Glossary

Week 1

- **Natural Language** - A language that emerged gradually and evolved over time. Even a newer language like French Sign Language (~200 years ago) evolved from the deaf community in France organically and is considered a natural language.
- **Artificial Language** - A language purposefully put together by people as a means of communication that has its own rules and structure. NOT a direct swap out like Morse Code which is just any given language spelled out.
- **NLU - Natural Language Understanding**, taking in a language and building data from it
- **NLG - Natural Language Generation** - taking data and building natural language from it
- **Text Annotations** - Taking Information from one document and pulling NLU data from it
- **Corpus Analytics** - Taking Information from a set of documents and taking information from the group
- **Query Repair** - *'Did you mean?*' Fixing typos or incorrect words
- **Query Refinement** - Taking an ambiguous query and making it more granular
- **Results Post Processing** - Pulling out information from a search result and displaying it for users to determine if it's the link they are looking for

Week 2

- **Words**: what's a real word, how it is spelled
- **Grammar**: how words combine to make sentences
- **Meanings**: what words and sentences mean in context
- **Inferences**: how people draw more out of what is said than what is literally stated

- Levels of NLP Analysis
- Words ->**Lexical analysis**
- Grammar-> **Syntactic analysis**
- Meanings-> **Semantic analysis**
- Inferences-> **Discourse/entailment analysis**

- **Morphology** means the study of “morphemes,” which are the units that our words are made of.
- **Collocations** - words that are in the corpus together
- **Word Senses** - the meaning of words while reading, those with multiple are polysemous
- **Domain Relations** - Ordering of word senses based on what they commonly mean in a given domain
- **POS Tagging** - Parts of Speech tagging; Commonly uses Penn Treebank Tagset
- **lemma** is the canonical (conventional) form that represents a set of related word forms, e.g., for run, runs, ran, running, the lemma is “run.”

Week 3

Shallow vs. deep

- **Shallow** - pick out a few elements needed for my application
- **Deep** - very robust, exhaustive representation of text meaning

Statistical vs. symbolic

- **Statistical** - leverage powerful, complex statistical methods
- **Symbolic** - use rules that can be operated on by strict logic

Feature engineering vs. feature learning

- **Engineering** - involve human experts to engineer features
- **Learning** - use experts only to establish a training set

Top-down vs. bottom-up

- **Top-Down**- start with high-level classifications and text characteristics, then gradually break them down into more detail
- **Bottom-Up**- Revers of Top-Down

Transparent vs. opaque (AI vs. XAI)

- **Transparent-** engineer a system that is explainable to an intelligent SME
- **Opaque** - build something only my fellow data scientists and engineers could understand

Week 4

ETL (“extract, transform, load”)

Week 5

- **Sentence segmentation** - sentence tokenization
- **Lexical analysis** - word tokenization
- **Text Normalization -** Handling/Removing: contractions, stopwords, misspellings; Stemming words if neccessary
- **Function words** - lack content in text, they do not answer the 6Ws, but they may structure
- **closed class** - fixed list to which no new words are added
- **Content Words** - do answer the 6Ws
- **open class** - unfinished list to which new words are readily added.
- **stemmer -** is tool that given a word, returns its stem:“running” à“run”

Primary features•Require us only to examine the document itself•Examples:•Word frequencies•Collocations (n-grams)

Secondary features•Require us to compare features of the document to those of other documents•Examples:•Differential frequency analysis (TF/IDF)•Relative lexical diversity•Reading level

differential frequency analysis—noticing phrases that are more common in a target text than in most other texts

- **Term frequency** measures how often a term occurs in a document—in raw form, it is a simply a word count divided by the total number of words in the document.
- **Document frequency** measures how common the term is within a domain represented by a corpus of documents (C).
- $$idf(t) = 1 + \log \frac{C}{1 + df(t)}$$

- **Inverse document frequency** is computed by dividing the total number of documents in the corpus by the number of documents containing our target term, and applying a log scale

Week 6

Lexicon - machine-readable dictionary and carries the information needed to perform major NLP functions: POS, inflection, t vs it verbs
lexical knowledge base (“lexical KB”) - breaks words into senses, and linking senses to senses via relations
Synonym/antonym

- **Synonym** - Two word that mean the same thing
- **Antonym** - Two words with opposing meanings

Hyponym/hypernym

- **Hyponym** - A word more specific than a general term (spoon vs cutlery)
- **Hypernym** - A less specific term (dog vs dalmatian)

Holonym/meronym

- **Holonym** - A term that denotes a whole, a part of which is denoted by a second term. The word "face" is a *holonym* of the word "eye".
- **Meronym** - a term which denotes part of something but which is used to refer to the whole of it, e.g. *faces* when used to mean *people* in *I see several familiar faces present*.

ontological distance -number of nodes “travelled” through the hierarchy to get from node A to node B
Monosemy - having only one sense
Polysemy - having more than one sense

Week 7

Hidden Markov models - Map POS based on probability and preceding words
Conjunctive Elimination - Combo candidate validation/elimination: eliminate a candidate NNP-collocation IF-AND-ONLY-IF not found in the NE database AND it has low frequency

NER - Named Entity Recognition - Finding the people, place, locations, and things that have names (like companies). Identifies the beginning (B) of an entity and continuation the name contains multiple words (I). All others are marked as (O).

WEEK 8

Parse Tree - Allows us to see the parts of speech in context of the possible noun and verb phrases. We can see how modifiers or direct objects etc could belong to the noun or verb phrase. Built off rules

Week 9 - Midterm

Week 10

Vector Semantics defines semantics & interprets word meaning to explain features such as word similarity. Its central idea is: Two words are similar if they have similar word context.
Medium Source

Term-context matrix - Each column is a context of a candidate word (e.g., boot, root). Each element of the matrix holds the number of occurrences of the th word of the vocabulary in the th context. For example, the word plant occurs three times in the 7 th context, which is the 3 rd context labeled with root.
Source

Semantic Similarity - Semantic similarity is a metric defined over a set of documents or terms, where the idea of distance between items is based on the likeness of their meaning or semantic content as opposed to lexicographical similarity.
Wiki Source

Word Similarity - how closely related words are (distance has multiple calculation to follow)

Structural Approach

Distance - semantic graph: Overlap in parse contexts and find the ontological distance. Start from word A node and traverse to node B.

Problem: Tree maintenance has to be managed.

Statistical approach

Jaccard Similarity: Use it where duplication of words don't matter.
Equation:

Jaccard similarity:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

Cosine Similarity: Measures the cosine between the angle of two vectors. Put the sentences into vector form.
Acute: Similar Scores; Right: Unrelated; Obtuse: Opposite score
Vectorize words:
TF (Term -Frequency): Just count words, but normalize the frequency relative to individual sentences, because it favors long sentences because they naturally will have higher counts for words.
TF-IDF (term-frequency - inverse document frequency) takes into account that rare words are more important

PMI - Pointwise mutual information
Metric to ID events that frequently co-occur (words that go together)

$$PMI(w_i, w_j) = \log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}$$

Want to strive for dense vectors, avoid empty storage.
Sparse to dense: SVD Singular Value Decomposition to turn these sparse vectors into dense vectors (Latent Semantic Analysis)

Use neural models to explicitly learn a dense vector representation (embedding) word2vec, glove

Term Dcument matrix is a 2D table:

	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

- Each cell contains the frequency count of the term (word) t in document d : tf(sub t,d)
- Each column is a vector of counts over words, representing a document
- Each row is a vector of counts over documents, representing a word

- Two documents are similar if their vectors are similar (you would compare columns' jaccard or cosine distance)
- Two words are similar if their vectors are similar

Sparse vectors - most entries are zero
Dense Vectors - most entries are non zero

Context and co-occurences

Decide on a fixed vocab of N context words c1...cn
Context words should occur freq enough in corpus to get reliable co-occurance counts

Define Nearby - ie, if within +- 5 words

Count be binary: Nearby or not?
Frequency Count: How often is it within our nearby window?

These are both very sparse vectors
Word vectors with word2vec
Vector of weights: 1 of N (one-hot encoding:
Word vector: corresponding element is set to one, and all elements are zero

Given Vocab : King, queen, man, woman, child
Queen [0,1,0,0,0]
This creates a distribution of words such that they are more stored as a vector representing it's meaning: King - Man + woman = Queen

Continuous Bag of Words (CBOW) - Sliding window of context words. Focus words and context (our +- window). Each word is 1-hot encode, single hidden layer and output. Maximize probability of encountering our focus word. This is how we get the meaning by the words in it's company.

Skip gram - opposite: Focus word is the single input vector and the target context words are output layer. Good for small amount of training data, good with rare words and phrases (but it takes a LONG time to train, so use CBOW on larger datasets)
Training objective: Minimize summed prediction error of all context words

Document Vectors

Doc2Vec - create a numeric representation of the document
PV-DM - Distributed memory version of paragraph vector

Enxtension to the CBOW model: ad another feature vector. This new feature vector is the topic of the paragraph

PV-DBOW: Distributed BagOfWords version of paragraph vector extension to skip gram

PV-dm is recommended by authors
Use Case : a training set of documents, a word vector W is trained for each word a document vector D is trained for every document, train weights for softmax hidden layer

Inference: given a document, return document vector

Dynamic Word Embeddings - Human language is a construct, things change over time. This is a way to show how words change association over time, think Apple from 1990 vs 2020

Alignment problem, when you learn an embedding for a word in one time window (eg bank) there is no guarantee that embedding will match another time window. Even if there isn't a semantic change: Bank 1990 vs 2008, vs 2021 (bitcoin, mortgage, etc may vary in relation by year)

Temporal Word Embeddings: learn te in all time slices concurrently and apply regularization terms to smooth embedding changes across time.

Count models: weighting scheme of downweight stop words and up weight unique words.
Predict model: word2vec, CBOS: down samples very freq 'a' 'the' et al

Training: models trained on a corpus of 2.8

Believe the Hype, A more realistic expectation: the models are good.

Week 11

Document Clustering is a way of grouping sets of text through unsupervised machine learning
Centroid Based Clustering Model - relating the documents by their closeness to other documents. The distance must be defined: Cosine similarity, jaccard distance, manhattan etc
Connectivity Based Model - relating documents in a hierarchical method through creating a dendrogram.

a) **Agglomerative** (bottom up): Start with the points as individual clusters and, at each step, merge the most similar or closest pair of clusters. This requires a definition of cluster similarity or distance.
b) **Divisive**(top down): Start with one, all-inclusive cluster and, at each step, split a cluster until only singleton clusters of individual points remain. In this case, we need to decide, at each step, which cluster to split and how to perform the split.

Internal quality measure - measure of goodness of a clustering structure without respect to external information.

Cluster Cohesion - measures how closely related are objects in a cluster. Within cluster sum of squares

Cluster Separation - measures how distinct or well separated a cluster is from other clusters. Between cluster sum of squares.

External quality measures - measure the extent to which cluster labels match externally supplied class labels

Entropy - each cluster compute the prob that a member of that cluster belongs to a specific class. $E_j = -\sum_i p_{ij} \log(p_{ij})$,

F-measures - treat each cluster as if it were the result of a query and each class as if it were the desired set of docs for a query

F(i, j) = (2 * Recall(i, j) * Precision(i, j)) / ((Precision(i, j) + Recall(i, j))

Where: Recall(i, j) = nij / ni , Precision(i, j) = nij / nj

Entity Resolution/Document Deduplication - identifying and linking grouping different manifestations of the same real-world object

Week 12

Classification and Semantic Matching

Supervised Learning - A learning algo that classifies new observations. Requires labelled training examples.

Regression - Prediction of a continuous value

Classification - Prediction of a label (level of a class)

Binary Classification - like spam/notspam

Multi-Class - mutually exclusive classification of multiple levels

Multi-Label class - Multiple levels where and observation can have more than one class (like tagging objects in an image.

Naive Bayes - Bayes Law -

- P(x), P(y) = probabilities for event x and y
- P(x,y) = joint probabilities (both events happen)
- P(x|y) = conditional probability (event x happens given that y happens)
- Bayes Law: p(y|x) = (p(x|y) p(y)) / p(x)

Bag of Words - vector of every email with word counts

Conditional Independence - feature prob are independent for a given class

Week 13

Topic Modeling

Topic - a cluster of words that together define a theme. Modeling this way allows us to analyze unlabeled text sets.

- **Canonical Topic Modeling** - match a preestablished list of domain specific topics
- **Organic Topic Modeling** - discover clusters from documents without establishing a list of topics beforehand
- **Entity-centric Topic Modeling** - topics are related to a set of named entities in a domain

Dimensionality reduction - representing text in topic space vs feature space. Topics can be used to improve classification (use topics as features)

Search and recommender systems - search results clustering/grouping search facets, recommending similar items

Uncovering themes in texts - extracting themes and topics from product reviews, comments, news articles, etc

Latent Semantic Analysis (LSA) - topic modeling that uncovers hidden terms which correlate semantically to form topics. Assumption: words that are close in meaning will occur in similar pieces of text

Singular Value Decomposition (SVD) - topic modeling that creates a set of vectors to describe content of text

Latent Dirichlet (LDA) - Generative probabilistic model - each item of a collection modelled as a finite mixture of topic probabilities. Parameters:

- **Alpha**: a dirichlet prior concentration parameter that represents **document-topic density** with high alpha. Documents are assumed to have many topics.
- **Beta**: represent **topic-word density** - with high beta, topics are assumed to contain a mixture of many words (topics contain similar word content)

Non-negative Matrix Factorization (NMF) - decomposes latent relationships in a data matrix, with each document reduced to a set of basis vectors (clusters)

Semantic Topic Modeling - find common topics in queries to get deeper insights. Queries are short-text, sparse and don't split the multiple meanings of a word

Topic Anchors - words most likely to be searched for in the context of queries (these are 'interesting' words in a query.

Word co-occurrence clustering - forms topic anchors, then finds words that co-occur with these.

Weighted-bigrah co-clustering - documents and words are nodes, term frequencies are edges

Week 14

Sentiment analysis (SA) - opinion mining/analysis; computational study of opinions, sentiments and emotions expressed in text.
Facts - objective data in a text document
Opinions - subjective data in a text document
Polarity - positive, negative, or a neutral sentiment
Opinion Holder - the entity in the text or alluded to that owns the opinion

Document-Level SA - classify a document based on the overall sentiment expressed by the opinion holder into a pos/neg class; must assume document only has ONE sentiment from one opinion holder
Sentence-Level SA - has two tasks for each sentence: find subjective vs objective fact. Classify that with a polarity
Feature Based SA - Pulls portions of a text and gets the sentiment of an object. For an ecommerce review it may be features of an item: touchscreen, battery, cost.

SentiWordNet Lexicon - taking each word and getting the sentiment to calculate an overall
Also: Vader and Pattern Lexicons

Week 15 - Finals week