2022

NLP Dictionary

DS7337 NATURAL LANGUAGE PROCESSING

GOODWIN, BEN

Table of Contents

Unit 1 – Introduction to NLP

Natural Language Processing – A subfield of linguistics and computer science, with the objective of studying and understanding how to process interactions of human language by computers.

Natural Language – Languages developed over time, without a purpose in mind
Examples: English, French, Thai, Chinese
Artificial Language – Language developed with a specific purpose
Examples: Python, Klingon
NLP = NLU + NLG

Natural Language Understanding – Subfield within NLP that deals with machine reading comprehension. Considered an AI-hard problem.

Examples: Spell-checking, auto complete, and iPhone word suggestions

Natural Language Generation – A software process that produces natural language output. Generally, the output is text, but there is some debate as to whether the inputs of an NLG system need to be non-linguistic.

Examples: Chat bots

Unit 2 – Levels of Analysis in NLP

Lexical Analysis – This is the process of converting a sequence of characters into a sequence of tokens.

Syntactic Analysis – Tells us that logical meaning of certain given sentences or parts of those sentences.

Example: "The dog went away" "The dog" (noun phrase) "Went away" (Verb phrase)

Semantic Analysis – This is the process of drawing meaning from text. More formally, this is the process of relating syntactic structures from the levels of phrases, clauses, sentences, and paragraphs to the level of the writing to their language independent meanings.

Color is a hypernymy while "blue" "red" "yellow" are color's hyponyms.

Discourse Analysis – An approach to the analysis of written, vocal, or sign language use, or any significant semiotic event. The objects of discourse analysis are variously defined in terms of coherent sequences of sentences.

Example: One may study the purposes and effects of different types of languages

Vocabulary - Number of words in a text or corpus

Vocabulary Size – Number of distinct words in a text

N-Gram – Grouping words around some whole number N

Example: Bigrams contain 2-word phrases, and a 3-gram would be 3-word phrases

Unit 3 – Trade-Offs in NLP

Shallow vs. Deep – At a high level this could refer to the computational resources available as these two terms are related to the amount of computation required for respective activities.

Deep: In the semantic sense this method looks at each word and its associated nuances. In the deep parsing sense, this breaks down all phrases and POS tags them.

Shallow: In the semantic sense this method looks at the document or text at a high level. This method typically breaks documents down at the chunk level.

Statistical vs. Symbolic – These two approaches differ in their approach to NLP. Currently statistical methods are favored, but in the past logical systems have proven effective.

Statistical: This approach is more math based and uses a handful of statistical approaches to perform NLP.

Example: Bayesian inference on words in corpus.

Symbolic: This approach is rule-based that applies criteria to inputs. The rules are clearly listed and explainable to anyone with basic familiarity with logic. The downside of this approach is that rules must be created for any possible situation, and situations can almost always be invented where the rules will fail.

Example: WordNet or POS Tagging

Feature Engineering vs. Feature Learning: In feature engineering, humans are involved in the process, defining, and developing features. Feature learning on the other hand follows the statistical approach where the features are learned based on a variety of statistical approaches.

Top-down vs. bottom up: In a top-down method, we start with a high level of classification and begin the process of slowly breaking down the details. In a bottom-up method, we start with the details and work our way up to the highest level of classification.

Transparent vs Opaque (AI vs. XAI): The difference between these terms relates to Artificial Intelligence versus explainable Artificial Intelligence. Transparent refers to the latter, and opaque, the former. Some AI methods are quite difficult to explain and get bucketed into “black boxes.”

Unit 4 – Working in NLP

NLP and Data Science: Natural Language processing is one of the most popular subfields in data science and is essential in progressing the field forward. Natural Language Processing is important because it helps to resolve ambiguity in language and adds useful numeric data for many downstream applications. Think of examples like chat bots or machine translation.

Job roles that utilize NLP: There is a laundry list of jobs that involve Natural Language processing some of those include: Email filters, Smart assistants (think Apple’s Siri, Amazon’s Alexa, or Google’s Assistant), predictive text, language translation, and text analytics.

Sectors that utilize NLP: Many different sectors utilize Natural Language Processing like healthcare, manufacturing, translation, and customer service.

Organizations that relate to NLP: Many businesses and organizations rely on NLP and may not be aware of it. Among these organizations: Google, InData Labs, Amazon, and IBM

Unit 5 – Low-Level Analysis

Text preprocessing: At a high-level text preprocessing in NLP is a method to clean the text data and make it ready to feed data to the model. This is a hugely important step in NLP as data in the wild is noisy and difficult for a computer to process. NLTK (Natural Language Tool Kit) can be used in this step for tasks like removing stop words, NEI (Named entity recognition), POS tagging (Part of speech tagging), and a handful of other useful tasks.

TF-IDF: Term frequency – inverse document frequency. This is a method of differential frequency analysis. This method looks at the frequency of the term in question and divides that by the phrases in the total document and determines its overall importance in each document.

Relative lexical diversity: Formula: Number of unique tokens/total number of tokens

Reading level: Examines syllable count, sentence length, and words relative to a compiled list.

Text Normalization: At a high-level text normalization is an attempt to reduce the randomness of the text. The goal is to reduce the amount of different information the computer must deal with. There are many details contained within this topic.

Stop words: A list of common words that serve no point in distinguishing between texts. These words are commonly removed and often are “a, an, the, on, of.”

Content and function words: Function words are best defined as words that do not contribute to the understanding of the text but serve purpose to the reader by defining answers to the subject. Content words are useful for determining meaning from a text.

Stemming: A stemmer reduces each word to its stem

Example: Running -> Run

Note: Stemmers don’t always produce a grammatically correct stem

Low-level document feature extraction: This is defined as the process of removing the smallest level of detail from a document such as stems.

Unit 6 – Lexical Knowledge Bases

Lexical knowledge bases: A repository of computational information about concepts intended to be generally useful in many application areas including computational linguistics (NLP). It contains information derived from machine-readable dictionaries, reference books, and results of statistical analyses of text usages. They’re to serve as a general repository of knowledge about lexical concepts.

Hyponym: These are generally subdivisions of more general words.

Example: Boston Terrier. Boston Terrier is a specific breed of dog.

Hypernym: A word meaning that encapsulates the meanings of other words. These are general words.

Example: Dog is more general towards an animal.

Meronym: A word that denotes a constituent part or a member of something.

Example: Peach tree

Resources for creating or extending lexical knowledge bases: There are a handful of extremely powerful lexical knowledge bases out there. IEEE contains many published articles on creating and extending lexical knowledge bases.

Applications of lexical knowledge bases: NLP is heavily reliant on these artifacts; they encode human knowledge of language in machine readable form.

WordNet: A series of lexical databases derived from the original Princeton WordNet which groups words into synonym sets and connects them using lexical and conceptual-semantic relations.

Unit 7 – Syntactic Analysis: POS-Tagging

POS-tagging: This task looks at text and determines the part of speech associated with it. A typical POS tagger will identify the different parts of a text or corpus and tag them (noun, adjective, verb). To implement POS tagging, investigate something like WordNet (Unit 6).

Users can also utilize the Penn Treebank which is defined at the bottom of this document.

Using POS tags: The Penn Treebank has greater than 90% accuracy, and most human experts only agree about 96% of the time, so this is still an unsolved problem. How can we teach computers if we don't even agree?

Unit 8 – Syntactic Analysis: Parsing

Parsing – The process of analyzing a string of symbols, either in natural language, computer languages, or data structures, conforming to the rules of grammar.

Shallow parsing: Shallow parsing in an analysis of a sentence which first identifies constituent parts of sentences and then links them to higher order units that have discrete grammatical meanings.

Chunks: Chunking is defined as the process of natural language processing used to identify parts of speech and short phrases present in each sentence. Chunks are what we're after in NLP (ex. Adjectives or noun phrases), whereas chinks are the unimportant parts.

Full grammar parsing: This is considered a complex and computationally intensive task to develop a full grammar tree.

Example: Useful for machine translation

Uses for full parse trees: Useful for machine translation

Unit 10 – Semantic Analysis: Semantic Relatedness

Word Similarity – At a very basic level, how similar are two words?
Text similarity is to calculate how two words/phrases/documents are close to each other. That closeness may be lexical or in meaning

Document Similarity – This is a big one, at a very high-level document similarity is exactly that, a means for comparing "how similar are two documents." Below we will examine different means to achieve this.

Cosine similarity is a metric used to measure how similar the documents are irrespective of their size. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space.

The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance (due to the size of the document), chances are they may still be oriented closer together. The smaller the angle, higher the cosine similarity.

Source: https://www.machinelearningplus.com/nlp/cosine-similarity/

Applications of Semantic Similarity – There are a multitude of semantic similarity measures a handful of the applications are as follows:

Text classification
Information Retrieval
Topic Tracking
Document Clustering
Machine Translation

Unit 11 – Semantic Analysis: Document Clustering

Document Clustering – Grouping together of similar data, usually obtained by calculating the distance between the points. No labels given in advance, and we don't know what the labels should be in advance.

Methods of clustering: Two high level methods of clustering: Centroid-based or hierarchical-based.

Centroid based: Trying to find the most representative document of one cluster versus a very different document that's considered most representative. The prototypical or paradigmatic document that represents all the other documents will be in that cluster. The focus is around one central item, and a handful of documents that are like that centroid document. We can spread this across multiple clusters, and then we should have some sensible looking clusters. Within centroid based is k-means.

Hierarchical based: This method is more akin to classifying, despite it not actually classifying. The technique here is to determine how to break all the documents into a handful of group and then take each of those groups and then repeat the process over again until we have built a hierarchy. This is considered a top-down approach, and there's also a bottom-up approach to doing hierarchical as well.

Within hierarchical clustering is Agnes and Diana.

Working with clusters:

> K-means: This is the most popular means of centroid clustering and its called k-means. The centroid doesn't have to be a particular document, it could be a mean of average set of feature values, it can even be a virtual theoretical document that represents what the center of that would be. We take the mean of all the features, and we have k (which is just tradition). The letter k defines the number of centroids. The centroid may not actually be in the center, this is because no document, but an imaginary document would have exactly the average term frequency of TFIDF values of all the documents assigned to that centroid. K-means is the classic centroid clustering algorithm.
>
> Agnes: You start with one big cluster; every document is an individual cluster and then documents start getting merged. This process continues, until maybe all the documents merged into one big super cluster. This process results in a hierarchical diagram. Also known as "agglomerative clustering" and works alongside with Ward's minimum variance.
>
> Diana: Essentially the Agnes process in reverse, starting with one big cluster of all the documents in one cluster, and then we try to find the farthest outlier- the document that looks least like the others, and this will become the subcluster. After this, we are left with two clusters, and then we take these two clusters and repeat the process making the most unusual document representative of a new cluster, and the reassign documents that are more like it. Like Agnes, if we keep dividing, we will be left with individual documents. We can halt this process when we are content with the number of clusters.
> As a final thought, most of the time that people are actually running Agnes or actually running Diana, they are using one of these early stop decision logics on it in actual practice.

Unit 12 – Semantic Analysis: Text Classification

Document Classification – There are two main types of document classification or text classification, they are content-based classification and descriptor-based classification.

Content-based classification is what is performed most of the time, beginning with two or more classes, usually more, but at least two. Thinking of spam and ham email.

Descriptor-based classification typically includes a written description of what the two or more different classes of content are that are desired. Typically, it is a request of what someone is after, no example documents yet. A good example of descriptor-based classification would be legal discovery or FOIA requests.

SVMs for Text Classification – In the field of text classification, a SVM classifier is pretty much a binary classifier. In the case where we desire a multiclass solution, there's a solution to stitch together a whole bunch of binary SVM classifiers.

Example: We need to classify documents as entrée recipes versus dessert recipes. In the ideal solution space is that if we were to graph all these data based on features, the weighted term frequencies. Another way to think of these of there being. Many possible ways to draw a line in two-dimensional space like this to separate everything that's on the graph. Some lines do a better job of separating the space, but what SVM does is that should we imagine we tried every which way there could be of dividing all the dots with a line until we found one that's sort of maximized the separations. With every conceivable line, there is one that has the maximum margin possible.

With this line, the features of the items on either side of the line are going to be the features that would predict when a new document comes in, we can identify where it belongs. SVM is extremely popular for document classification.

Unit 13 – Semantic Analysis: Topic Modeling

Topic Model Paradigms:

Canonical: Based on the word canon and a canon is a body of official literature of some kind. When something is canonical, it means that there is an authoritative source that is handed down from on high. Unofficial set of documents or an official set of topics or an official list of words or an official set of rules or an official story, namely anything like that being handed down from above. You are obligated to use what they gave you, meaning "you've been given a canonical source, per chance, of topics."

Example: Index literature into the library of congress

Organic: Organic topic modeling is almost the opposite of canonical, there is no canon whatsoever, the ask is to discover the natural or "organic" topics of a corpus of documents. What just comes naturally in these documents as the predominating topics of discourse throughout all the documents. It boils down to "tell us what the topics are."

Example: Determine the topics from a large collection of documents.

Entity-Centric: Often, the topics that are desired are related to a set of named entities (people). This may change over time, but the set is always known, as it is an established list of named entities, and the ask is requesting you to find all the topics that are related to this list of names.

Example: Index the player roster of every team in Major League Baseball.

Unit 14 – Semantic Analysis: Sentiment and Rhetoric

General Sentiment Analysis: At a high level sentiment analysis is considered a kind of sematic analysis because sentiment can involve either the feeling, the emotion, or just the opinion, judgement, evaluation that goes along with language. Advanced Sentiment Analysis takes this further and can include scoring metrics and dig into more of the specifics of the question we are after.

General sentiment scoring exists and is considered positive or negative sentiment analysis, the limitation here is that it doesn't tell you how it's negative or how it's positive, for example both angry and sad are both negative, but general sentiment scoring claims, "negative is negative." The scoring aspect comes into play with the scoring metric of "polarity" and this is scored from negative one to one, for example 0.3 would be a little bit negative, and negative 0.9 would be more intensely negative or a greater amount of negative sentiment in a text.

Sometimes there are systems where the sentiment scores can go above a one, sometimes without a limit, reaching negative infinity, and positive infinity.

Typically, a zero represents that the polarity is neutral, however sometimes it can represent a few different things:

It can mean that the system only knows how to recognize a certain set of negative and positive things and thus zero means non-detection of polarity. A key piece being that it doesn't necessarily notice words that are neutral.

Sometimes zero is the net score when adding together the negative and positive
Main idea: make sure you understand the scoring if you are using someone else's system.

There are a few approaches for general sentiment analysis.
Supervised Machine Learning Approach: Uses SDG (an SVM) implementation, as a binary classifier, give it training data, and then it will train in SVM to build a model.

Pro: Quickly build a sentiment analyzer if a large amount of training data is available.

Con: Opacity versus transparency in AI, this approach uses thousands and thousands of cases, so if someone asks why a particular comment is negative sentiment, an answer is difficult to come by.

Unsupervised Approach with a built lexical knowledgebase: This process is going to start with the decision on which lexicon to use (sentiment lexicon). The lexicon in this case will simply have a smaller vocabulary. (Typically, smaller than the English language) with just the words that have a clear positive or negative connotation. This method relies on "clues" where we put a bunch of these clues together and try to measure the overall sentiment is of an entire sentence, an entire paragraph or an entire document, and the clues will eventually tell us if this person's being more negative in their expression or positive.

Aside: There have been a handful clue-based lexicons with a varying number of clues (AFINN (affective lexicon by Finn Nielsen), and Ben Liu's with over 6,800 clues)) WordNet has also been attempted to be labeled but was mainly machine generated.

Vader also exits (The Valence Aware) The weights from positive and negative, they are called valences, valence Aware Dictionary for Sentiment Reasoning.

An alternative approach also exists that you have the option to create a custom lexicon, where you make a lexicon from scratch, this approach allows you to create as many clues as you feel are necessary.

This is sometimes considered the best approach, because each domain is different, for example if your project is about movie reviews, there's vocabulary in there that contains positive, negative, and neutral sentiment for your domain.

Example: "The actor is one-dimensional" we know that this is an insult, and we also know that one-dimensional may not be in any of these vocabularies because only if we talk about fiction, fictional character in literature.

Personal example: In the homework where we to take movie review data and analyze the sentiment, I chose the movie "Austin Powers" as one of my movies, and one of the main characters is named, "Dr. Evil" and after running a stemmer on the data, the word "evil" remained and this word without context is typically a word with negative connotations. Unfortunately, in the case of the movie "Austin Powers" Dr.Evil is a main character and is actually pretty funny, the movie is a comedy after all. This is an important case where a custom lexicon would have been useful for this project, I could have personally defined the lexicon to indicate that evil is probably associated with the name of a character and doesn't typically represent the sentiment of the movie reviewers toward the movie.

Pros and cons of the lexical based approach:

Pros: Opaque versus transparent. In this case transparency wins, this method is quite explainable. In the example where someone asks why a theoretical tweet analysis system labeled a tweet as negative and we now have a clear means to explain the "why." We can simply point to the word and show that it identifies as positive or negative or neutral. This can be a huge benefit, as it can easily be explained in layman's terms.

Cons: A huge con of this approach is that we need very specific vocabulary, and not just any vocabulary. We need a list of all the negative words, and it also needs to be coded. An example of this would be, that we don't want to think that the words "painful" and "excruciating" are just both negative. We need to know that "excruciating" is the really intense version of "painful" the former needs to have more negative weight to it.

Additionally, this is a considerable amount of work, and an entire lexicon will have to be built up. The creators will also have to main the lexicon going forward as new tropes come in.

Building a lexicon also requires awareness of the meaning of all the words, and sometimes words that don't necessarily exist in pre-built lexicons.

Finally, how do we choose between the two options?

The answer as expected is that it depends. Better results will often come from the hand-built lexical approach, but we can run into plenty of problems with this approach, such as maintenance and finding the labor necessary to build up such a tool.

In the laptop review example, if we're asked to build up a sentiment lexicon for consumer reviews of laptops, we will want to gather a bunch of those reviews and examine what people say negatively and positively.

We will then put them into a list and then assign sentiment, the big idea here is that the human plays a large role.

We can also take the sentiment analysis further; we can ask questions like determining the referent of what the sentiment in a sentence is about or the sentiment in a whole document. We can also inquire about the specifics of positive or negative sentiment toward what? We can determine what object that attaches to.

To answer some of these questions we can use some of the following approaches:

Hierarchical Sentiment Scoring: Where we build a taxonomy of what the different kinds of sentiment are, we could break down positive and negative into their individual components.

Hybrid Approach – Semiautomated Feature Engineering for Sentiment Lexicography: This method breaks the barrier and combined both the statistical approach and the rule-based approach. This method does its best to give us the best of both worlds. In this realm where we have automated statistical approaches where we can bootstrap the process that a lexicographer must go through to create a lexicon, this is called:

Differential Frequency Analysis: This approach looks at some term frequencies contrasted with some term frequencies over there. For sentiment analysis, it leverages positive sentiment versus negative sentiment. We will also give candidate clues, because they are sort of singals of some possible sentiment.

Penn Treebank Parts of Speech Tags

1) CC Coordinating conjunction – joins phrases
    (1) Ex. and, but, or
2) CD Cardinal number – Numbers spelled out
    (1) Ex. one, two, three, etc.
3) DT Determiner – Decides or selects something

(1) Ex. a, the, every

4) EX Existential *there* – used to mark the existence of something.

(1) Ex. There were many people at someplace

5) FW Foreign word - Word not in OED and some abbreviations

(1) Ex. E.g., etc., i.e., foreign language words Latin, urban dictionary, etc…

6) IN Preposition or subordinating conjunction - A preposition usually indicates the temporal, spatial or logical relationship of its object to the rest of the sentence

(1) Ex. Onto, since, above, along, and many more

7) JJ Adjective – Describes a noun

(1) Red, large, old, good

8) JJR Adjective, comparative – an adjective which denotes some sort of comparison

(1) Ex. More red, Larger, older, better

9) JJS Adjective, superlative – the "most" and the "-est"

(1) Ex. Largest, Oldest, most red, best

10) LS List item marker –

(1) Ex. These lists, 1., a., bullet points

11) MD Modal – indicate present operative and normative states

(1) Ex. Shall Should, can could, may might, etc.

12) NN Noun, singular or mass – common person, place, thing, idea

(1) Ex. Girl, car, tree

13) NNS Noun, plural – plural forms of NN

(1) Ex. Girls, Cars, Trees

14) NNP Proper noun, singular – Specific names of NN

(1) Jane, Thunderbird, Yggdrasil

15) NNPS Proper noun, plural – plural NNP

(1) Cardinals (Team), Smiths (refers to lots of Smith)

16) PDT Predeterminer – Central, pre and Post determiners indicate things about the determiners

(1) Ex. My, this, all, both, many, several

17) POS Possessive ending - Indicate ownership

(1) Ex. He's, boy's, Mary's

18) PRP Personal pronoun – self references

(1) Ex. I, me, we, us, our

19) PRP$ Possessive pronoun – possessive versions of PRP

(1) Ex. Mine, ours, yours

20) RB Adverb – modifies verbs like adjectives with nouns

(1) Ex. Slowly, never, rarely,

21) RBR Adverb, comparative – indicates some intensity order to the verbs

(1) More slowly, more happily,

22) RBS Adverb, superlative – the most

(1) Ex. Most slowly, most happily

23) RP Particle – unaccompanied prepositions

(1) Ex. Along, away, off

24) SYM Symbol – non-alpha characters

(1) Ex. &, -,

25) TO *to* – self explanatory
26) UH Interjection - a word used to convey emotion
    (1) Ex. Congratulations, peace, thanks, whoa
27) VB Verb, base form – action words
    (1) Ex. Run, swim, drive
28) VBD Verb, past tense – actions that happened in the past
    (1) Ex. Ran, swam, drove
29) VBG Verb, gerund, or present participle – indicate present states
    (1) Ex. Running, driving, dying
30) VBN Verb, past participle – past states of being
    (1) Ex. Studied, been living
31) VBP Verb, non-3rd person singular present – same as verb but convey habits, routines, thoughts
    (1) Ex. Same as verb forms run, swim, etc.
32) VBZ Verb, 3rd person singular present
    (1) Ex. Argues, Applies, tries
33) WDT Wh-determiner – determines selections not at beginning of sentence
    (1) Which, what, whatever – I don't know **what** you are talking about
34) WP Wh-pronoun – same as above but at beginning of sentence
    (1) **Which** bowl of soup
35) WP$ Possessive wh-pronoun – same as above but possessive
    (1) Whose, whom
36) WRB Wh-adverb – the events
    (1) When, where, why, how