score package, Code: ConceptIO
10	Sep 19 (Th)	Towards Language Model: Vector embeddings, Embeddings, CNN/ RNN		Code: embedding, genism word vector, tf-idf