CSCE 771: Computer Processing of Natural Language

Lecture 9: Semantics, ML Basics (Review)

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

17TH SEPTEMBER, 2024

Acknowledgement: Used materials by Jurafsky & Martin, 2nd edition

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

Organization of Lecture 9

- Opening Segment
  - Announcement
- Main Lecture
- Concluding Segment
  - Reading material:
  - About Next Lecture – Lecture 10

Main Section
- Semantics
  - Shallow: similarity, relatedness; frames
  - Propbank
  - Deep: AMR
  - ConceptNet
- ML Basics
  - Supervised learning

Announcements

GUEST LECTURES ON LANGUAGE MODELS

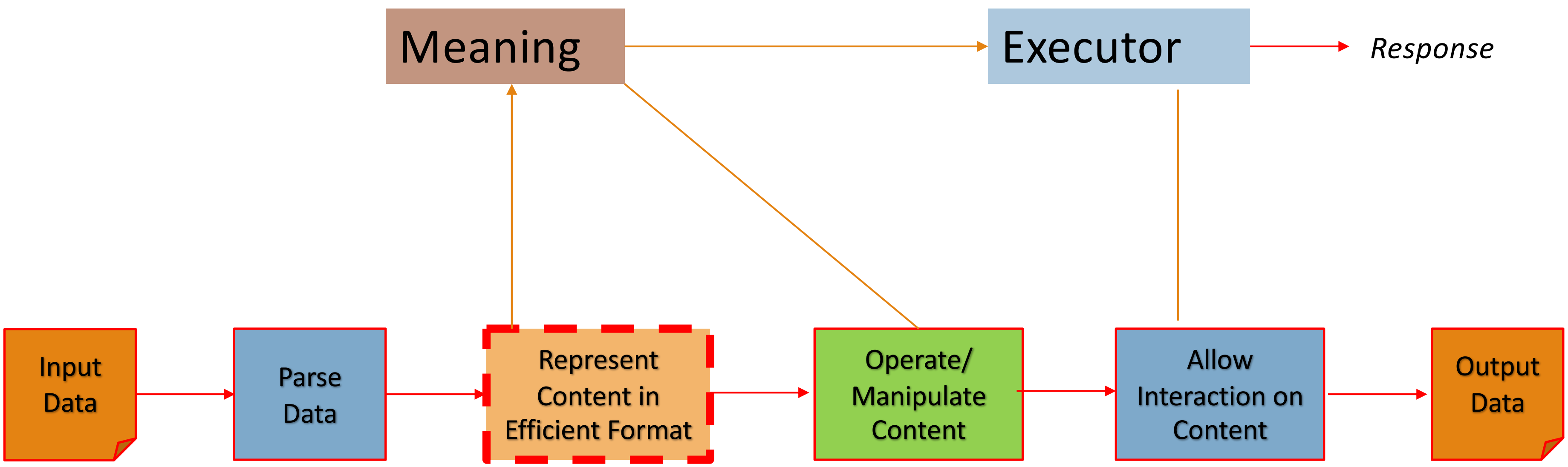
Sep 24 (Tu)	Language Model – PyTorch, BERT, {Resume data, two tasks} **– Guest Lecture**
Sep 26 (Th)	Language Model – Finetuning, Mamba - **Guest Lecture**
Oct 1 (Tu)	Language model – comparing arch, finetuning - **Guest Lecture**
Oct 3 (Th)	Language model – comparison of results, discussion, ongoing trends– **Guest Lecture**

QUIZ 1

- Submit using Black board
- Includes resume exercise
- Was due Monday, Sep 16, 2024

Main Lecture

Semantics, Parsing and Representation



Semantics

- ***lexical semantics**:* studies word meanings and word relations, and
- ***formal semantics**:* studies the logical aspects of meaning, such as sense, reference, implication, and logical form
- ***conceptual semantics**:* studies the cognitive structure of meaning

Source: Jurafsky & Martin,
Wikipedia (https://en.wikipedia.org/wiki/Semantics)

From Text to Meaning

- Shallow semantics
  - Input: text
  - Output: ***lexical semantics***
- Deep semantics
  - Input: text
  - Output: ***formal semantics***

Source: Abstract Meaning Representation for Sembanking, https://amr.isi.edu/a.pdf

LOGIC format:

$\exists$ w, b, g:
instance(w, want-01) $\wedge$ instance(g, go-01) $\wedge$
instance(b, boy) $\wedge$ arg0(w, b) $\wedge$
arg1(w, g) $\wedge$ arg0(g, b)

AMR format (based on PENMAN):

```
(w / want-01
  :arg0 (b / boy)
  :arg1 (g / go-01
          :arg0 b))
```

GRAPH format:

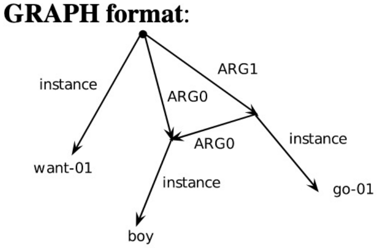


Figure 1: Equivalent formats for representating the meaning of "The boy wants to go".

Frames, Slots: Frame Semantics

- Examples
  - "John sold a car to Mary”
  - "Mary bought a car from John”
  - “Mary paid John a undisclosed amount to get his car”
- To understand a word, one needs to understand the knowledge related to the word
  - In example: sell, buy, pay
- Capture knowledge in structures called **semantic frames** which have placeholders called slots (variables)
  - During parsing of sentences, values are filled
- **Frame semantics** is a theory of linguistic meaning developed by Charles J. Fillmore; related notion is semantic parsing

PropBank FrameSet

- A repository of formalized predicates
  https://propbank.github.io/

<roleset id="care.01”
name="having an opinion, feeling tenderly/strongly for/about">
<roleset id="care.02"
name="liking/desiring/wanting">
<roleset id="care.03"
name="tending, taking care of">
<roleset id="care.04"
name="being cautious, taking care to">

Example: Care
https://github.com/propbank/propbank-frames/blob/main/frames/care.xml

Hindi – भेजा - ***Beja***

Credits: https://verbs.colorado.edu/propbank/framesets-hindi/Beja-v.html

Example: Hindi Propbank

Roleset id: Beja.03 , *to make someone send something to someone*

- **Argc**: *the causer- the one who makes someone send something*
- **Arga**: *the intermediate causer*
- **Arg0**: *the agent- the one who sends something*
- **Arg2**: *the one to whom something is sent*
- **Arg1**: *the thing that is sent*

Roleset id: Beja.01 , *to send, transport, ship something*

- **Arg0**: *the one who sends something*
- **Arg2**: *the recipient to whom something is sent*
- **Arg1**: *the thing that is sent*

Roleset id: Beja.04 , *to make someone send something to someplace*

- **Argc**: *the causer- the one who makes someone send something*
- **Arga**: *the intermediate causer*
- **Arg0**: *the agent- the one who sends something*
- **Arg2-gol**: *the place where something is sent*
- **Arg1**: *the thing that is sent*

Roleset id: Beja.02 , *to send, transport, ship something*

- **Arg0**: *the one who sends something*
- **Arg2-gol**: *the place where something is sent*
- **Arg1**: *the thing that is sent*

Abstract Meaning Representation (AMR)

- Example: “The boy wants to go”
- AMR concepts are
  - English words (“boy”),
  - PropBank framesets (“want-01”), or
  - special key-words.
- Keywords include special entity types(“date-entity”, “world-region”, etc.), quantities(“monetary-quantity”, “distance-quantity”, etc.)
- logical conjunctions (“and”, etc).
- AMR uses approximately 100 relations

Source: Abstract Meaning Representation for Sembanking, https://amr.isi.edu/a.pdf

LOGIC format:

$\exists$ w, b, g:
instance(w, want-01) $\wedge$ instance(g, go-01) $\wedge$
instance(b, boy) $\wedge$ arg0(w, b) $\wedge$
arg1(w, g) $\wedge$ arg0(g, b)

AMR format (based on PENMAN):

```
(w / want-01
  :arg0 (b / boy)
  :arg1 (g / go-01
          :arg0 b))
```

GRAPH format:

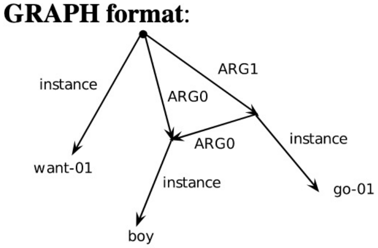


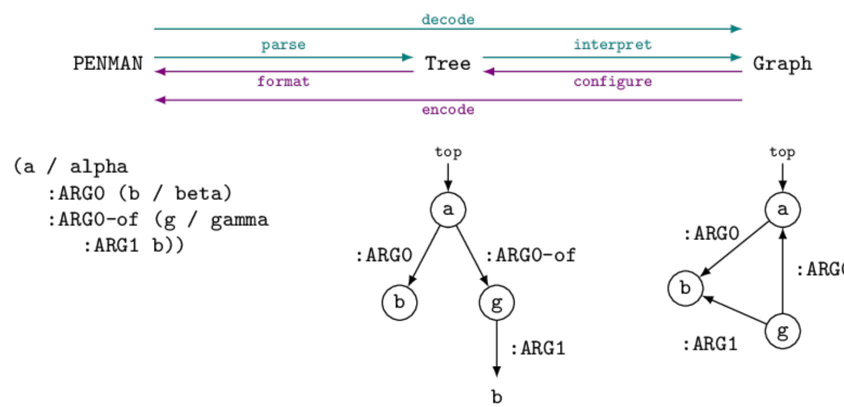
Figure 1: Equivalent formats for representating the meaning of “The boy wants to go”.

PENMAN Notation

```
;  ┌──────────────────────── Variable (this one is the graph's top)
;  │   ┌──────────────────── Instance relation
;  ┴ ──┴──
  (d / drive-01
;  ┬ ──┬──
;  │   └──────────────────── Concept (node label)
;  └──────────────────────── Indicates the node's concept
;      ┌──────────────────── Edge relation
;  ────┴────
   :ARG0 (h / he)
;  ─┬─
;   └─────────────────────── Role (edge label)
   :manner (c / care-04
;          ┌──────────────── Attribute relation
;      ────┴────
     :polarity -))
;          ┬
;          └── Atom (or "constant")
```

<node> ::= '(' <id> '/' <node-label> <edge>* ')'
<edge> ::= ':'<edge-label> (<const>|<id>|<node>)

Credit: https://penman.readthedocs.io/en/latest/notation.html



Credit: https://penman.readthedocs.io/en/latest/structures.html

Sample Code – PENMAN/ AMR

Sample code:
https://github.com/biplav-s/course-nl-f22/blob/main/sample-code/l9-semantics/PENMAN%20Notation%20-%20AMR.ipynb

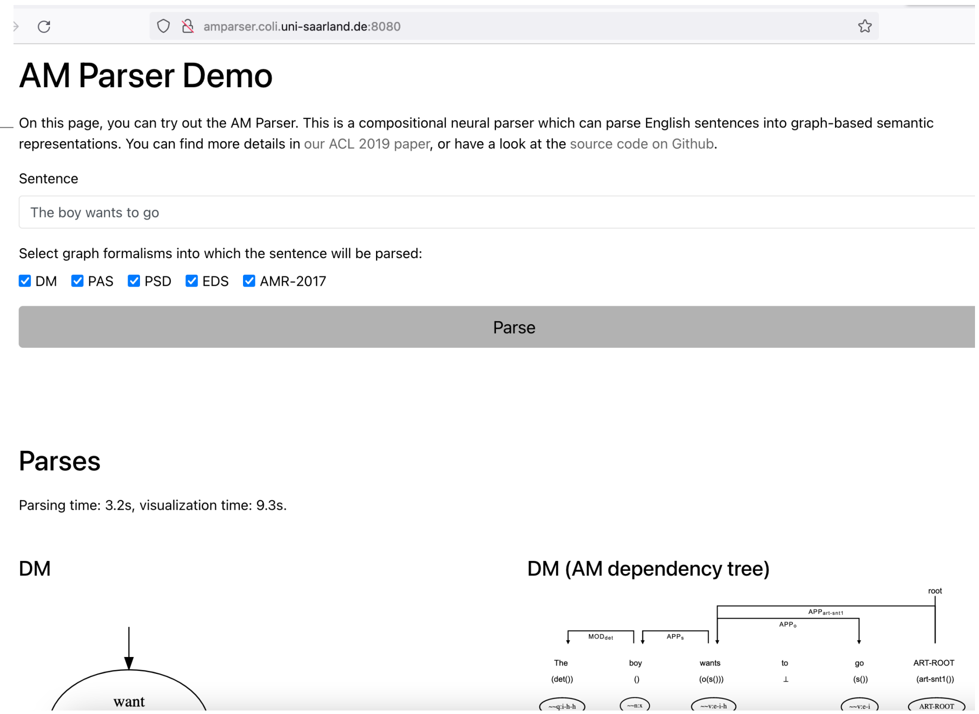
AMR Tools/ Libraries

- Libraries
  - Spacy: https://spacy.io/universe/project/amrlib
  - IBM Research: https://github.com/IBM/transition-amr-parser

- Tools
  - AMR Eager: https://bollin.inf.ed.ac.uk/amreager.html

AMR Demo

http://amparser.coli.uni-saarland.de:8080/



Exercise: 5 mins

- Try your sentences online
- Look at output in different formats

Semantic Parsing

- Shallow semantic parsing
  - Also called: slot-filling or frame semantic parsing
  - "show me flights from Boston to Dallas"
- Deep semantic parsing
  - “show me flights from Boston to anywhere that has flights to Dallas”
  - Reference to quantifiers

Applications
- Paraphrasing
- Machine comprehension
- Question-answering
- Dialog

References:
- ACL 2020 Tutorial on Semantic Parsing
- https://en.wikipedia.org/wiki/Semantic_parsing

Semantic Parsing

Language to Meaning

Semantic
Parsing

Recover **complete**
meaning
representation

More informative

Task-specific parsing

Example Task

Database Query

What states
border Texas?

Oklahoma
New Mexico
Arkansas
Louisiana

Source:
ACL 2020 Tutorial on Semantic Parsing

Resources: Semantic Parsing Libraries

- Open Sesame
  - Given English sentence, predicts FrameNet frames
  - https://github.com/swabhs/open-sesame

- AMRLib
  - Python library for AMR parsing, generation and visualization simple
  - https://github.com/bjascob/amrlib

Review: 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) different meanings (i.e., senses)
  - Duck, sharp

Source: Mausam

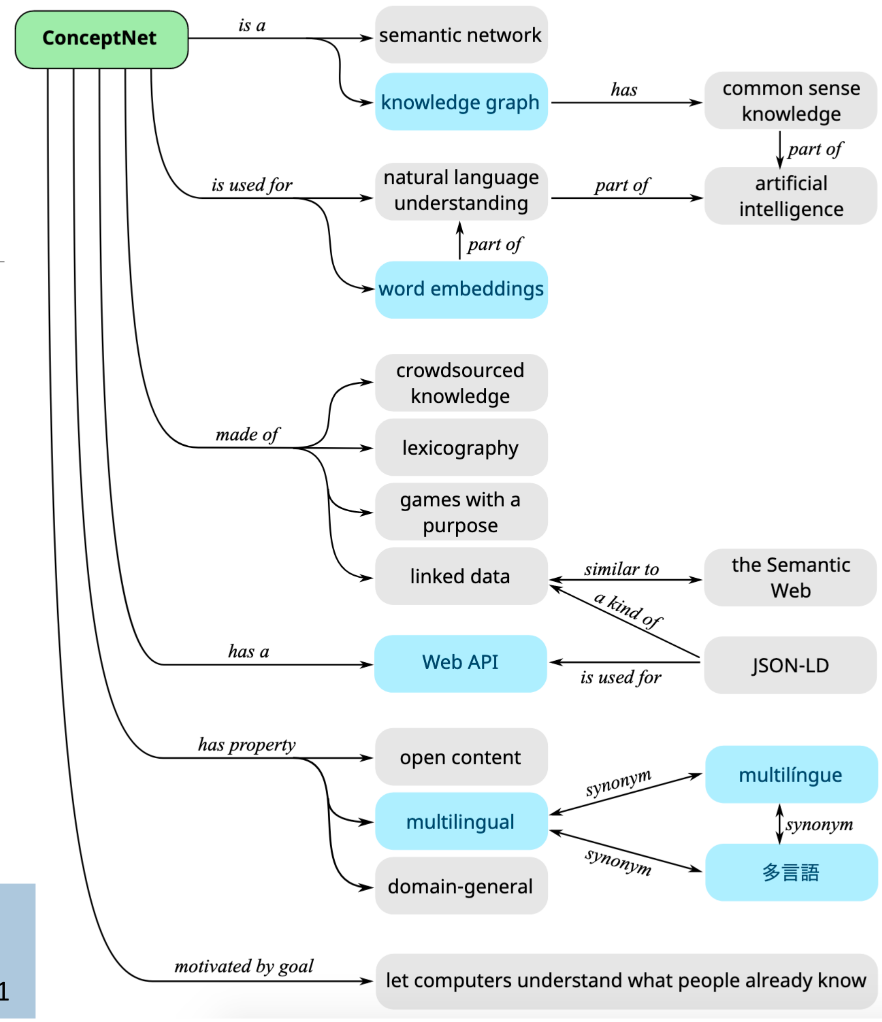
More Terms

- **Affective meanings** or **connotation**: word's meaning that are related to a writer or reader's emotions, sentiment, opinions, or evaluations
  - Positive evaluation: good, happy
  - Negative evaluation:
- **Sentiment**: Positive or negative evaluation expressed through language
  - Scherer's Typology of Affective States

Source: Jurafsky & Martin

ConceptNet

- NLP focused graph knowledge graph that connects words and phrases of natural language with labeled edges.
- Concepts collected from experts, crowd-sourcing, and games with a purpose
- Supports multiple languages
- Provides ”loose” semantics - relatedness



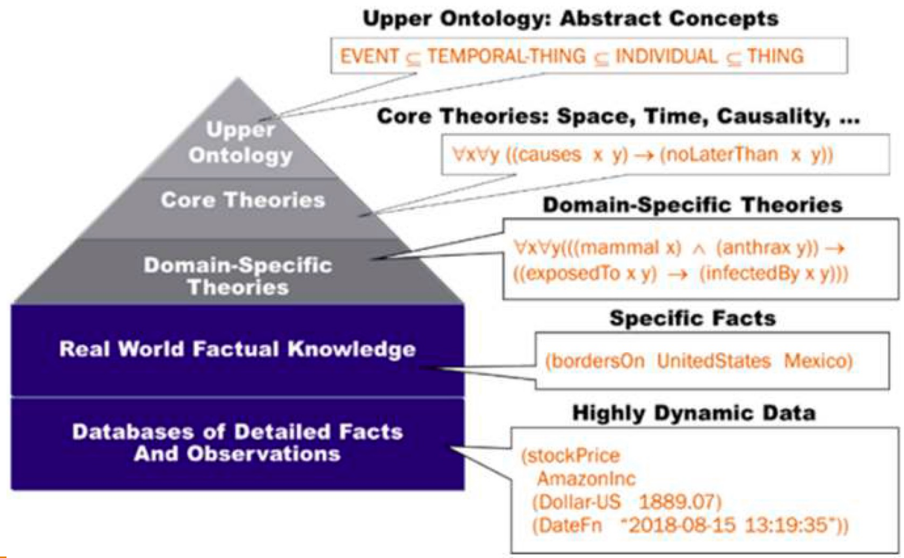
Details: http://conceptnet.io/, https://github.com/commonsense/conceptnet5/wiki,
Paper: https://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/viewFile/14972/14051

Demonstration - ConceptNet

Examples:
- Concepts:
  - Word: http://conceptnet.io/c/en/word ,
  - duck: http://conceptnet.io/c/en/duck
- Relationships:
  - http://conceptnet.io/s/resource/wordnet/rdf/3.1

Project CYC

- A large ontology to capture the world and human common sense
  - Doug Lenat lead team of computer scientists, computational linguists, philosophers, and logicians
  - Identify and formally axiomatize the tens of millions of rules about world
  - 35+ years effort by Cycorp
- Reasoners on the ontology to make decisions
  - 1000+ specialized reasoners



Details: https://www.cyc.com/

Source: Cyc White Paper

Cyc Details

- Ontology of about 1.5 million general concepts (e.g., taxonomically "placing" terms like eyes, sleep, night, person, unhappiness, hours, posture, being woken up, etc.);
- More than 25 million general rules and assertions involving those concepts
  - *"Most people sleep at night, for several hours at a time, lying down, with their eyes closed, they can be awakened by a loud noise but don't like that, "*
- Domain-specific extensions to the common sense ontology and knowledge base
  - healthcare, intelligence, defense, energy, transportation and financial services.
- Promoting synergistic use of ontology and learning based approaches (now)

Source: White Paper – Cyc Technology Overview

Machine Language Basics (Review)

Machine Learning – Insights from Data

- Descriptive analysis
  - Describe a past phenomenon
  - **Methods**: classification, clustering, dimensionality reduction, anomaly detection, neural methods
- Predictive analysis
  - Predict about a new situation
  - **Methods**: time-series, neural networks
- Prescriptive analysis
  - What an agent should do
  - **Methods**: simulation, reinforcement learning, reasoning

- New areas
  - Counterfactual analysis
  - Causal Inferencing
  - Scenario planning

Machine Learning – Label Based View

- Label available – Supervised Learning
  - Example: Classification

- Label unavailable – Unsupervised Learning
  - Example: Clustering

Common Textual Data Processing Steps for ML

- Input: strings / documents/ corpus
- Processing steps (task dependent / optional - *)
  - Parsing
  - Word pre-processing
    - Tokenization – getting tokens for processing
    - Normalization* - making into canonical form
    - Case folding* – handling cases
    - Lemmatization* – handling variants (shallow)
    - Stemming* – handling variants (deep)
  - Semantic parsing – representations for reasoning with meaning *
  - Embedding – creating vector representation*

Impt: 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

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

Classification

Classifier Method Types

- Individual methods
  - Decision Tree
  - Naïve Bayes
- Ensemble
  - Bagging: Aggregate classifiers ("bootstrap aggregation" => bagging)
    - Random Forest
    - Samples are chosen with replacement (bootstrapping), and combined (aggregated) by taking their average
  - Gradient Boosting: aggregate to turn weak learners into strong learners
    - Boosters (aggregators) turn weak learners into strong learners by focusing on where the individual weak models (decision trees, linear regressors) went wrong
    - Gradient Boosting
    - XGBoost: "eXtreme Gradient Boosting."

Source:

- Data Mining: Concepts and Techniques, by Jiawei Han and Micheline Kamber
- https://towardsdatascience.com/getting-started-with-xgboost-in-scikit-learn-f69f5f470a97

ML - Supervised

- By Example:
  - https://github.com/biplav-s/course-nl/blob/master/l9-ml-review/Classification%20-%20Fake%20news.ipynb

- Fake news dataset

Metrics: Accuracy, Precision, Recall

	Predicted class		
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Accuracy =
(TP+TN)/
(TP+FP+FN+TN)

Precision =
(TP)/
(TP+FP)

Recall =
(TP)/
(TP+FN)

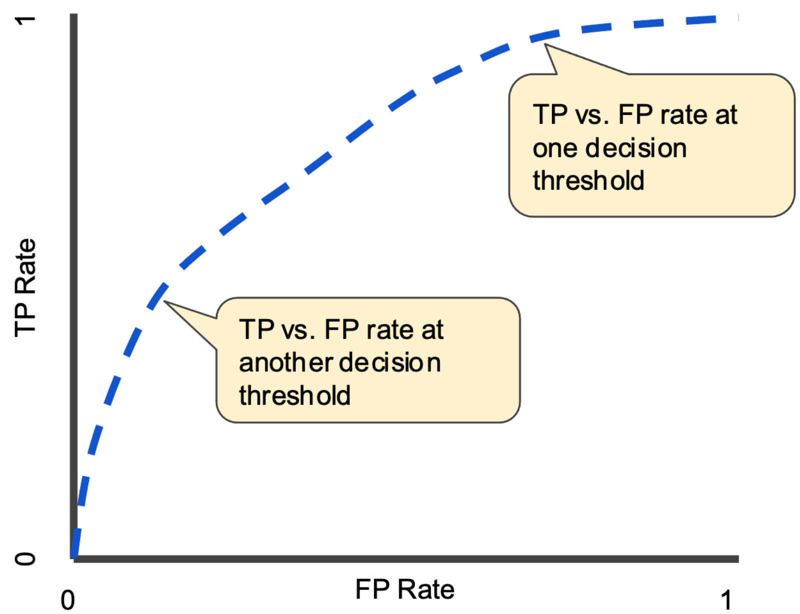
F1 Score: *Harmonic Mean*

1/F1 = 1/Precision + 1/Recall

F1 = 2*(Recall * Precision) /
(Recall + Precision)

ROC – Receiver Operating Characteristic curve

An ROC curve plots TPR vs. FPR at different classification thresholds



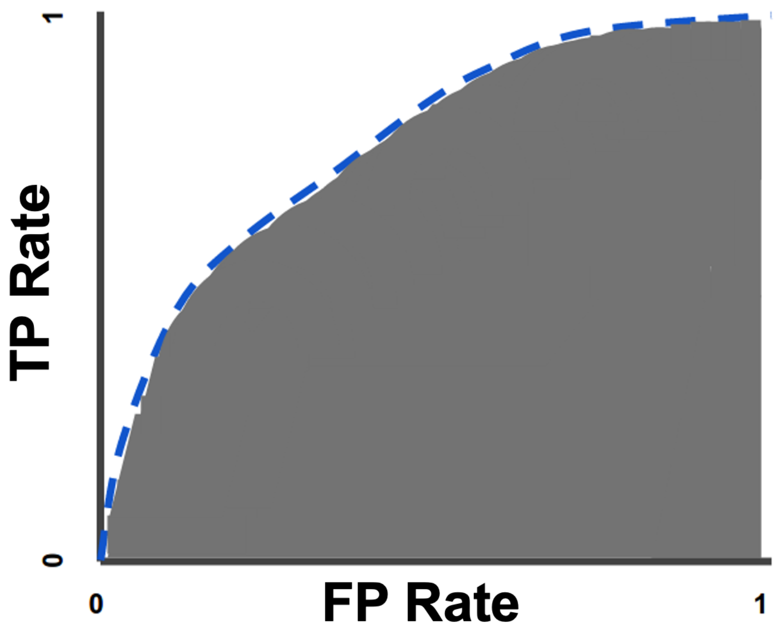
True Positive Rate = Recall = (TP)/ (TP+FN)

False Positive Rate = (FP)/ (FP+TN)

		Predicted class	
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

Source: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

AUC – Area Under the ROC Curve



- Aggregate measure of performance across all possible classification thresholds.
- Interpretation: probability that the model ranks a random positive example more highly than a random negative example

Not helpful when the **cost** of false negatives vs. false positives are asymmetric

Source: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

References

- Blogs: https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/
- Google: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

AutoML – Automated Machine Learning

- Automate data preparation
- Automate feature selection
- Hyperparameter tuning

- Demo:
  - IBM Auto AI - https://www.youtube.com/watch?v=ILCsbh9IKT0

Lecture 9: Concluding Comments

- We reviewed how to give semantics to words and documents
- Can be human supervised or learning based or combined
- Can be generic or task-oriented

Concluding Segment

Course Project

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

Obtaining Election Data

Here are a few things to do:
A) **Official data** backed by laws: state election commission
a) Find the state's election commission
b) Find the Q/As they provide. They may be as FAQs or on different web pages.
c) Collect the Q/A programmatically

B) **Secondary data** sources: non-profit
a) Find Q/As from Vote 411 which is supported by the non-profit: LWV.

For reference, for SC,

A) Official - https://scvotes.gov/voters/voter-faq/

B) Secondary - https://www.vote411.org/south-carolina

For extraction, one or more approaches:
- Manually annotating
- BeautifulSoup,
- Tika
- or other open source libraries.

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 10

Lecture 10 Outline

- Machine Learning for NLP
  - Supervised learning
  - Unsupervised learning
  - Neural methods
- Language Models

7	Sep 10 (Tu)	Statistical parsing, QUIZ
8	Sep 12 (Th)	Evaluation, Semantics
9	Sep 17 (Tu)	Semantics, Machine Learning for NLP, Evaluation - Metrics
10	Sep 19 (Th)	Towards Language Model: Vector embeddings, Embeddings, CNN/ RNN
11	Sep 24 (Tu)	Language Model – PyTorch, BERT, {Resume data, two tasks} **– Guest Lecture**
12	Sep 26 (Th)	Language Model – Finetuning, Mamba - **Guest Lecture**
13	Oct 1 (Tu)	Language model – comparing arch, finetuning - **Guest Lecture**
14	Oct 3 (Th)	Language model – comparison of results, discussion, ongoing trends– **Guest Lecture**
15	Oct 8 (Tu)	PROJ REVIEW