# Natural Language Processing (NLP)

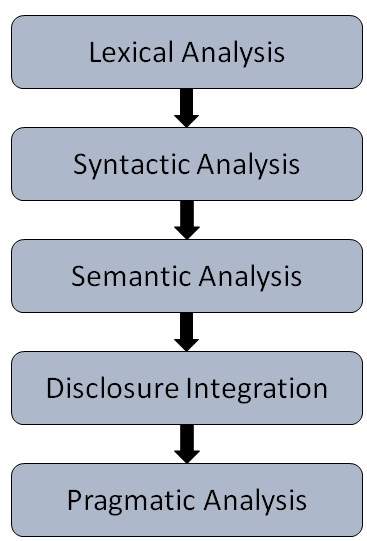
## What is NLP ?

Natural Language Processing (NLP) refers to AI method of communicating with an intelligent system using a natural language such as English.

Processing of Natural Language is required when you want an intelligent system like robot to perform as per your instructions, when you want to hear decision from a dialogue based clinical expert system, etc.

## Steps in NLP

* **Lexical Analysis** − It involves identifying and analyzing the structure of words. Lexicon of a language means the collection of words and phrases in a language. Lexical analysis is dividing the whole chunk of txt into paragraphs, sentences, and words.
* **Syntactic Analysis (Parsing)** − It involves analysis of words in the sentence for grammar and arranging words in a manner that shows the relationship among the words. The sentence such as “The school goes to boy” is rejected by English syntactic analyzer.



* **Semantic Analysis** − It draws the exact meaning or the dictionary meaning from the text. The text is checked for meaningfulness. It is done by mapping syntactic structures and objects in the task domain. The semantic analyzer disregards sentence such as “hot ice-cream”.
* **Discourse Integration** − 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.
* **Pragmatic Analysis** − During this, what was said is re-interpreted on what it actually meant. It involves deriving those aspects of language which require real world knowledge.

## Five open source nlp tools

1. [Stanford's Core NLP Suite](http://nlp.stanford.edu/software/corenlp.shtml) : A GPL-licensed framework of tools for processing English, Chinese, and Spanish. Includes tools for tokenization (splitting of text into words), part of speech tagging, grammar parsing (identifying things like noun and verb phrases), named entity recognition, and more. Once you've got the basics, be sure to check out the [other projects](http://nlp.stanford.edu/software/) from the same group at Stanford.
2. [Natural Language Toolkit](http://www.nltk.org/) :  If your language of choice is Python, then look no further than NLTK for many of your NLP needs. Like the Stanford library, it includes capabilities for tokenizing, parsing, and identifying named entities as well as many more features.
3. [Apache Lucene and Solr](http://lucene.apache.org/)  : While not technically targeted at solving NLP problems, Lucene and Solr contain a powerful number of tools for working with text ranging from advanced string manipulation utilities to powerful and flexible tokenization libraries to blazing fast libraries for working with finite state automatons. On top of it all, you get a search engine for free!
4. [Apache Open-NLP](http://opennlp.apache.org/) :  Using a different underlying approach than Stanford's library, the Open-NLP project is an Apache-licensed suite of tools to do tasks like tokenization, part of speech tagging, parsing, and named entity recognition. While not necessarily state of the art anymore in its approach, it remains a solid choice that is easy to get up and running.
5. [GATE](https://gate.ac.uk/)**and**[Apache UIMA](https://uima.apache.org/)  : As your processing capabilities evolve, you may find yourself building complex NLP workflows which need to integrate several different processing steps. In these cases, you may want to work with a framework like GATE or UIMA that standardizes and abstracts much of the repetitive work that goes into building a complex NLP application.

## Steps for NLP

### Tokenize and tag some text:

**>>> import** **nltk**

**>>>** sentence = """At eight o'clock on Thursday morning

**...** Arthur didn't feel very good."""

**>>>** tokens = nltk.word\_tokenize(sentence)

**>>>** tokens

['At', 'eight', "o'clock", 'on', 'Thursday', 'morning',

'Arthur', 'did', "n't", 'feel', 'very', 'good', '.']

**>>>** tagged = nltk.pos\_tag(tokens)

**>>>** tagged[0:6]

[('At', 'IN'), ('eight', 'CD'), ("o'clock", 'JJ'), ('on', 'IN'),

('Thursday', 'NNP'), ('morning', 'NN')]

### Identify named entities:

**>>>** entities = nltk.chunk.ne\_chunk(tagged)

**>>>** entities

Tree('S', [('At', 'IN'), ('eight', 'CD'), ("o'clock", 'JJ'),

('on', 'IN'), ('Thursday', 'NNP'), ('morning', 'NN'),

Tree('PERSON', [('Arthur', 'NNP')]),

('did', 'VBD'), ("n't", 'RB'), ('feel', 'VB'),

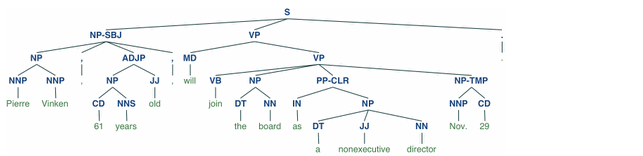
('very', 'RB'), ('good', 'JJ'), ('.', '.')])

### Display a parse tree:

**>>> from** **nltk.corpus** **import** treebank

**>>>** t = treebank.parsed\_sents('wsj\_0001.mrg')[0]

**>>>** t.draw()



## Does a Bot Need Natural Language Processing?

One of the most exciting things about the rise of chatbots is their use of artificial intelligence — especially machine learning — to mass-accomplish tasks that neither an army of interns nor an army of experts could match, and to derive wisdom beyond that of the crowd by analyzing the crowd’s billions of conversations with cold math. Yet anyone who chats with a few bots knows the frequent frustration: *This thing doesn’t understand what I’m saying.*

There are basically two kinds of chatbots in early 2017, while natural language processing is still learning to understand human conversational speech: [Bots that risk trying to parse anything you type at them, and bots that limit your input to a few safe option buttons](https://chatbotsmagazine.com/which-is-best-for-you-rule-based-bots-or-ai-bots-298b9106c81d). Octane AI, which publishes Chatbots Magazine, currently opts for the button approach. But of course we wonder ourselves: Is that doomed?

Bot developers will tell you that it depends what your bot is trying to accomplish. In theory, a bot with a human adult level of linguistic skills would be awesome. In practice, [natural language processing](https://chatbotsmagazine.com/natural-language-processing-and-machine-learning-the-core-of-the-modern-smart-chatbot-8755c6343fa5) — NLP to anyone in the field — isn’t there yet. But in the right contexts for the right applications, NLP can make for an easier-to-use interface to features and services. Moreover, an NLP-equipped bot can give the human on the other end the feeling that they’re having a conversation, rather than poking through tedious software menus in yet another part of their lives — first it was the coffee maker, now it’s Facebook.

## How Does NLP Work?

To put it *very* simply, NLP software isn’t looking for keywords in your text, like a search engine. It uses knowledge of sentence structure, idioms, and machine-learned pattern recognition to try to match what you say to an “intent” which has been “classified,” which means the bot has been programmed to identify certain things people want from it, and act upon them. This involves four different areas of AI — see our [What is NLP](https://chatbotsmagazine.com/i-have-the-best-words-a-simple-explanation-of-nlp-in-chatbots-a9d408c76304?source=---------8-----------) for an intro. (We’ll be explaining much more about AI and NLP going forward.)

The scope of all human intents is a lot for a bot to deal with. But your bot doesn’t need to. Gag’s team kept it down to fewer than three dozen intents:

Whenever an intent is “classified” and used in a conversation, the bot can provide an action or quick response. This bot is trained with 20 small-talk intents (e.g. ask it: “tell me a joke”, “what are you up to?” or stupid things like “catch me outside”) and 12 specific intents that are specifically trained for this customer. These are not a lot of classified intents, but it is enough to solve a niche problem for a campaign and it gives users instant gratification.



NLP works well for discussing a focused topic

By integrating the bot on their website, the candidates could use a web page when that was best — such as long platform statements — and let the bot serve as a sort of front-desk personality. It didn’t try to understand and answer anything and everything students typed at it, so that it could focus on the campaign and have ready answers for idle banter.

## DEEP LEARNING FOR CHATBOTS OVERVIEW

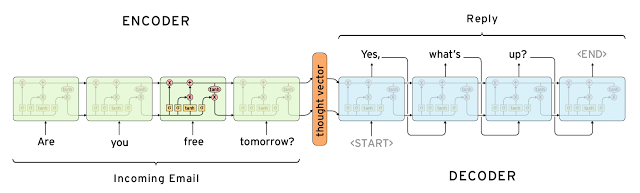
Chatbots, are a hot topic and many companies are hoping to develop bots to have natural conversations indistinguishable from human ones, and many are claiming to be using NLP and Deep Learning techniques to make this possible. But with all the hype around AI it’s sometimes difficult to tell fact from fiction.

In this series I want to go over some of the Deep Learning techniques that are used to build conversational agents, starting off by explaining where we are right now, what’s possible, and what will stay nearly impossible for at least a little while.

## RETRIEVAL-BASED VS. GENERATIVE MODELS

**Retrieval-based models (easier)** use a repository of predefined responses and some kind of heuristic to pick an appropriate response based on the input and context. The heuristic could be as simple as a rule-based expression match, or as complex as an ensemble of Machine Learning classifiers. These systems don’t generate any new text, they just pick a response from a fixed set.

**Generative models (harder)** don’t rely on pre-defined responses. They generate new responses from scratch. Generative models are typically based on Machine Translation techniques, but instead of translating from one language to another, we “translate” from an input to an output (response).



Machine Learning

Both approaches have some obvious pros and cons. Due to the repository of handcrafted responses, retrieval-based methods don’t make grammatical mistakes. However, they may be unable to handle unseen cases for which no appropriate predefined response exists. For the same reasons, these models can’t refer back to contextual entity information like names mentioned earlier in the conversation. Generative models are “smarter”. They can refer back to entities in the input and give the impression that you’re talking to a human. However, these models are hard to train, are quite likely to make grammatical mistakes (especially on longer sentences), and typically require huge amounts of training data.

Deep Learning techniques can be used for both retrieval-based or generative models, but research seems to be moving into the generative direction. Deep Learning architectures like [Sequence to Sequence](http://arxiv.org/abs/1409.3215) are uniquely suited for generating text and researchers are hoping to make rapid progress in this area. However, we’re still at the early stages of building generative models that work reasonably well. Production systems are more likely to be retrieval-based for now.