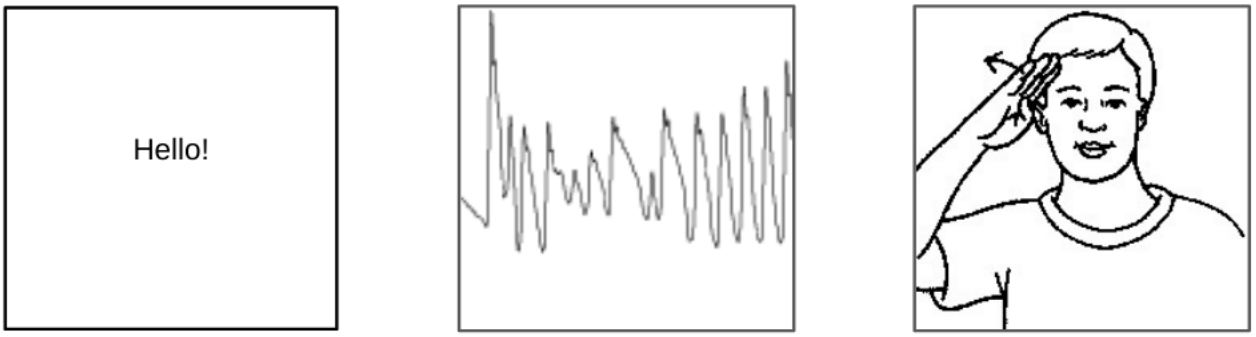
Introduction to Natural Language Processing

Ahmet Melek
ahmetmeleq@gmail.com

Contents

1 Natural Language Processing

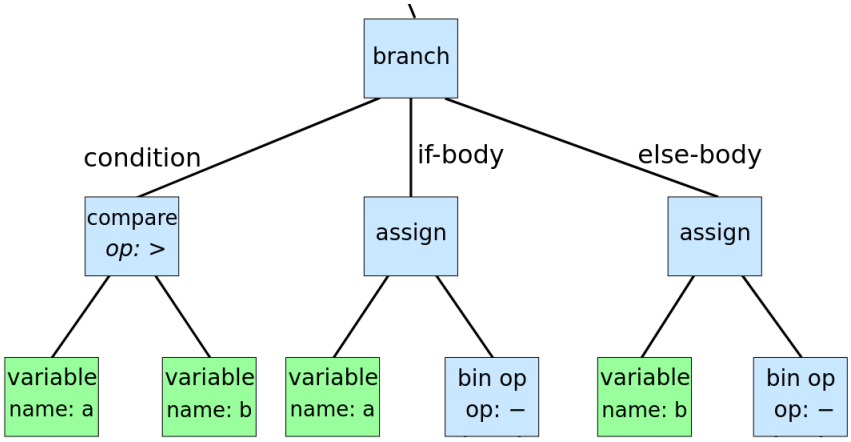
In its broadest sense, Language Processing is analysing or modelling any information that could be considered as "language". Many things can be considered as language. Just as the written text here is expressed with language, the same content would be expressed with a language even if it was in audio format. Or even if it would be expressed in "sign language" with video format, it would still be in the borders of language processing.



Expressions in text, audio, and visual format, all including language components.

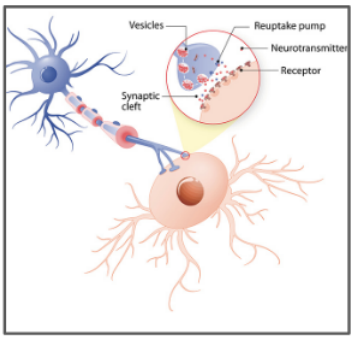
All of the above could be considered natural language, a type of language which is made by humans, and evolved in the course of history. However, there are also formal languages, languages

which are both "constructed" by humans, and which also abide by strict language rules, with no exceptions. These include programming languages, or formal theories in mathematics.



A parse tree for a programming statement, with strict definitions on which expression stands for which operation / data item.

Apart from the categories above, there are also structures in nature that are not made by humans, and that still can be considered as language, or alike. The scientific field that is interested in this kind of languages is Biocommunications, and, there is a popular approach in the field of Biocommunications to use NLP techniques to understand these biological structures better.



Ants and bees utilize chemical signals to find their ways, likewise, nerve cells use electrical signals to transmit information. These chemical and electrical signals are similar to language in terms of how they help individuals communicate.

As you can see, all of these different "languages" above, have similarities. This is why the popular methods in natural language processing are being tried to get utilized in all of these fields. One of the well known applications of using NLP in a different field is understanding genome structures with language modelling, and using this knowledge to make drug discoveries.

Now that we have more or less knowledge on the definition of language, we can move on to inquire on how we will process that language (tasks in nlp), and later, on the terminology that we will use when doing language processing (terms for nlp).

2 Tasks in NLP

What are some ways to analyze, and utilize "language" such that it will produce academic, industrial, or social benefit? It is always good to inquire about various issues on our own before getting other people's answers to them. So try to imagine how we could make automatic language tools that could make our lives easier, or, in which ways we could analyze the language to come up with enlightening conclusions. After that, let's move to the next paragraph.

To get a chronological view on the evolution of tasks in NLP, you can check out the wikipedia page for "Natural Language Processing". Here is a snapshot of the page. To keep this document simple, we will be discussing the current state of NLP, from a view that is biased towards computer science, and engineering. We could also look onto the issue from the eyes of a linguist, a cognitive scientist, a mathematician, or a statistician; views that are all indispensable in various ways. This section will include a taxonomy for NLP tasks, formed by the author. Please note that this is not the only way to categorize NLP tasks and that it will not be a suitable taxonomy for every situation.

1. Classification Tasks

   (a) Text Classification

   Classifying a whole unit of text. It can be a message, a tweet, a paragraph, or a news article.

   Examples: Sentiment Analysis, Intent Recognition

   (b) Token Classification

   Classifying a token, or a set of tokens. It can be a word, a subword, or a morpheme.

   Examples: Entity Recognition, Part of Speech Tagging, Relationship Extraction

2. Transformation Tasks

   (a) Token-to-Token Transformations

   Transforming a token / or a set of tokens into another token / set of tokens. Difference with text-to-text transformations is that we are interested in the transformation of each token, rather than the general transformation of the broader text.

   Examples: Stemming, Lemmatization, Morphological Segmentation, Tokenization

   (b) Text-to-Text Transformations

   General transformation between whole units of texts.

   Examples: Translation, Summarization, Sentence Boundary Disambiguation (Sentence Segmentation), Paragraph Segmentation

   (c) Audio-to-Text and Text-to-Audio Transformations

   Examples: Speech Recognition, Speech Segmentation

   (d) Image/Video-to-Text and Text-to-Image/Video Transformations

   Examples: Optical Character Recognition, Text Guided Image Generation

3. Information Retrieval, Scoring Tasks, and Probability Modelling Tasks

   Given a query (a question) and a set of contextual information units (for example news articles), bring the relevant units (relevant news articles). Units can be whole units of text, or a set of tokens.

   Examples: Text Retrieval, Question Answering, Word Ambiguity Resolution, Relationship Extraction, Coreference Resolution, Extractive Summarization, Extractive Argument Mining, Language Modelling, Language Generation

4. Task Examples that are Modelled in an Unusual Way

   Text Classification with Text Generation, (other examples coming soon)

5. Higher Level Tasks

   Discourse Analysis, Conversation (Chatbots)

3 Terms for NLP

Here are some terms that will help you get started in NLP. For each term that you don't know, search on the term and try to learn how it is used as a term in the field of NLP. With this you will be much more able to understand the articles and content about NLP.

1. Preprocessing

   Corpus, Tokenizers, Special Tokens, Lowercasing, Stopword and Punctuation Cleaning, Stemming and Lemmatization

2. Obtaining Word Features

   Vector Space Models (Embedding Layers), N-grams, Word Counts, Vocabulary

3. Models and Algorithms

   Transformers (and attention), Sequential Models (RNN, GRU, LSTM), Cosine Similarity, Edit Distance Logistic Regression, Bayesian Models, kNN algorithm, Hash Tables, Index Search Algorithms, Hidden Markov Models, Viterbi Algorithm, Dynamic Programming

4. Problems

   Sentiment Analysis, Intent Recognition, Named Entity Recognition, Part of Speech Tagging, Relationship Extraction, Named Entity Linking, Stemming, Lemmatization, Morphological Segmentation, Tokenization, Translation, Summarization, Sentence Boundary Disambiguation (SentenceSegmentation), Paragraph Segmentation, Speech Recognition, Speech Segmentation, Optical Character Recognition, Text Guided Image Generation, Text Retrieval, Question Answering, Word Ambiguity Resolution, Relationship Extraction, Coreference Resolution, Extractive Summarization, Extractive Argument Mining,Language Modelling, Language Generation

5. Visualization

   Dimensionality Reduction: PCA, LDA, tSNE

6. Evaluation

   Accuracy, Precision, Recall, F1 Score, Confusion Matrix, Perplexity, BLEU, ROGUE

7. Experimentation and Development

   Experiment Tracking for ML, ML Deployment

4 Methods for Input Representation in NLP

Before solving any problem in NLP, we need to convert the text data into other formats, because none of the NLP algorithms use the text in raw format. Methods separate the text either into tokens or documents.

- "Tokens" could be characters, sub-words, words, or n-grams. It depends on our problem type to chose either of these. After representing the data as tokens, we process each token to achieve our task goal.
- We could also represent our data with "Documents". In this case, rather than having each character/sub-word/word/n-gram as a unit, we rather consider a whole document as a unit. A document could be a sentence, a paragraph, a group of paragraphs, a whole PDF document, or a whole webpage.

1. Representing the Data as Tokens

   (a) Representing Characters
   Check out the ElMo paper and Character Embeddings

   (b) Representing the Data as Words and Subwords
   Tokenization (and Vocabulary formation), Word Embeddings
   Examples of unsupervised sub-word tokenization algorithms:
   i. BytePair Encoding
   ii. SentencePiece Algorithm

   (c) Representing the data as N-grams, or, as a sequence of tokens
   i. N-grams: N-gram is used to catch multi-word expressions in the text, such as the expression "Homo Sapiens". To catch that expression with word-level tokens, we would need to consider 2-length sequences, because the expression is two words long. That would be a 2-gram representation.
   ii. Sequence of Tokens: There are also other approaches in which we identify the length and content of an expressions with algorithms, rather than a set length N (n of the n-gram). In these approaches we would consider our representations to be "entities", or "text spans", rather than "N-grams". To obtain this kind of a representation, we would apply NER, or similar algorithms, to the data.

2. Representing the Data as Documents

   Tfidf algorithm and other similar algorithms are examples to the document representation. In this approach, we find some characteristics for each document, called "features", and represent a document as a set of features.

3. Representing the Data as a KnowledgeBase

   A knowledgebase is mostly a post-processed product that we obtain after we perform most of the NLP tasks on our text data. It can be a graph of connected documents, or a graph of connected entities.

5 Popular Algorithms in NLP

As we have discussed on the tasks in NLP and input representation, we can now get into detail on the types of algorithms that we use to perform the tasks. These algorithms could be machine learning related, or not.

1. Sequence Models

   In all languages, sequentiality plays a great role. For example, a sentence is a sequence of words. A paragraph is a sequence of sentences. The order of the words in a sentence is important for the meaning in the sentence.

   A non-sequential example would be the money amounts in your wallet. If you have a 50 dollar bill, 10 dollar bill, and a 5 dollar bill, you have 65 dollars. If you change the order like 10 dollar / 50 dollar / 5 dollar, then you would still have 65 dollars. The order does not matter.

   A sequence is basically an ordered list. When you shuffle it, it loses some meaning. A set is different than that. Sets do not have an order, or sequentiality.

   There are some popular algorithms that are capable to process sequences well, such as Attention Algorithm (transformer architecture utilizes this), LSTMs, GRUs and RNNs.

2. Language Models

   BERT vb.

3. Unsupervised Tokenization

4. Document Featurization (tfidf)

5. Popular Architectures (neural networks, encoder-decoder, transformer, siamese nets)

6. Other

   Alignment, CRF, Similarity Measures, Clustering Algorithms

7. Extra

   Cosine Similarity, Edit Distance, Logistic Regression, Bayesian Models, kNN algorithm, Hash Tables, Index Search Algorithms, Hidden Markov Models, Viterbi Algorithm, Dynamic Programming.

6 Popular Software Tools / Platforms in NLP

NLTK, spaCy, CoreNLP, allenNLP, huggingface, scikit-learn, tensorflow, keras, deeppavlov, polyglot