A COMPLETE GUIDE TO

Natural Language Processing

Last updated on Jan 11, 2023

TABLE OF CONTENTS

RELEVANT COURSES

Introduction

Natural Language Processing (NLP) is one of the hottest areas of artificial intelligence (AI) thanks to applications like text generators that compose coherent essays, chatbots that fool people into thinking they're sentient, and text-to-image programs that produce photorealistic images of anything you can describe. Recent years have brought a revolution in the ability of computers to understand human languages, programming languages, and even biological and chemical sequences, such as DNA and protein structures, that resemble language. The latest AI models are unlocking these areas to analyze the meanings of input text and generate meaningful, expressive output.

What is Natural Language Processing (NLP)

Natural language processing (NLP) is the discipline of building machines that can manipulate human language — or data that resembles human language — in the way that it is written, spoken, and organized. It evolved from computational linguistics, which uses computer science to understand the principles of language, but rather than developing theoretical frameworks, NLP is an engineering discipline that seeks to build technology to accomplish useful tasks. NLP can be divided into two overlapping subfields: natural language understanding (NLU), which focuses on semantic analysis or determining the intended meaning of text, and natural language generation (NLG), which focuses on text generation by a machine. NLP is separate from — but often used in conjunction with — speech recognition, which seeks to parse spoken language into words, turning sound into text and vice versa.

Why Does Natural Language Processing (NLP) Matter?

and medicine (interpreting or summarizing electronic health records). Conversational agents such as Amazon's Alexa and Apple's Siri utilize NLP to listen to user queries and find answers. The most sophisticated such agents — such as GPT-3, which was recently opened for commercial applications — can generate sophisticated prose on a wide variety of topics as well as power chatbots that are capable of holding coherent conversations. Google uses NLP to improve its search engine results, and social networks like Facebook use it to detect and filter hate speech.

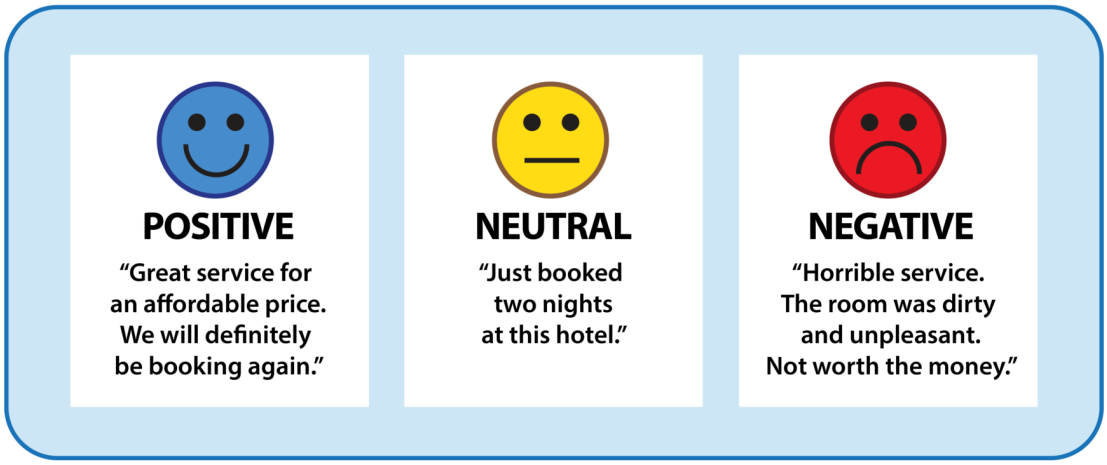
NLP is growing increasingly sophisticated, yet much work remains to be done. Current systems are prone to bias and incoherence, and occasionally behave erratically. Despite the challenges, machine learning engineers have many opportunities to apply NLP in ways that are ever more central to a functioning society.

What is Natural Language Processing (NLP) Used For?

NLP is used for a wide variety of language-related tasks, including answering questions, classifying text in a variety of ways, and conversing with users.

Here are 11 tasks that can be solved by NLP:

- **Sentiment analysis** is the process of classifying the emotional intent of text. Generally, the input to a sentiment classification model is a piece of text, and the output is the probability that the sentiment expressed is positive, negative, or neutral. Typically, this probability is based on either hand-generated features, word n-grams, TF-IDF features, or using deep learning models to capture sequential long- and short-term dependencies. Sentiment analysis is used to classify customer reviews on various online platforms as well as for niche applications like identifying signs of mental illness in online comments.



Given text, sentiment analysis classifies its emotional quality.

- **Toxicity classification** is a branch of sentiment analysis where the aim is not just to classify hostile intent but also to classify particular categories such as threats, insults, obscenities, and hatred towards certain identities. The input to such a model is text, and the output is generally the probability of each class of toxicity. Toxicity classification models can be used to moderate and improve online conversations by silencing offensive comments, detecting hate speech, or scanning documents for defamation.
- **Machine translation** automates translation between different languages. The input to such a model is text in a specified source language, and the output is the text in a specified target language. Google Translate is perhaps the most famous mainstream application. Such models are used to improve communication between people on social-media platforms such as Facebook or Skype. Effective approaches to machine translation can distinguish between words with similar meanings. Some systems also perform language identification; that is, classifying text as being in one language or another.
- **Named entity recognition** aims to extract entities in a piece of text into predefined categories such as personal names, organizations, locations, and quantities. The input to such a model is generally text, and the output is the various named entities along with their start and end positions. Named entity recognition is useful in applications such as summarizing news articles and combating disinformation. For example, here is what a named entity recognition model could provide:

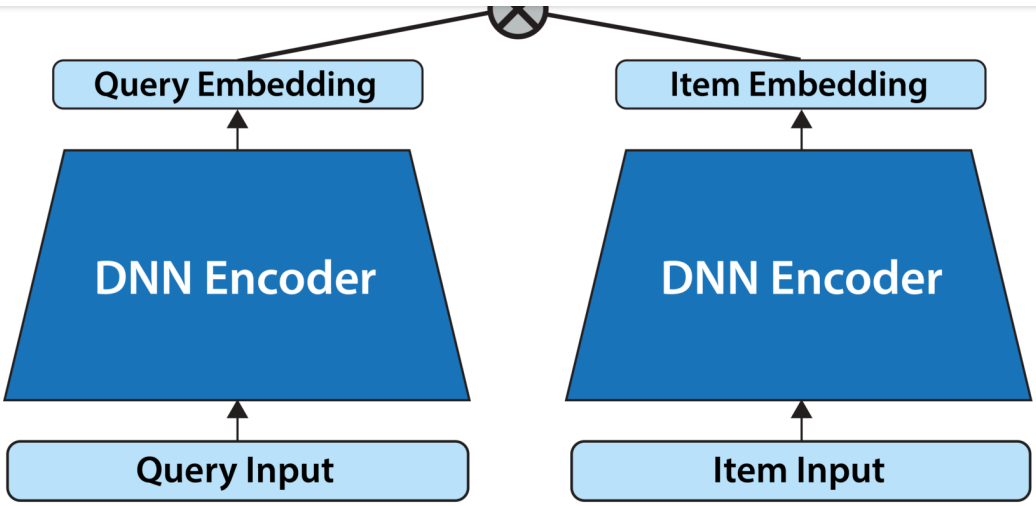
spaCy named entity recognition tagging of the first paragraph of Andrew Ng's Wikipedia page. "NORP" stands for nationalities or religious or political groups. Note that spaCy incorrectly labels "AI" as "GPE," for geopolitical entity.

- **Spam detection** is a prevalent binary classification problem in NLP, where the purpose is to classify emails as either spam or not. Spam detectors take as input an email text along with various other subtexts like title and sender's name. They aim to output the probability that the mail is spam. Email providers like Gmail use such models to provide a better user experience by detecting unsolicited and unwanted emails and moving them to a designated spam folder.
- **Grammatical error correction** models encode grammatical rules to correct the grammar within text. This is viewed mainly as a sequence-to-sequence task, where a model is trained on an ungrammatical sentence as input and a correct sentence as output. Online grammar checkers like Grammarly and word-processing systems like Microsoft Word use such systems to provide a better writing experience to their customers. Schools also use them to grade student essays.
- **Topic modeling** is an unsupervised text mining task that takes a corpus of documents and discovers abstract topics within that corpus. The input to a topic model is a collection of documents, and the output is a list of topics that defines words for each topic as well as assignment proportions of each topic in a document. Latent Dirichlet Allocation (LDA), one of the most popular topic modeling techniques, tries to view a document as a collection of topics and a topic as a collection of words. Topic modeling is being used commercially to help lawyers find evidence in legal documents.

tuned to produce text in different genres and formats — including tweets, blogs, and even computer code. Text generation has been performed using Markov processes, LSTMs, BERT, GPT-2, LaMDA, and other approaches. It's particularly useful for autocomplete and chatbots.

- **Autocomplete** predicts what word comes next, and autocomplete systems of varying complexity are used in chat applications like WhatsApp. Google uses autocomplete to predict search queries. One of the most famous models for autocomplete is GPT-2, which has been used to write articles, song lyrics, and much more.
- **Chatbots** automate one side of a conversation while a human conversant generally supplies the other side. They can be divided into the following two categories:
  - Database query: We have a database of questions and answers, and we would like a user to query it using natural language.
  - Conversation generation: These chatbots can simulate dialogue with a human partner. Some are capable of engaging in wide-ranging conversations. A high-profile example is Google's LaMDA, which provided such human-like answers to questions that one of its developers was convinced that it had feelings.

- **Information retrieval** finds the documents that are most relevant to a query. This is a problem every search and recommendation system faces. The goal is not to answer a particular query but to retrieve, from a collection of documents that may be numbered in the millions, a set that is most relevant to the query. Document retrieval systems mainly execute two processes: indexing and matching. In most modern systems, indexing is done by a vector space model through Two-Tower Networks, while matching is done using similarity or distance scores. Google recently integrated its search function with a multimodal information retrieval model that works with text, image, and video data.
-



A two-tower network creates a representation of an input query and a group of documents (or items) through two separate networks. Then it compares the representation of the query with that of the documents to find documents that are most relevant to the query.

- **Summarization** is the task of shortening text to highlight the most relevant information. Researchers at Salesforce developed a summarizer that also evaluates factual consistency to ensure that its output is accurate. Summarization is divided into two method classes:
  - **Extractive summarization** focuses on extracting the most important sentences from a long text and combining these to form a summary. Typically, extractive summarization scores each sentence in an input text and then selects several sentences to form the summary.
  - **Abstractive summarization** produces a summary by paraphrasing. This is similar to writing the abstract that includes words and sentences that are not present in the original text. Abstractive summarization is usually modeled as a sequence-to-sequence task, where the input is a long-form text and the output is a summary.
- **Question answering** deals with answering questions posed by humans in a natural language. One of the most notable examples of question answering was Watson, which in 2011 played the television game-show *Jeopardy* against human champions and won by substantial margins. Generally, question-answering tasks come in two flavors:
  - **Multiple choice:** The multiple-choice question problem is composed of a question and a set of possible answers. The learning task is to pick the correct answer.

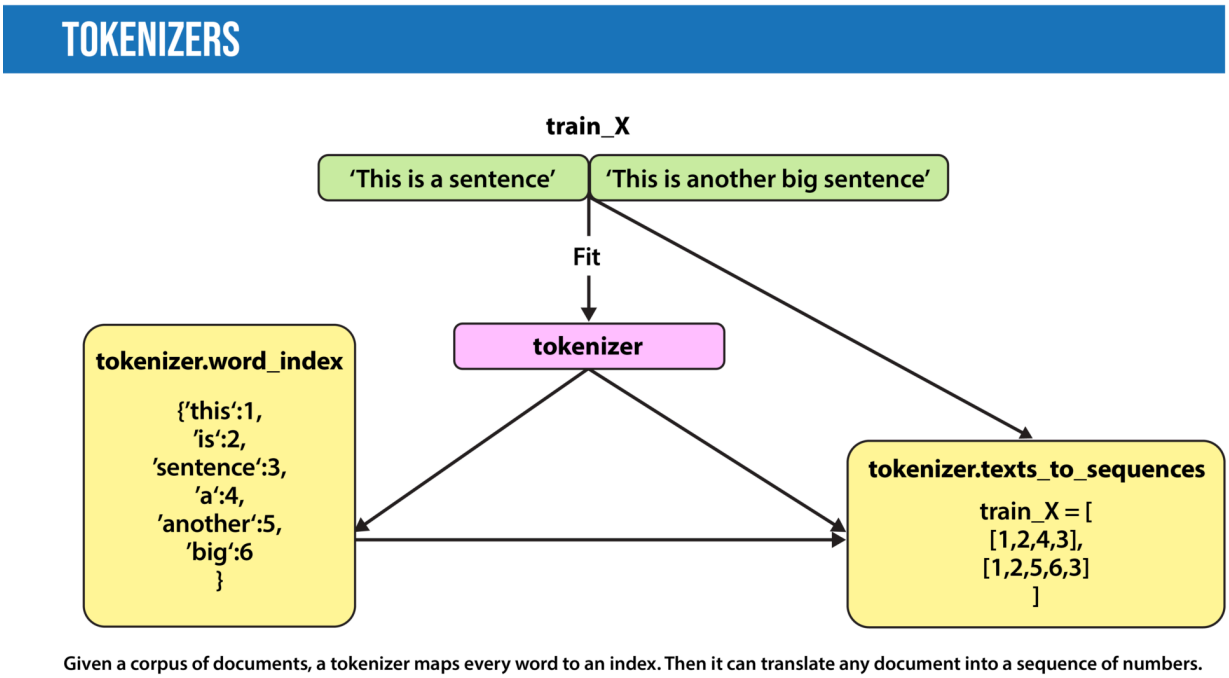
by querying a large number of texts.

How Does Natural Language Processing (NLP) Work?

NLP models work by finding relationships between the constituent parts of language — for example, the letters, words, and sentences found in a text dataset. NLP architectures use various methods for data preprocessing, feature extraction, and modeling. Some of these processes are:

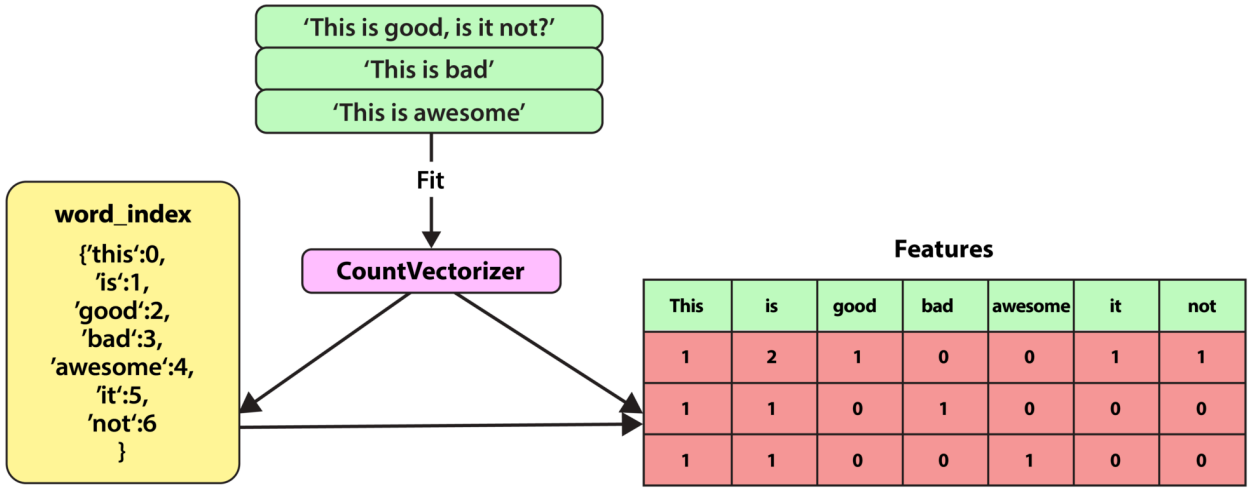
- **Data preprocessing:** Before a model processes text for a specific task, the text often needs to be preprocessed to improve model performance or to turn words and characters into a format the model can understand. Data-centric AI is a growing movement that prioritizes data preprocessing. Various techniques may be used in this data preprocessing:
  - **Stemming and lemmatization**: Stemming is an informal process of converting words to their base forms using heuristic rules. For example, "university," "universities," and "university's" might all be mapped to the base *univers*. (One limitation in this approach is that "universe" may also be mapped to *univers*, even though universe and university don't have a close semantic relationship.) Lemmatization is a more formal way to find roots by analyzing a word's morphology using vocabulary from a dictionary. Stemming and lemmatization are provided by libraries like spaCy and NLTK.
  - **Sentence segmentation** breaks a large piece of text into linguistically meaningful sentence units. This is obvious in languages like English, where the end of a sentence is marked by a period, but it is still not trivial. A period can be used to mark an abbreviation as well as to terminate a sentence, and in this case, the period should be part of the abbreviation token itself. The process becomes even more complex in languages, such as ancient Chinese, that don't have a delimiter that marks the end of a sentence.
  - **Stop word removal** aims to remove the most commonly occurring words that don't add much information to the text. For example, "the," "a," "an," and so on.

represented as numerical tokens for use in various deep learning methods. A method that instructs language models to ignore unimportant tokens can improve efficiency.



Given a corpus of documents, a tokenizer maps every word to an index. Then it can translate any document into a sequence of numbers.

- **Feature extraction:** Most conventional machine-learning techniques work on the features – generally numbers that describe a document in relation to the corpus that contains it – created by either Bag-of-Words, TF-IDF, or generic feature engineering such as document length, word polarity, and metadata (for instance, if the text has associated tags or scores). More recent techniques include Word2Vec, GLoVE, and learning the features during the training process of a neural network.
  - **Bag-of-Words:** Bag-of-Words counts the number of times each word or n-gram (combination of n words) appears in a document. For example, below, the Bag-of-Words model creates a numerical representation of the dataset based on how many of each word in the word_index occur in the document.



Bag-of-Words (through the CountVectorizer method) encodes the total number of times a document uses each word in the associated corpus.

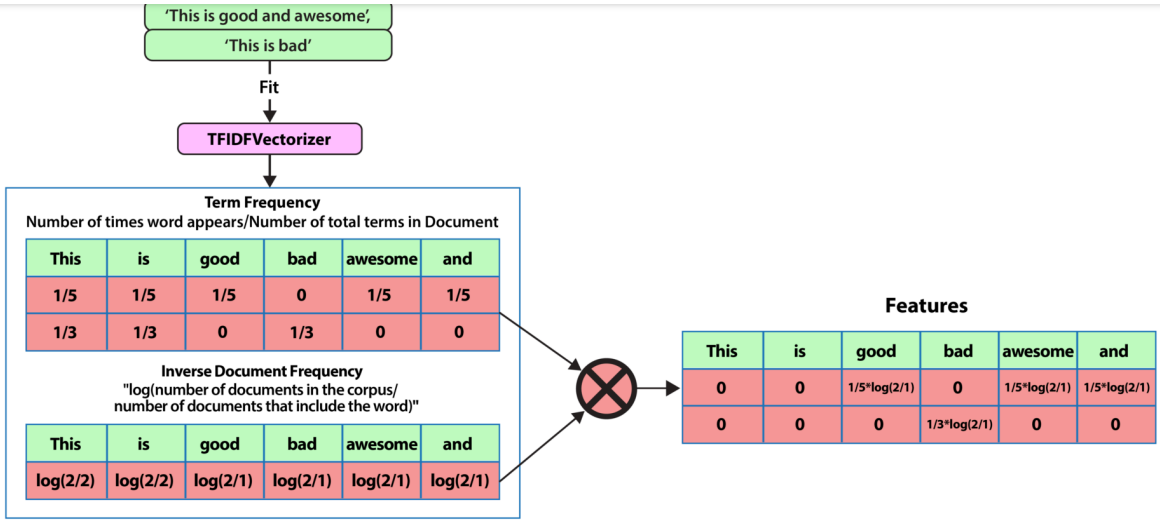
- **TF-IDF:** In Bag-of-Words, we count the occurrence of each word or n-gram in a document. In contrast, with TF-IDF, we weight each word by its importance. To evaluate a word's significance, we consider two things:
  - **Term Frequency:** How important is the word in the document?

TF(word in a document)= Number of occurrences of that word in document / Number of words in document

- **Inverse Document Frequency:** How important is the term in the whole corpus?

IDF(word in a corpus)=log(number of documents in the corpus / number of documents that include the word)

A word is important if it occurs many times in a document. But that creates a problem. Words like "a" and "the" appear often. And as such, their TF score will always be high. We resolve this issue by using Inverse Document Frequency, which is high if the word is rare and low if the word is common across the corpus. The TF-IDF score of a term is the product of TF and IDF.



TF-IDF creates features for each document based on how often each word shows up in a document versus the entire corpus.

- **Word2Vec**, introduced in 2013, uses a vanilla neural network to learn high-dimensional word embeddings from raw text. It comes in two variations: Skip-Gram, in which we try to predict surrounding words given a target word, and Continuous Bag-of-Words (CBOW), which tries to predict the target word from surrounding words. After discarding the final layer after training, these models take a word as input and output a word embedding that can be used as an input to many NLP tasks. Embeddings from Word2Vec capture context. If particular words appear in similar contexts, their embeddings will be similar.
- **GLoVE** is similar to Word2Vec as it also learns word embeddings, but it does so by using matrix factorization techniques rather than neural learning. The GLoVE model builds a matrix based on the global word-to-word co-occurrence counts.
- **Modeling:** After data is preprocessed, it is fed into an NLP architecture that models the data to accomplish a variety of tasks.
    - Numerical features extracted by the techniques described above can be fed into various models depending on the task at hand. For example, for classification, the output from the TF-IDF vectorizer could be provided to logistic regression, naive Bayes, decision trees, or gradient boosted trees. Or, for named entity recognition, we can use hidden Markov models along with n-grams.
    - Deep neural networks typically work without using extracted features, although we can still use TF-IDF or Bag-of-Words features as an input.

models that use Markov assumption are one example:

$P(W_n)=P(W_n|W_{n-1})$

Deep learning is also used to create such language models. Deep-learning models take as input a word embedding and, at each time state, return the probability distribution of the next word as the probability for every word in the dictionary. Pre-trained language models learn the structure of a particular language by processing a large corpus, such as Wikipedia. They can then be fine-tuned for a particular task. For instance, BERT has been fine-tuned for tasks ranging from fact-checking to writing headlines.

Top Natural Language Processing (NLP) Techniques

Most of the NLP tasks discussed above can be modeled by a dozen or so general techniques. It's helpful to think of these techniques in two categories: Traditional machine learning methods and deep learning methods.

Traditional Machine learning NLP techniques:

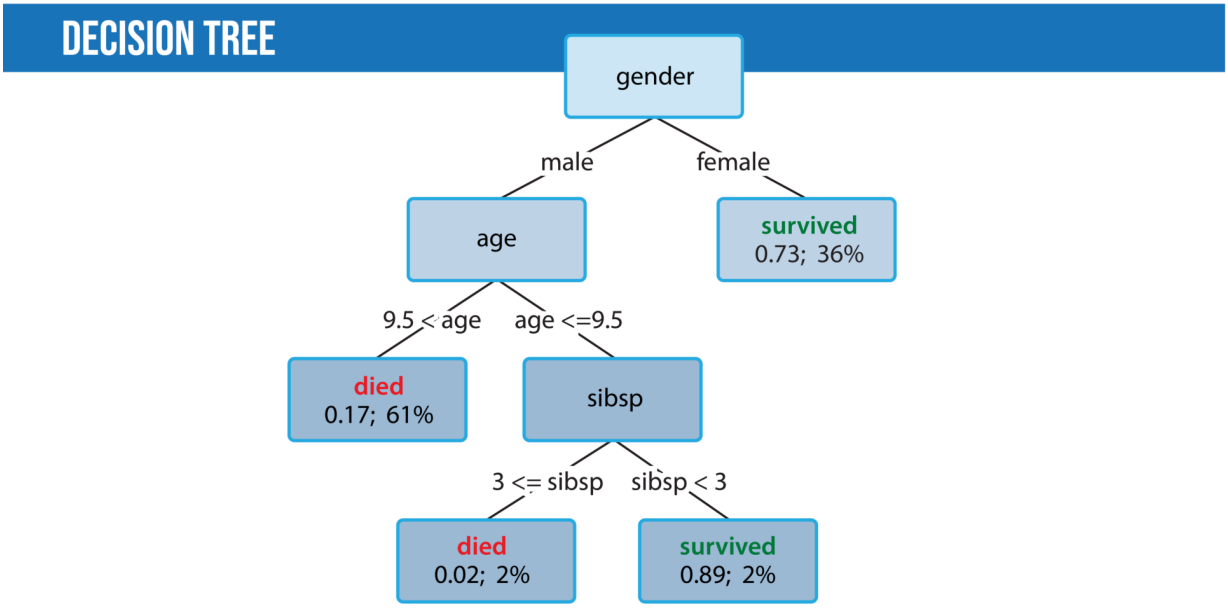
- **Logistic regression** is a supervised classification algorithm that aims to predict the probability that an event will occur based on some input. In NLP, logistic regression models can be applied to solve problems such as sentiment analysis, spam detection, and toxicity classification.
- **Naive Bayes** is a supervised classification algorithm that finds the conditional probability distribution P(label | text) using the following Bayes formula:

P(label | text) = P(label) x P(text|label) / P(text)

and predicts based on which joint distribution has the highest probability. The naive assumption in the Naive Bayes model is that the individual words are independent. Thus:

*P(text|label) = P(word_1|label)*P(word_2|label)*...P(word_n|label)*

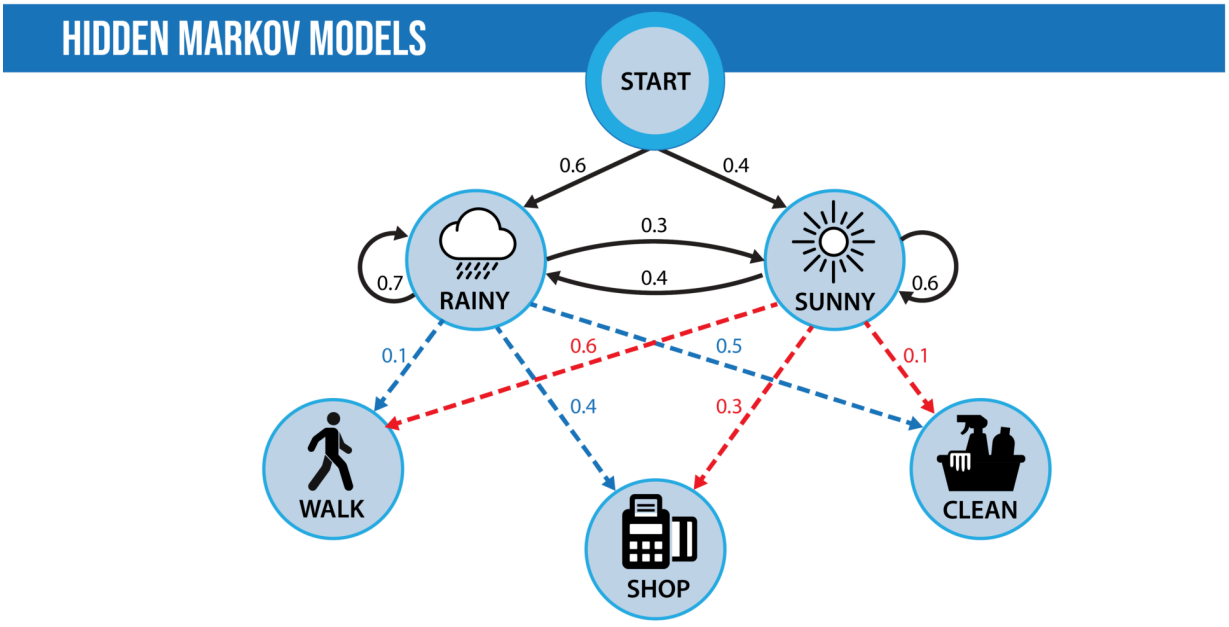
- **Decision trees** are a class of supervised classification models that split the dataset based on different features to maximize information gain in those splits.
-



This decision tree assesses the survival of males and females aboard the Titanic when the ship sank. "Sibsp" stands for the number of siblings and spouses. The numbers of each leaf of the tree show first the probability of survival and second the percentage of examples classified by each leaf. Females had a 73 percent chance of survival, and young males with less than three siblings had a 89 percent chance of survival.

- **Latent Dirichlet Allocation (LDA)** is used for topic modeling. LDA tries to view a document as a collection of topics and a topic as a collection of words. LDA is a statistical approach. The intuition behind it is that we can describe any topic using only a small set of words from the corpus.
- **Hidden Markov models:** Markov models are probabilistic models that decide the next state of a system based on the current state. For example, in NLP, we might suggest the next word based on the previous word. We can model this as a Markov model where we might find the transition probabilities of going from word1 to word2, that is, P(word1|word2). Then we can use a product of these transition probabilities to find the probability of a sentence. The hidden Markov model (HMM) is a probabilistic modeling technique that introduces a hidden state to the Markov model. A hidden state is a property of the data that isn't directly observed. HMMs are used for part-of-speech (POS) tagging where the words of a sentence are the observed states and the POS tags are the hidden states. The HMM adds a concept called emission probability; the probability of an

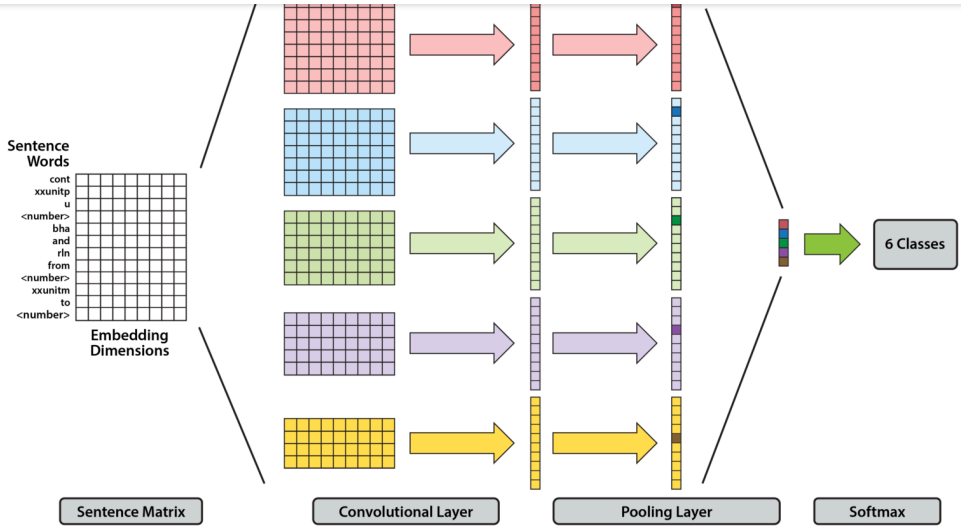
Given a sentence, we can calculate the part-of-speech tag from each word based on both how likely a word was to have a certain part-of-speech tag and the probability that a particular part-of-speech tag follows the part-of-speech tag assigned to the previous word. In practice, this is solved using the Viterbi algorithm.



Someone is trying to recall the weather of last week based on what they did last week. Their actions are the observed states, and the different types of weather are the hidden states. Lines between weather states show the transition probabilities, while lines from weather states to actions show the emission probabilities (likelihood that they performed one action given the weather).

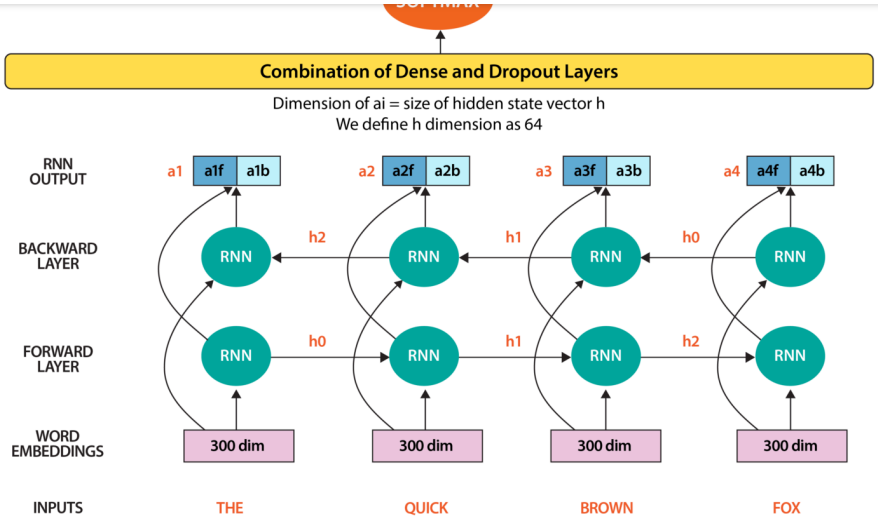
Deep learning NLP Techniques:

- **Convolutional Neural Network (CNN):** The idea of using a CNN to classify text was first presented in the paper "Convolutional Neural Networks for Sentence Classification" by Yoon Kim. The central intuition is to see a document as an image. However, instead of pixels, the input is sentences or documents represented as a matrix of words.



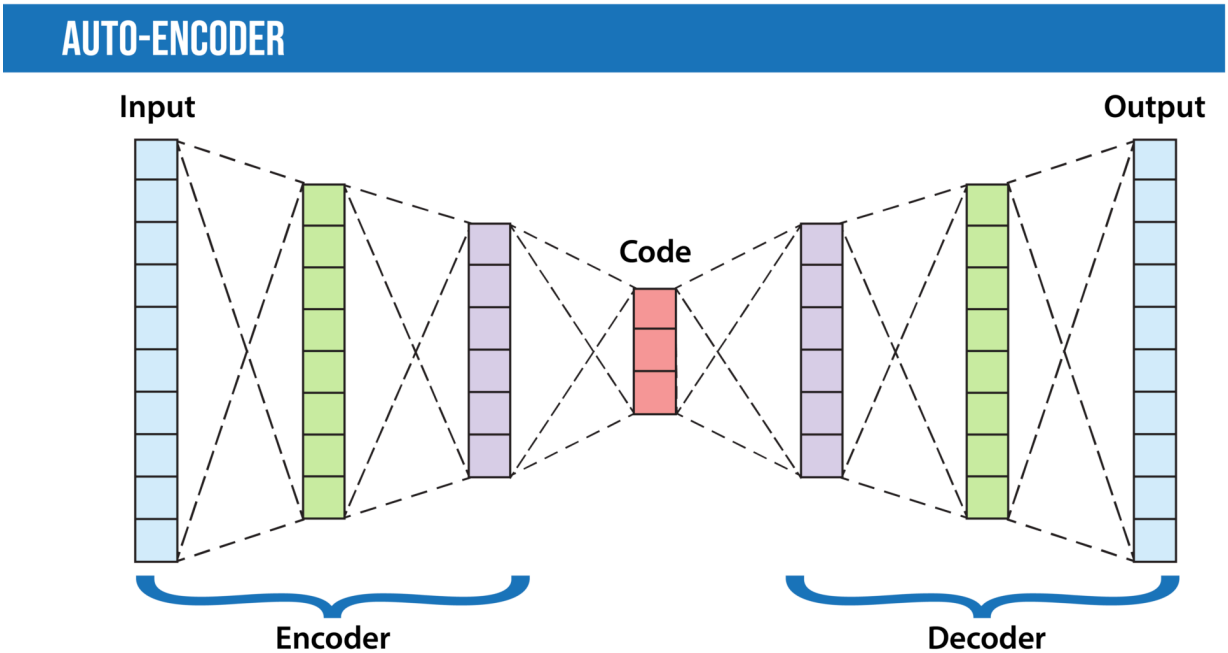
Given a sentence, a convolutional neural network uses convolutional layers to refine representations of input words, before combining them to render a classification.

- **Recurrent Neural Network (RNN)**: Many techniques for text classification that use deep learning process words in close proximity using n-grams or a window (CNNs). They can see “New York” as a single instance. However, they can’t capture the context provided by a particular text sequence. They don’t learn the sequential structure of the data, where every word is dependent on the previous word or a word in the previous sentence. RNNs remember previous information using hidden states and connect it to the current task. The architectures known as Gated Recurrent Unit (GRU) and long short-term memory (LSTM) are types of RNNs designed to remember information for an extended period. Moreover, the bidirectional LSTM/GRU keeps contextual information in both directions, which is helpful in text classification. RNNs have also been used to generate mathematical proofs and translate human thoughts into words.



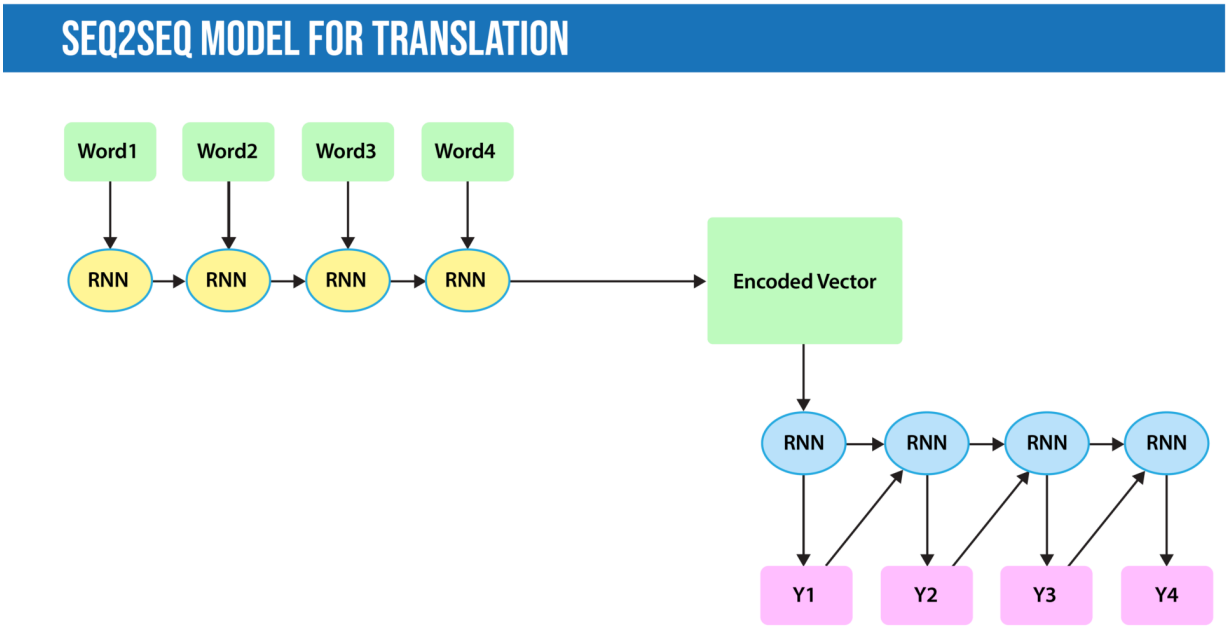
A bidirectional recurrent neural network processes the input both forward and backward to improve the representations it produces.

- **Autoencoders** are deep learning encoder-decoders that approximate a mapping from X to X, i.e., input=output. They first compress the input features into a lower-dimensional representation (sometimes called a latent code, latent vector, or latent representation) and learn to reconstruct the input. The representation vector can be used as input to a separate model, so this technique can be used for dimensionality reduction. Among specialists in many other fields, geneticists have applied autoencoders to spot mutations associated with diseases in amino acid sequences.



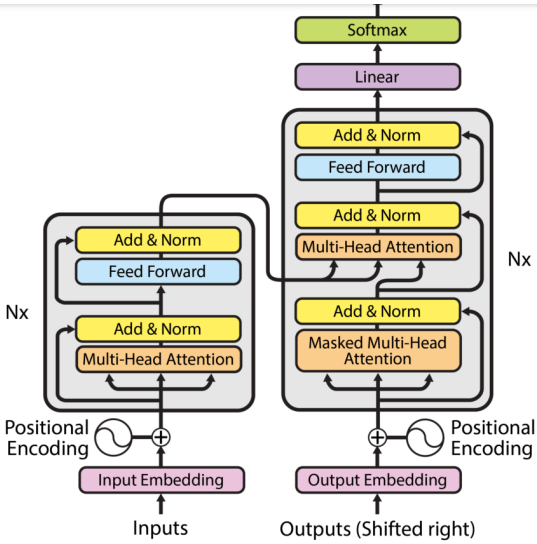
An autoencoder uses an encoder to compress an input into a representation and a decoder to reconstruct the input from the representation.

summarization, and similar tasks. The encoder encapsulates the information in a text into an encoded vector. Unlike an autoencoder, instead of reconstructing the input from the encoded vector, the decoder's task is to generate a different desired output, like a translation or summary.



Given a sentence, a Recurrent Neural Network encodes the sentence and then iteratively generates a translation.

- **Transformers:** The transformer, a model architecture first described in the 2017 paper "Attention Is All You Need" (Vaswani, Shazeer, Parmar, et al.), forgoes recurrence and instead relies entirely on a self-attention mechanism to draw global dependencies between input and output. Since this mechanism processes all words at once (instead of one at a time) that decreases training speed and inference cost compared to RNNs, especially since it is parallelizable. The transformer architecture has revolutionized NLP in recent years, leading to models including BLOOM, Jurassic-X, and Turing-NLG. It has also been successfully applied to a variety of different vision tasks, including making 3D images.
-



The encoder-decoder transformer used for translation. Encoder on the left, decoder on the right. Note that the decoder takes in its previously generated words during generation.

Six Important Natural Language Processing (NLP) Models

Over the years, many NLP models have made waves within the AI community, and some have even made headlines in the mainstream news. The most famous of these have been chatbots and language models. Here are some of them:

- **Eliza** was developed in the mid-1960s to try to solve the Turing Test; that is, to fool people into thinking they're conversing with another human being rather than a machine. Eliza used pattern matching and a series of rules without encoding the context of the language.
- **Tay** was a chatbot that Microsoft launched in 2016. It was supposed to tweet like a teen and learn from conversations with real users on Twitter. The bot adopted phrases from users who tweeted sexist and racist comments, and Microsoft deactivated it not long afterward. Tay illustrates some points made by the "Stochastic Parrots" paper, particularly the danger of not debiasing data.
- **BERT** and his Muppet friends: Many deep learning models for NLP are named after Muppet characters, including ELMo, BERT, Big BIRD, ERNIE, Kermit, Grover, RoBERTa, and Rosita. Most of these models are good at providing contextual embeddings and enhanced knowledge representation.

input prompt. The model is based on the transformer architecture. The previous version, GPT-2, is open source. Microsoft acquired an exclusive license to access GPT-3's underlying model from its developer OpenAI, but other users can interact with it via an application programming interface (API). Several groups including EleutherAI and Meta have released open source interpretations of GPT-3.

- **Language Model for Dialogue Applications (LaMDA)** is a conversational chatbot developed by Google. LaMDA is a transformer-based model trained on dialogue rather than the usual web text. The system aims to provide sensible and specific responses to conversations. Google developer Blake Lemoine came to believe that LaMDA is sentient. Lemoine had detailed conversations with AI about his rights and personhood. During one of these conversations, the AI changed Lemoine's mind about Isaac Asimov's third law of robotics. Lemoine claimed that LaMDA was sentient, but the idea was disputed by many observers and commentators. Subsequently, Google placed Lemoine on administrative leave for distributing proprietary information and ultimately fired him.
- **Mixture of Experts (MoE):** While most deep learning models use the same set of parameters to process every input, MoE models aim to provide different parameters for different inputs based on efficient routing algorithms to achieve higher performance. Switch Transformer is an example of the MoE approach that aims to reduce communication and computational costs.

Programming Languages, Libraries, And Frameworks For Natural Language Processing (NLP)

Many languages and libraries support NLP. Here are a few of the most useful.

- **Python** is the most-used programming language to tackle NLP tasks. Most libraries and frameworks for deep learning are written for Python. Here are a few that practitioners may find helpful:
  - **Natural Language Toolkit (NLTK)** is one of the first NLP libraries written in Python. It provides easy-to-use interfaces to corpora and lexical resources such as WordNet. It also provides a suite of text-processing libraries for classification, tagging, stemming, parsing, and semantic reasoning.

implements many popular models like BERT. spaCy can be used for building production-ready systems for named entity recognition, part-of-speech tagging, dependency parsing, sentence segmentation, text classification, lemmatization, morphological analysis, entity linking, and so on.

    - **Deep Learning libraries:** Popular deep learning libraries include TensorFlow and PyTorch, which make it easier to create models with features like automatic differentiation. These libraries are the most common tools for developing NLP models.
    - **Hugging Face** offers open-source implementations and weights of over 135 state-of-the-art models. The repository enables easy customization and training of the models.
    - **Gensim** provides vector space modeling and topic modeling algorithms.
- **R:** Many early NLP models were written in R, and R is still widely used by data scientists and statisticians. Libraries in R for NLP include TidyText, Weka, Word2Vec, SpaCyR, TensorFlow, and PyTorch.
- Many other languages including JavaScript, Java, and Julia have libraries that implement NLP methods.

Controversies Surrounding Natural Language Processing (NLP)

NLP has been at the center of a number of controversies. Some are centered directly on the models and their outputs, others on second-order concerns, such as who has access to these systems, and how training them impacts the natural world.

- **Stochastic parrots:** A 2021 paper titled "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" by Emily Bender, Timnit Gebru, Angelina McMillan-Major, and Margaret Mitchell examines how language models may repeat and amplify biases found in their training data. The authors point out that huge, uncurated datasets scraped from the web are bound to include social biases and other undesirable information, and models that are trained on them will absorb these flaws. They advocate greater care in curating and documenting datasets, evaluating a model's potential impact prior to development, and

- **Coherence versus sentience:** Recently, a Google engineer tasked with evaluating the LaMDA language model was so impressed by the quality of its chat output that he believed it to be sentient. The fallacy of attributing human-like intelligence to AI dates back to some of the earliest NLP experiments.
- **Environmental impact:** Large language models require a lot of energy during both training and inference. One study estimated that training a single large language model can emit five times as much carbon dioxide as a single automobile over its operational lifespan. Another study found that models consume even more energy during inference than training. As for solutions, researchers have proposed using cloud servers located in countries with lots of renewable energy as one way to offset this impact.
- **High cost leaves out non-corporate researchers:** The computational requirements needed to train or deploy large language models are too expensive for many small companies. Some experts worry that this could block many capable engineers from contributing to innovation in AI.
- **Black box:** When a deep learning model renders an output, it's difficult or impossible to know why it generated that particular result. While traditional models like logistic regression enable engineers to examine the impact on the output of individual features, neural network methods in natural language processing are essentially black boxes. Such systems are said to be "not explainable," since we can't explain how they arrived at their output. An effective approach to achieve explainability is especially important in areas like banking, where regulators want to confirm that a natural language processing system doesn't discriminate against some groups of people, and law enforcement, where models trained on historical data may perpetuate historical biases against certain groups.

"**Nonsense on stilts**": Writer Gary Marcus has criticized deep learning-based NLP for generating sophisticated language that misleads users to believe that natural language algorithms understand what they are saying and mistakenly assume they are capable of more sophisticated reasoning than is currently possible.

If you are just starting out, many excellent courses can help.

COURSE

Machine Learning Specialization

A foundational set of three courses that introduces beginners to the fundamentals of learning algorithms. Prerequisites include high-school math and basic programming skills

View Course

COURSE

Deep Learning Specialization

An intermediate set of five courses that help learners get hands-on experience building and deploying neural networks, the technology at the heart of today's most advanced NLP and other sorts of AI models.

View Course

COURSE

Natural Language Processing Specialization

An intermediate set of four courses that provide learners with the theory and application behind the most relevant and widely used NLP models.

View Course

If you want to learn more about NLP, try reading research papers. Work through the papers that introduced the models and techniques described in this article. Most are easy to find on arxiv.org. You might also take a look at these resources:

- *NLP News*: A newsletter from Sebastian Ruder, a research scientist at Google, focused on what's new in NLP.
- Papers with Code: A web repository of machine learning research, tasks, benchmarks, and datasets.

We highly recommend learning to implement basic algorithms (linear and logistic regression, Naive Bayes, decision trees, and vanilla neural networks) in Python. The next step is to take an open-source implementation and adapt it to a new dataset or task.

Conclusion

NLP is one of the fast-growing research domains in AI, with applications that involve tasks including translation, summarization, text generation, and sentiment analysis. Businesses use NLP to power a growing number of applications, both internal — like detecting insurance fraud, determining customer sentiment, and optimizing aircraft maintenance — and customer-facing, like Google Translate.

Aspiring NLP practitioners can begin by familiarizing themselves with foundational AI skills: performing basic mathematics, coding in Python, and using algorithms like decision trees, Naive Bayes, and logistic regression. Online courses can help you build your foundation. They can also help as you proceed into specialized topics. Specializing in NLP requires a working knowledge of things like neural networks, frameworks like PyTorch and TensorFlow, and various data preprocessing techniques. The transformer architecture, which has revolutionized the field since it was introduced in 2017, is an especially important architecture.

NLP is an exciting and rewarding discipline, and has potential to profoundly impact the world in many positive ways. Unfortunately, NLP is also the focus of several controversies, and understanding them is also part of being a responsible practitioner. For instance, researchers have found that models will parrot biased language found in their training data, whether they're counterfactual, racist, or hateful. Moreover, sophisticated language models can be used to generate disinformation. A broader concern is that training large models produces substantial greenhouse gas emissions.

beginners and those who are ready to specialize. No matter your current level of expertise or aspirations, remember to keep learning!