Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

Introduction to Natural Language Processing With Python

Ms.Warsha M.Choudhari[1] , Ms.Rinku Rajankar[2]
[1]*Professor, Information Technology, Datta Meghe Institute of Engineering, Technology & Research, Wardha, India*
[2]*Professor, Computer Science & Engineering,ITM College of Engineering, Nagpur, India*

Abstract—Natural Language Processing holds great promise for making computer interfaces that are easier to use for people, since people will be able to talk to the computer in their own language, rather than learn a specialized language of computer commands. For programming, however, the necessity of a formal programming language for communicating with a computer has always been taken for granted.

I. INTRODUCTION

The term Natural Language Processing encompasses a broad set of techniques for automated generation, manipulation and analysis of natural or human languages. Although most NLP techniques inherit largely from Linguistics and Artificial Intelligence, they are also influenced by relatively newer areas such as Machine Learning, Computational Statistics and Cognitive Science.

Token: Before any real processing can be done on the input text, it needs to be segmented into linguistic units such as words, punctuation, and numbers or alphanumeric. These units are known as tokens.

Sentence: An ordered sequence of tokens.

Tokenization: The process of splitting a sentence into its constituent tokens. For segmented languages such as English, the existence of whitespace makes tokenization relatively easier and uninteresting.

Corpus: A body of text, usually containing a large number of sentences.

Part-of-speech (POS) Tag: A word can be classified into one or more of a set of lexical or part-of-speech categories such as Nouns, Verbs, Adjectives and Articles, to name a few. A POS tag is a symbol representing such a lexical category - NN (Noun), VB (Verb), JJ (Adjective), AT (Article).

Parse Tree: A tree defined over a given sentence that represents the syntactic structure of the sentence as defined by a formal grammar.

II. BASIC TERMINOLOGY

POS Tagging: Given a sentence and a set of POS tags, a common language processing task is to automatically assign POS tags to each word in the sentences. For example, given the sentence The ball is red, the output of a POS tagger would be The/AT ball/NN is/VB red/JJ.

Computational Morphology: Natural languages consist of a very large number of words that are built upon basic building blocks known as morphemes (or stems), the smallest linguistic units possessing meaning. Computational morphology is concerned with the discovery and analysis of the internal structure of words using computers.

Parsing: In the parsing task, a parser constructs the parse tree given a sentence. Some parsers assume the existence of a set of grammar rules in order to parse but recent parsers are smart enough to deduce the parse trees directly from the given data using complex statistical models. Most parsers also operate in a supervised setting and require the sentence to be POS-tagged before it can be parsed.

Machine Translation (MT): In machine translation, the goal is to have the computer translate the given text in one natural language to fluent text in another language without any human in the loop.

III. LEVELS OF NATURAL LANGUAGE PROCESSING

The most explanatory method for presenting what actually happens within a Natural Language Processing system is by means of the 'levels of language' approach. This is also referred to as the synchronic model of language and is distinguished from the earlier sequential model, which hypothesizes that the levels of human language processing follow one another in a strictly sequential manner. Psycholinguistic research suggests that language processing is much more dynamic, as the levels can interact in a variety of orders. Introspection reveals that we frequently use information we gain from what is typically thought of as a higher level of processing to assist in a lower level of analysis. For example, the pragmatic knowledge that the document you are reading is about biology will be used when a particular word that has several possible senses (or meanings) is encountered, and the word will be interpreted as having the biology sense of necessity, the following description of levels will be presented sequentially.

The key point here is that meaning is conveyed by each and every level of language and that since humans have been shown to use all levels of language to gain understanding, the more capable an NLP system is, the more levels of language it will utilize.

Phonology: This level deals with the interpretation of speech sounds within and across words.

There are, in fact, three types of rules used in phonological analysis:

1) Phonetic rules – for sounds within words;

2) Phonemic rules – for variations of pronunciation when words are spoken together, and;

3) Prosodic rules – for fluctuation in stress and intonation across a sentence.

In an NLP system that accepts spoken input, the sound waves are analyzed and encoded into a digitized signal for interpretation by various rules or by comparison to the particular language model being utilized.

Morphology: This level deals with the componential nature of words, which are composed of morphemes – the smallest units of meaning. For example, the word preregistration can be morphologically analyzed into three separate morphemes: the prefix pre, the root registration, and the suffix. Since the meaning of each morpheme remains the same across words, humans can break down an unknown word into its constituent morphemes in order to understand its meaning. Similarly, an NLP system can recognize the meaning conveyed by each morpheme in order to gain and represent meaning. For example, adding the suffix –ed to a verb, conveys that the action of the verb took place in the past.

Lexical: At this level, humans, as well as NLP systems, interpret the meaning of individual words. Several types of processing contribute to word-level understanding – the first of these being assignment of a single part-of-speech tag to each word. In this processing, words that can function as more than one part-of-speech are assigned the most probable part-of-speech tag based on the context in which they occur. Additionally at the lexical level, those words that have only one possible sense or meaning can be replaced by a semantic representation of that meaning. The nature of the representation varies according to the semantic theory utilized in the NLP system.

The lexical level may require a lexicon, and the particular approach taken by an NLP system will determine whether a lexicon will be utilized, as well as the nature and extent of information that is encoded in the lexicon. Lexicons may be quite simple, with only the words and their part(s)-of-speech, or may be increasingly complex and contain information on the semantic class of the word, what arguments it takes, and the semantic limitations on these arguments, definitions of the sense(s) in the semantic representation utilized in the particular system, and even the semantic field in which each sense of a polysemous word is used.

Syntactic: This level focuses on analyzing the words in a sentence so as to uncover the grammatical structure of the sentence. This requires both a grammar and a parser. The output of this level of processing is a representation of the sentence that reveals the structural dependency relationships between the words.

Syntax conveys meaning in most languages because order and dependency contribute to meaning. For example the two sentences: 'The dog chased the cat.' and 'The cat chased the dog.' differ only in terms of syntax, yet convey quite different meanings.

Semantic: Semantic processing determines the possible meanings of a sentence by focusing on the interactions among word-level meanings in the sentence. This level of processing can include the semantic disambiguation of words with multiple senses; in an analogous way to how syntactic disambiguation of words that can function as multiple parts-of-speech is accomplished at the syntactic level. Semantic disambiguation permits one and only one sense of polysemous words to be selected and included in the semantic representation of the sentence. For example, amongst other meanings, 'file' as a noun can mean either a folder for storing papers, or a tool to shape one's fingernails, or a line of individuals in a queue. If information from the rest of the sentence were required for the disambiguation, the semantic, not the lexical level, would do the disambiguation. A wide range of methods can be implemented to accomplish the disambiguation, some which require information as to the frequency with which each sense occurs in a particular corpus of interest, or in general usage, some which require consideration of the local context, and others which utilize pragmatic knowledge of the domain of the document.

Discourse while syntax and semantics work with sentence-length units, the discourse level of NLP works with units of text longer than a sentence. That is, it does not interpret multisentence texts as just concatenated sentences, each of which can be interpreted singly. Rather, discourse focuses on the properties of the text as a whole that convey meaning by making connections between component sentences. Several types of discourse processing can occur at this level, two of the most common being anaphora resolution and discourse/text structure recognition. Anaphora resolution is the replacing of words such as pronouns, which are semantically vacant, with the appropriate entity to which they refer.

Discourse/text structure recognition determines the functions of sentences in the text, which, in turn, adds to the meaningful representation of the text. For example, newspaper articles can be deconstructed into discourse components such as: Lead, Main Story, Previous Events, Evaluation, Attributed Quotes, and Expectation.

Pragmatic: This level is concerned with the purposeful use of language in situations and utilizes context over and above the contents of the text for understanding The goal is to explain how extra meaning is read into texts without actually being encoded in them. This requires much world knowledge, including the understanding of intentions, plans, and goals. Some NLP applications may utilize knowledge bases and inference modules.

IV. NATURAL LANGUAGE PROCESSING APPLICATIONS

• **Information Retrieval** – given the significant presence of text in this application, it is surprising that so few implementations utilize NLP.

• **Information Extraction (IE)** – a more recent application area, IE focuses on the recognition, tagging, and extraction into a structured representation, certain key elements of information, e.g. persons, companies, locations, organizations, from large collections of text. These extractions can then be utilized for a range of applications including question-answering, visualization, and data mining.

• **Question-Answering** – in contrast to Information Retrieval, which provides a list of potentially relevant documents in response to a user's query, question-answering provides the user with either just the text of the answer itself or answer-providing passages.

• **Summarization** – the higher levels of NLP, particularly the discourse level, can empower an implementation that reduces a larger text into a shorter, yet richly constituted abbreviated narrative representation of the original document.

• **Machine Translation** – perhaps the oldest of all NLP applications, various levels of NLP have been utilized in MT systems, ranging from the 'word-based' approach to applications that include higher levels of analysis.

• **Dialogue Systems** – perhaps the omnipresent application of the future, in the systems envisioned by large providers of end-user applications. Dialogue systems, which usually focus on a narrowly defined application (e.g. your refrigerator or home sound system), currently utilize the phonetic and lexical levels of language.

V. PYTHON

The Python programming language is a dynamically-typed, object-oriented interpreted language. Although, its primary strength lies in the ease with which it allows a programmer to rapidly prototype a project, its powerful and mature set of standard libraries make it a great fit for large-scale production-level software engineering projects as well. Python has a very
shallow learning curve and an excellent online learning resource [1].

Python's feature highlights include:

Easy-to-learn: Python has relatively few keywords, simple structure, and a clearly defined syntax.
Easy-to-read: Python code is much more clearly defined and visible to the eyes.
Easy-to-maintain: Python's success is that its source code is fairly easy-to-maintain.
A broad standard library: One of Python's greatest strengths is the bulk of the library is very portable and cross-platform compatible on UNIX, Windows and Macintosh
Interactive Mode: Support for an interactive mode in which you can enter results from a terminal right to the language, allowing interactive testing and debugging of snippets of code.
Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
Extendable: You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
Databases: Python provides interfaces to all major commercial databases.
GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh and the X Window system of UNIX.
Scalable: Python provides a better structure and support for large programs than shell scripting.

Python has a big list of good features, few are listed below:

- Support for functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- Very high-level dynamic data types and supports dynamic type checking.
- Supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA and Java.

VI. CONCLUSION

NLP allows a coherent study of the human language from the vantage points of several disciplines - Linguistics, Psychology, Computer Science and Mathematics.

Python allow any programmer to get acquainted with NLP tasks easily without having to spend too much time on gathering resources.

VII. REFERENCES

[1] The Official Python Tutorial. http://docs.python.org/tut/tut.html
[2] Dan Bikel. 2004. On the Parameter Space of Generative Lexicalized Statistical Parsing Models. Ph.D. Thesis. http://www.cis.upenn.edu/~dbikel/papers/thesis.pdf
[3] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In International Conference on Machine Learning (ICML), 2009.

