| Term | Definition |
|---|---|
| **Unit 1** | |
| **Natural Language** | Natural languages are those that evolved or emerged gradually over time, largely unconsciously. |
| **Artificial Language** | Artificial languages are those that were designed, crafted, or invented with conscious purpose, largely all at once and not gradually. |
| **NLP** | NLP is “natural language processing (by machine)” |
| **NLU** | Natural Language Understanding – We try to get the machine to produce a useful representation of some inputted natural language. |
| **NLG** | Natural Language Generation – We try to get the machine to produce usable, natural language output that is not just identical to its input. |
| **Unit 2** | |
| **Lexical Analysis** | Dealing with words – what counts as a real word in the language, what is a inflection or a plural versus a singular form of the same noun-- that's lexical analysis. |
| **Syntactic Analysis** | It involves analysis of words in the sentence for grammar and arranging words in a manner that shows the relationship among the words. |
| **Semantic Analysis** | The purpose of this phase is to draw exact meaning from the text. The text is checked for meaningfulness. It is done by mapping syntactic structures and objects in the task domain. |
| **Discourse/entailment Analysis** | The meaning of any sentence depends upon the meaning of the sentence just before it. In addition, it also brings about the meaning of immediately succeeding sentence. |
| **Lexicon** | Machine readable dictionary; the list of stems and affixes, together with basic information about them (whether a stem is a Noun stem or a Verb stem, etc.) |
| **Morphology** | Sort of the arrangement and manipulation of morphemes |
| **Morphemes** | Tiny units that our words are made of. A morpheme is the smallest meaningful unit in a language. |
| **Stemmer** | Piece of software that you feed words into it. And it stems all of them. In other words, it strips off the morphemes that are not the root, like "ing." |
| **Corpus-derived metadata** | The corpus derived metadata includes simplest descriptive metadata along with editorial metadata (providing information about the relationship between corpus components and their original source), analytic metadata (providing information about the way in which corpus components have been interpreted and analyzed), descriptive metadata (providing classificatory information derived from internal or external properties of the corpus components), and administrative metadata (providing documentary information about the corpus). |
| **Collocations** | Words commonly occurring together that sort of take on a different meaning as a unit when they're together |
| **Polysemous** | Words having many meanings |
| **Terminology extraction** | Extraction of key terms from a collection of documents |
| **Lexical diversity measurement** | Token Size / Vocabulary Size |
| **Parts of Speech (POS) Tagging** | The process of classifying words into their parts of speech and labeling them accordingly is known as part-of-speech tagging. Parts of speech are also known as word classes or lexical categories. The collection of tags used for a task is known as a tagset. |
| **Grammar Parser** | A natural language parser is a program that works out the grammatical structure of sentences, for instance, which groups of words go together (as "phrases") and which words are the subject or object of a verb. |
| **Lemmatization** | Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma. |
| **Unitizing Data** | Unitizing just means breaking discrete data points into their units that have previously been jumbled all together as one. |

| Term | Definition |
|---|---|
| **Normalizing Data** | Getting consistent nomenclature and units of measure for how information is conveyed. |
| **Named Entity Extraction / Named Entity Recognition** | It just recognizes entities (people, places, and things), without telling you what type of entity it is, whether a string of words in a sentence is a named entity or not. |
| **Relationship Extraction** | Relationship between 2 or more named entities. |
| **Word Sense Disambiguation** | The problem of resolving semantic ambiguity |
| **Anaphora Resolution** | The process of resolving what a pronoun, or a noun phrase refers to. |
| **Discourse Modeling** | Being able to predict almost like a script what's going to come next, when people are having discourse. Question an |
| **Question Answering** | Adding discourse analysis can make question answering even better. |
| **Textual Entailment** | Drawing logical conclusions from the text |
| **Pragmatic Analysis** | That's the practical reason that allows human beings to make amazing inferences beyond what is strictly explicated in the text that they read or the speech that they hear. |
| **Unit 3** | |
| **Shallow approach** | Or are you going to do something shallow, where you just kind of scrape the surface of all the documents, and you pull out sort of a lightweight representation of every document and do a shallow treatment? |
| **Deep approach** | Are you going to do a deep parse of every sentence? Are you going to do deep semantics and know every nuance of every word sense of every word in every sentence? That would be deep NLP. |
| **Statistical approach** | So, this is where we sort of compute statistics on large amounts of data. To simplify just four sentences in just three terms, three words. Suppose we had three words - dog, cat, and mouse was term one, term two, term three, and we had four sentences. And we just mapped in this three-dimensional space a vector of how many times sentence one had dog, cat, and mouse in it-- term one, two, and three. There's no rules here. We are looking at the statistics and vectors here. |
| **Symbolic approach** | Think of symbolic is just no statistics allowed. So, it's all rule based. We build a knowledge base of rules |
| **Feature Engineering** | Feature engineering is a human being manually defining what all the features are that we're going to use in order to do machine learning on things. So, think of it as like a lexicographer. That's a person who writes a dictionary, handwriting all the dictionary definitions. And if every word is going to be a feature, then you just manually engineered what the features are. |
| **Feature Learning** | And think of feature learning, if you take the human out of it-- take the human out of the system diagram - and you just throw documents at machine learning. And they look at everything. They start combining things and trying to learn what the features are. |
| **Top Down approach** | Start with high-level classifications of texts, and then gradually break down into more and more detail. So, the big categories are what matter to me the most. |
| **Bottom Up approach** | Start by looking at every single word and trying to disambiguate what it means and look at what words count; which words are the most frequent. And only later down the road gets to the big trends and get to summarizing the big trends and the big picture later. |
| **AI approach** | You don't know what's under the hood. It's harder to explain what it's doing |
| **XAI approach** | XAI mean explainable AI. And it just means the transparent approach to AI. And this really tells you something. We got so biased in favor of opaque methods. It's because we favored the statistical over the symbolic, and so on, that it started to be considered exceptional to have your AI be explainable. So that we had to make a new word, XAI, when your AI is something you could actually explain to someone |

| Term | Definition |
|---|---|
| **LSA** | Technique of analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. |
| **LSI** | Indexing and retrieval method that uses a mathematical technique to identify patterns in the relationships between the terms and concepts |
| **Unit 4** | |
| **NLP Software engineer** | Becomes a master of one or a few methods—perhaps ontologies, or semantic parsing, or the ML-related aspects of NLP |
| **Knowledge engineer** | Interfaces regularly with AI engineers and SMEs, demands high “people skills”, Codifies trade secret knowledge of an organization |
| **Data scientist** | You could have the job title data scientist and use a lot of NLP. You could be someone who appreciates NLP, learns how to do it well. |
| **DBA** | DBA who knows how to use NLP to manipulate textual data and turn it more into the format that your database likes to see. |
| **Applied linguistics researcher** | Applied linguistics means taking the results of linguistics research, that can include NLP, and using it in an applied setting. It's a little bit of a loaded term in that if you go to an applied linguistics conference-- it's a huge area. |
| **Cognitive scientist** | It's where you kind of bring together neuroscientist, biologists, psychologists, economists, study human behavior and human decision making-- philosophers, computer scientists, AI people, NLP people. And you look at how thinking works, how language is learned, how decisions are made. And a lot of that is about language. So, an NLP can make a great contribution to that.<br><br>And if you're part of a cognitive science consortium or a project anywhere, most universities, big universities have something going on like this, then you get to intermingle with people in other disciplines. And you become a little bit more of a Leonardo da Vinci, right, the Renaissance man as he was called that kind of knew a lot about a bunch of different disciplines of academic endeavors. |
| **Marketing technologist** | Marketing technology, in most cases, is trying to get the right product or service in front of the right person in the right place at the right time. |
| **Unit 5** | |
| **Sentence Tokenizer / Segmentation** | Treat each sentence as a token. |
| **Word Tokenizer** | Split the text into meaningful words. Take care of punctuation, contractions, etc. |
| **Text Normalization** | Address the Contractions, expansions, stop words, misspellings, stemming, etc. |
| **Content words** | They are an unfinished list to which new words are added. Open class. |
| **Function words** | They tend to not grow. Closed class. |
| **Edit distance method** | How many edits do we make to an apparent misspelt word to turn it into a proper word? |
| **Fuzzy string compare** | Looks for a % of how many characters in common two strings have. And when it's a high percentage, then it assumes that this misspelling might have been intended to be this other word. |
| **Stemming** | Breaking a word apart in the morphemes |
| **Primary Feature** | Requires us to examine the document itself. E.g.: Word Frequencies, Collocations, etc. |
| **Secondary Feature** | Requires us to compare features of the document to those of other documents. E.g.: Differential frequency (TF/IDF), Relative lexical diversity, reading level, etc. |
| **Bigrams** | 2-word pair occurring in a document. |
| **Term Frequency-Inverted Differential Frequency (TF-IDF)** | The number of times a word appears in a document divided by the total number of words in the document. Every document has its own term frequency. |

| Term | Definition |
|---|---|
| | The log of the number of documents divided by the number of documents that contain the word w. Inverse data frequency determines the weight of rare words across all documents in the corpus.<br><br>TF-IDF = tf*idf |
| **Unit 6** | |
| **Lexical knowledge bases** | They are built around a lexicon, and then they go further to create kind of a rich database of how all these words relate to each other. The break words into senses and linking senses to senses via relations. |
| **Hyponym** | A word of more specific meaning than a general or superordinate term applicable to it. For example, spoon is a hyponym of cutlery. |
| **Hypernym** | A word with a broad meaning that more specific words fall under; a superordinate. For example, color is a hypernym of red. |
| **Holonym** | A term that denotes a whole whose part is denoted by another term, such as 'face' in relation to '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. |
| **Simple ontological distance** | It is the method by which we navigate through hypernym/hyponym to move from one object to another. e.g.: the lexical distance between chair and table is 3 (chair → seat → furniture → table). |
| **Monosemy** | The property of having only one meaning. |
| **Polysemy** | The coexistence of many possible meanings for a word or phrase. |
| **Applications of Lexical knowledge bases** | • Enhance usability of search engines<br>• Writing evaluation and advice<br>• Smarter tag clouds |
| **Unit 7** | |
| **POS Tagging** | Also called grammatical tagging is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition and its context. |
| **Unit 8** | |
| **Full Parse Tree** | Parse a sentence into phrases, then could be parsed into other phrases or POS tag. This is continued till we get the POS tag for all the phrases. |
| **Shallow Parse Tree / Light Parser / Chunker** | It is the in-between layer between POS tagging and a full-blown Full Parse Tree. It breaks down sentences into Noun phrase, Verb phrase, Prepositional phrase, etc. |
| **Chinks** | Syntactic elements we leave out of our chunks |
| **IOB** | I – Inside a Chunk<br>O – Outside a Chunk (Chink)<br>B – Beginning of a new Chunk |
| **Full Grammar Parser** | Full analysis of a text, including as detailed a description of its elements as possible. |
| **Constituency Parse** | Breaks a sentence into sub-phrases, sub-sub-phrases, etc. |
| **Dependency Parse** | Provides labeled relations between words. |
| **Unit 10** | |
| **Semantic Similarity** | Semantics may be similar in one or more ways:<br>• Word Similarity<br>• Sense Similarity<br>• Text Similarity<br>• Taxonomy Similarity<br>• Frame Similarity<br>• Context Similarity |
| **Word Similarity** | Statistical approach – how closely associated are the 2 words in a corpus<br>• PPMI (Positive Pointwise Mutual Information)<br>• Vector semantics and LSA (Latent Semantic Analysis) |

| Term | Definition |
|---|---|
| | • Cosine similarity<br>Structural approaches – how close are two words within a semantic graph<br>• Ontological distance<br>• Overlap of parse contexts |
| **Positive Pointwise Mutual Information (PPMI)** | PMI is the "pointwise mutual information" measure, and a positive PMI means the words are related (associated). |
| **Vector Semantics** | Used to judge word similarity as well as text similarity. It represents a distribution of other features found in the same context as each target word. |
| **Latent Semantic Analysis (LSA)** | The smaller number of dense vectors is valuable—it tells which words are most associated by vector semantics without needing huge vectors. |
| **Cosine Similarity for Word Similarity** | Tells us which pair of words are more similar. |
| **Similarity of Parse Contexts** | Two words are similar if they have similar parse contexts |
| **Word Similarity as Ontological Distance** | Number of steps it takes to get from one word to another in a Ontology<br>Shorter Distance ➔ More Similar |
| **Document Similarity** | Measuring the similarity between documents. Common methods are<br>• Jaccard distance<br>• Cosine similarity<br>• Hellinger distance |
| **Jaccard similarity** | Measures how many terms the 2 documents share, compared to the total vocabulary of both documents.<br><br>Does not pay attention to how frequently the word is present in the document. |
| **Cosine similarity for document similarity** | Long sparse vectors are exactly what we have when we generate TF-IDF vectors for documents across a broad vocabulary and it auto-normalizes to document length. |
| **Probability Distributions for document similarity** | It is possible to construe documents as discrete probability distributions. each text is a bag of words, from which we randomly pull out a word. |
| **Hellinger distance** | The Hellinger distance is used to quantify the similarity between 2 probability distributions. |
| **Applications of Semantic Similarity** | • Disambiguating acronyms<br>• Plagiarism detection |
| **Unit 11** | |
| **Document Clustering** | In document clustering, we organize a set of documents into groups having similar characteristics. |
| **Clustering vs. Classification** | Clustering is a very different approach to grouping documents, from classification.<br><br>Clustering is just the documents grouping in a certain way because of similarities between them. The same document may be part of a different cluster based on the parameters that determine the cluster.<br><br>Classification of documents may be based on taxonomy. There is a defined categorization that determines how the documents need to be classified. |
| **Hierarchical Clustering** | AGNES<br>DIANA |
| **Ward's minimum variance** | We decide which cluster to merge based on whatever would produce the smallest increase or within-cluster variance. |
| **Unit 12** | |
| **Types of Text Classification** | • Content-based classification<br>  ○ Metadata-based<br>  ○ Subject-based |

| Term | Definition |
|---|---|
| | • Descriptor-based classification<br>  ○ Taxonomy-based<br>  ○ Query-based |
| **Content-based classification** | Start with 2 of more classes of existing content, where our task is to classify documents into the same categories. |
| **Descriptor-based classification** | Instead of example documents, there is a user-inputted description of the content desired. |
| **Prior probability** | The global distribution of individuals into that predictor |
| **Posterior probability** | The probability of having the predictor attributes |
| **Unit 13** | |
| **Canonical topic modeling** | Match a preestablished list of topics for our domain |
| **Organic topic modeling** | Discover the “natural” topics of a corpus |
| **Entity-centric topic modeling** | Topics are strongly related to sets of NEs that may change over time |
| **Latent Semantic Analysis (LSA)** | LSA-based topic modeling tries to find groups of words associated with the largest variances between documents in the corpus. |
| **Latent Dirichlet Allocation (LDA)** | LDA construes topics as groups of words that have high cooccurrences among different documents in the corpus. |
| **Non-negative Matrix Factorization (NMF)** | We can view NMF as a version of LDA in which the parameters have been tweaked to enforce a sparse number of topics.<br><br>The inherent sparseness of NMF means it’s not the best solution for finding lots of topics in long documents, but it is well suited to handling projects where all the documents are very short. |
| **Topic Coherence** | |
| **Statistical Interference** | |
| **Goal of Canonical Topic Modeling** | The goal of canonical topic modeling is to determine a subset of canonical topics that are materially treated in each corpus, showing which topics are contextually related in that corpus. |
| **Unit 14** | |
| **General Sentiment Scoring** | A simple 2-d detection of general negative or positive sentiment |
| **Two approaches to Sentiment** | • Supervised ML approach<br>• Unsupervised Lexical KB approach |
| **Procedure for Sentiment Analysis with ML** | • Establish a training set<br>• Normalize texts<br>• Extract Feature Vectors<br>• Train a binary classifier<br>• After QA, decide if more training data is needed |
| **Types of Lexical Approach to Sentiment Analysis** | • AFINN<br>• Liu’s Lexicon<br>• MPQA<br>• SentiWordNet<br>• VADER<br>• Pattern library lexicon<br>• Custom |
| **Procedure for Sentiment Analysis with Lexical KB** | • Establish valence-weighted vocabularies.<br>• Normalize texts.<br>• Extract feature vectors.<br>• Execute a scoring algorithm.<br>• After QA, tweak vocabulary and rerun until it passes QA. |