Natural Language Processing (NLP): What Is It & How Does it Work?
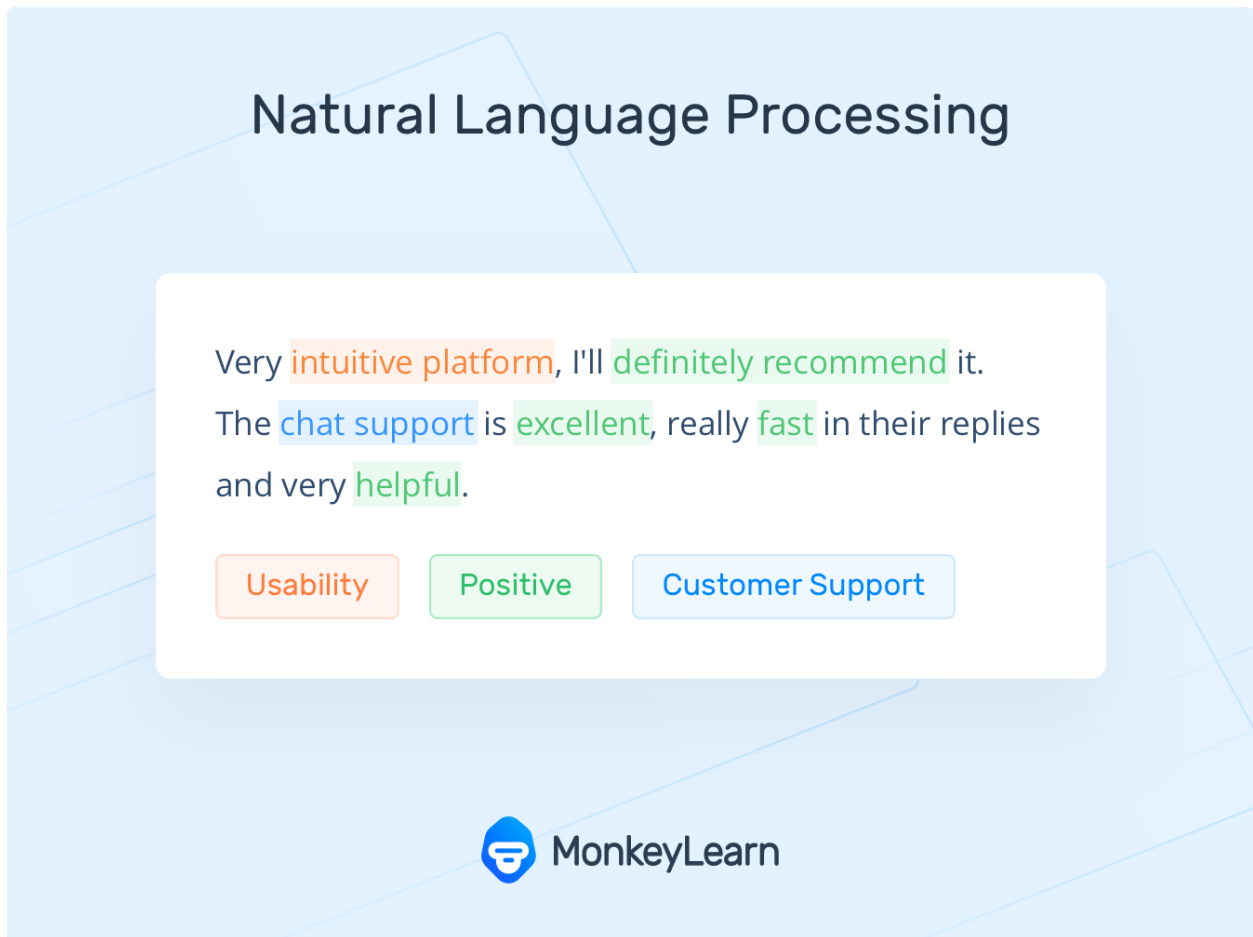
Natural Language Processing (NLP) allows machines to break down and interpret human language. It's at the core of tools we use every day – from translation software, chatbots, spam filters, and search engines, to grammar correction software, voice assistants, and [social media monitoring tools](#).


Start your NLP journey with no-code tools

[TRY NOW](#)

In this guide, you'll learn about the basics of [Natural Language Processing](#) and some of its challenges, and discover the most popular [NLP applications](#) in business. Finally, you'll see for yourself just how easy it is to get started with code-free [natural language processing tools](#).

What Is Natural Language Processing (NLP)?
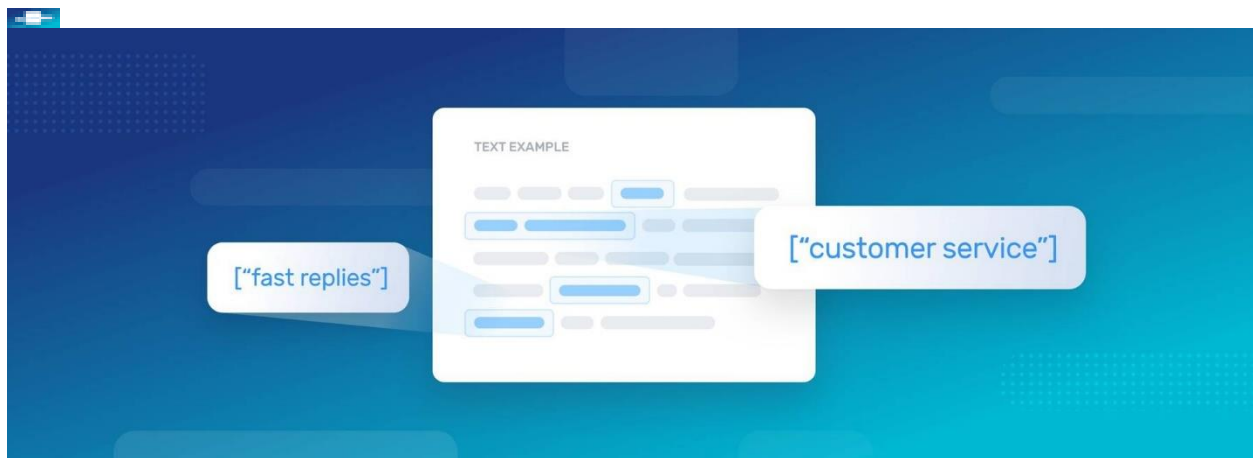

[Natural Language Processing (NLP) is a field of Artificial Intelligence (AI)](#) that makes human language intelligible to machines. NLP combines the power of linguistics and computer science to study the rules and structure of language, and create intelligent systems (run on machine learning and NLP algorithms) capable of understanding, analyzing, and extracting meaning from text and speech.

What Is NLP Used For?

NLP is used to [understand the structure and meaning of human language](#) by analyzing different aspects like syntax, semantics, pragmatics, and morphology. Then, computer science transforms this linguistic knowledge into rule-based, [machine learning algorithms](#) that can solve specific problems and perform desired tasks.

Take Gmail, for example. Emails are automatically categorized as *Promotions*, *Social*, *Primary*, or *Spam*, thanks to an NLP task called [keyword extraction](). By "reading" words in subject lines and associating them with predetermined tags, machines automatically learn which category to assign emails.

NLP Benefits
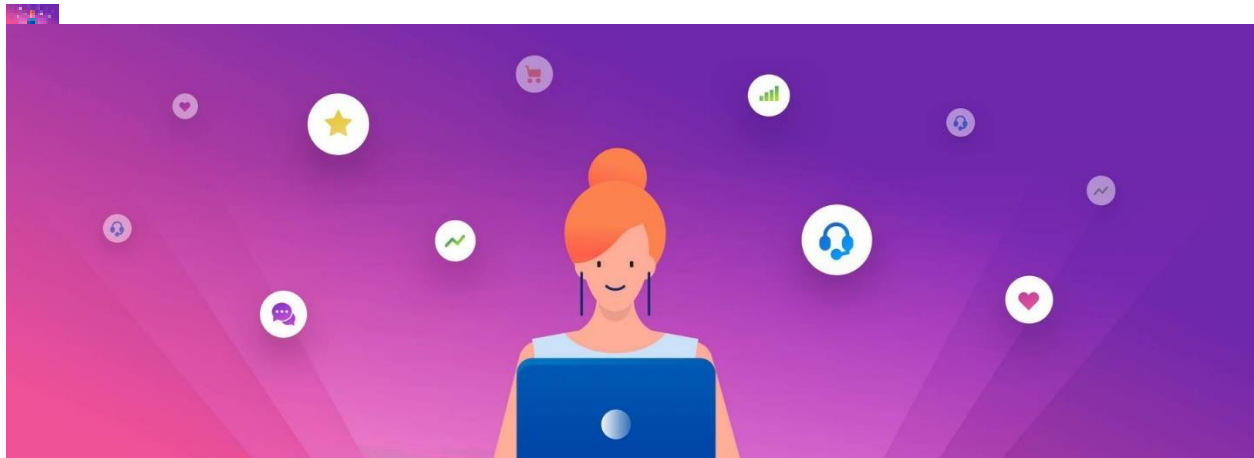
There are many [benefits of NLP](), but here are just a few top-level benefits that will help your business become more competitive:

- **Perform large-scale analysis.** Natural Language Processing helps machines automatically understand and analyze huge amounts of [unstructured text data](), like social media comments, customer support tickets, online reviews, news reports, and more.
- **Automate processes in real-time.** Natural language processing tools can help machines learn to sort and route information with little to no human interaction – quickly, efficiently, accurately, and around the clock.
- **Tailor NLP tools to your industry.** Natural language processing algorithms can be tailored to your needs and criteria, like complex, industry-specific language – even sarcasm and misused words.

How Does Natural Language Processing Work?

Using [text vectorization](#), NLP tools transform text into something a machine can understand, then machine learning algorithms are fed [training data](#) and expected outputs (tags) to train machines to make associations between a particular input and its corresponding output. Machines then use statistical analysis methods to build their own "knowledge bank" and discern which features best represent the texts, before making predictions for unseen data (new texts):

Natural Language Processing (NLP)

1. NLP and its Applications:

  - **NLP** refers to the branch of artificial intelligence that focuses on the interaction between computers and humans through natural language.

  - **Applications**: NLP is used in various applications such as sentiment analysis, machine translation, question answering systems, chatbots, text summarization, information extraction, and more.

2. Natural Language Understanding (NLU) vs. NLP:

  - **NLP** deals with the automatic manipulation of natural language, including tasks such as text generation, part-of-speech tagging, named entity recognition, etc.

- **NLU** is a subset of NLP that focuses on comprehension, where the system must understand the meaning and context of the text. It involves tasks like semantic analysis, sentiment analysis, etc.

3. Features and Stages of NLP:

- **Features**: NLP involves various features such as tokenization, stemming, lemmatization, part-of-speech tagging, named entity recognition, sentiment analysis, etc.

- **Stages**:

  1. **Text Preprocessing**: Cleaning and preparing text data.

  2. **Tokenization**: Breaking text into smaller units (tokens).

  3. **Normalization**: Converting tokens to their base or root form (stemming/lemmatization).

  4. **Part-of-Speech Tagging**: Assigning grammatical information to tokens.

  5. **Named Entity Recognition**: Identifying and classifying entities in text.

  6. **Parsing**: Analyzing the syntactic structure of sentences.

  7. **Semantic Analysis**: Extracting meaning from text.

  8. **Feature Engineering**: Selecting and extracting relevant features for a specific task.

  9. **Model Building**: Developing machine learning models for various NLP tasks.

  10. **Evaluation**: Assessing the performance of the NLP system.

NLP Techniques and Components

1. Named Entity Recognition (NER) and Applications:

  - **NER** is the task of identifying and classifying named entities (such as people, organizations, locations) in text.

  - **Applications**: Named entity recognition is used in information retrieval, question answering systems, named entity disambiguation, etc.

2. Stop Words, Tokenization, Stemming, Lemmatization, POS Tagging:

  - **Stop Words**: Common words (e.g., "the", "is", "and") that are often filtered out during text processing.

  - **Tokenization**: Splitting text into tokens (words, phrases, symbols).

  - **Stemming**: Reducing words to their root form by removing suffixes (e.g., "running" to "run").

  - **Lemmatization**: Similar to stemming but results in valid words (e.g., "ran" to "run").

  - **POS Tagging**: Assigning parts of speech to tokens (e.g., noun, verb, adjective).

3. Types of Tokenization Techniques:

  - **Sentence Tokenization**: Splitting text into sentences.

  - **Word Tokenization**: Splitting text into words.

  - **Whitespace Tokenization**: Splitting text based on whitespace.

  - **Regex Tokenization**: Splitting text using regular expressions.

  - **Custom Tokenization**: Tokenization based on specific rules or requirements.

4. Lexicons and WordNet:

   - **Lexicons**: Collections of words or phrases with associated information such as definitions, parts of speech, sentiment scores, etc.

   - **WordNet**: A lexical database of the English language that groups words into sets of synonyms (synsets) and provides semantic relationships between them.

5. Regular Expressions (RegEx):

   - **RegEx** is a sequence of characters that forms a search pattern. It is used for pattern matching within text data.

   - **Example**: Matching all email addresses in a text using the pattern `[\w.-]+@[\w.-]+`.

Ambiguity in NLP

1. Syntactic Ambiguity: Ambiguity arising from the structure or grammar of a sentence (e.g., "I saw the man with the telescope").

2. Semantic Ambiguity: Ambiguity related to the meaning of words or phrases (e.g., "bank" can refer to a financial institution or the edge of a river).

3. Difference: Syntactic ambiguity deals with the structure of sentences, while semantic ambiguity deals with the meaning of words or phrases.

Naive Bayes Algorithm

1. Principle and Steps:

- **Principle**: Naive Bayes is a probabilistic classifier based on Bayes' theorem, which calculates the probability of a label given some observed features.

  - **Steps**:

    1. **Data Preprocessing**: Prepare the training data.

    2. **Feature Extraction**: Represent text data as feature vectors.

    3. **Training**: Calculate probabilities based on the training data.

    4. **Prediction**: Predict the label for new instances using Bayes' theorem.

2. Limitations:

  - **Assumption of Independence**: Naive Bayes assumes that features are independent, which may not hold true in real-world data.
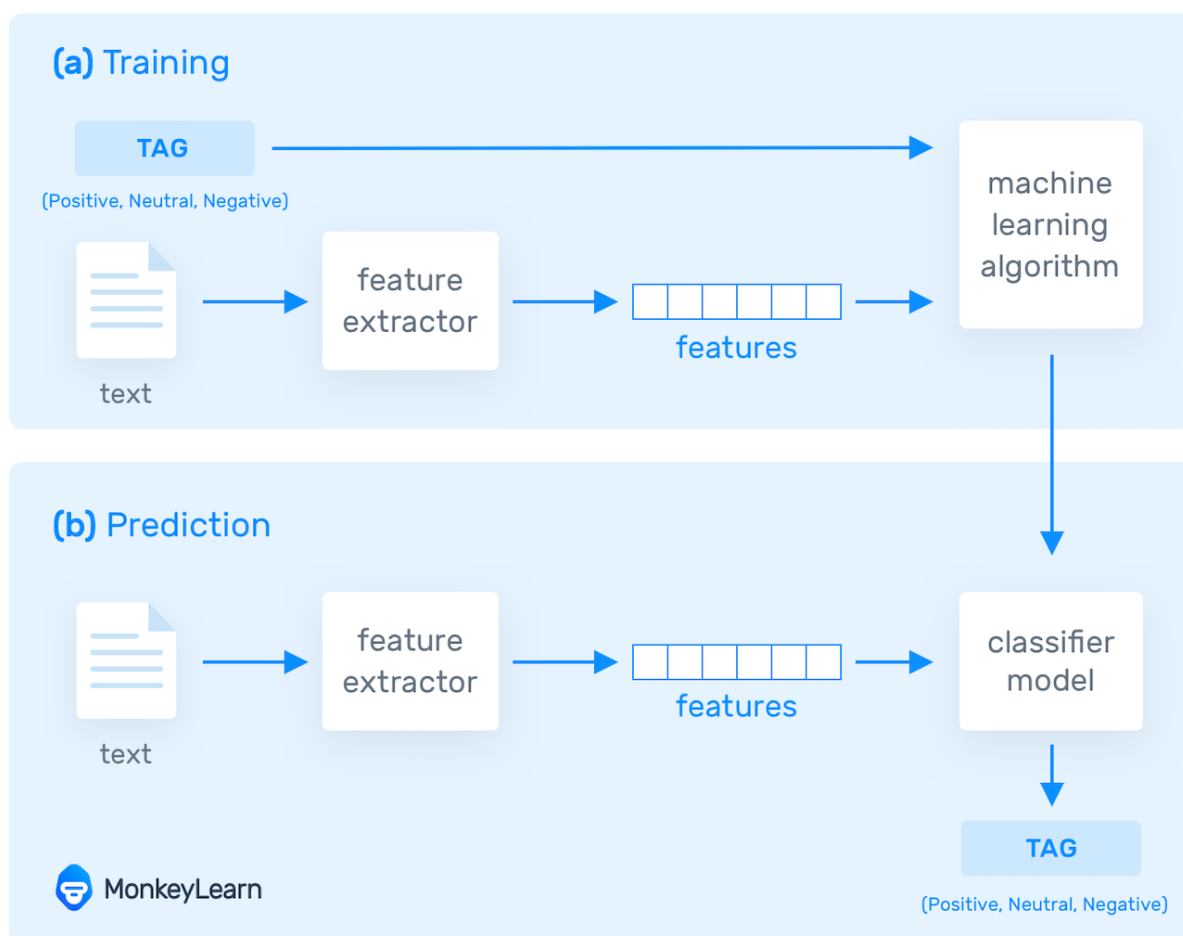
  - **Sensitive to Feature Correlation**: It may perform poorly if features are highly correlated.

  - **Requires Sufficient Training Data**: It requires a large amount of training data to estimate probabilities accurately.

  - **Inability to Handle Out-of-Vocabulary Words**: It struggles with words that are not present in the training data.

In summary, NLP encompasses various techniques and components for processing and understanding natural language, including tokenization, named entity recognition, part-of-speech tagging, and more.

Naive Bayes algorithm is one of the techniques used for classification tasks in NLP, but it has limitations such as the assumption of feature independence and sensitivity to feature correlation.

(a) Training

TAG
(Positive, Neutral, Negative)

text → feature extractor → features → machine learning algorithm

(b) Prediction

text → feature extractor → features → classifier model → TAG
(Positive, Neutral, Negative)

MonkeyLearn

Ultimately, the more data these NLP algorithms are fed, the more accurate the text analysis models will be.

Sentiment analysis (seen in the above chart) is one of the most popular NLP tasks, where machine learning models are trained to classify text by polarity of opinion (positive, negative, neutral, and everywhere in between).

Try out sentiment analysis for yourself by typing text in the NLP model, below

## Test with your own text

Classify Text

Results

TAGCONFIDENCE

Positive99.1%

The biggest advantage of machine learning models is their ability to learn on their own, with no need to define manual rules. You just need a set of relevant training data with several examples for the tags you want to analyze. And with advanced deep learning algorithms, you're able to chain together multiple natural language processing tasks, like sentiment analysis, keyword extraction, topic classification, intent detection, and more, to work simultaneously for super fine-grained results.

Common NLP Tasks & Techniques
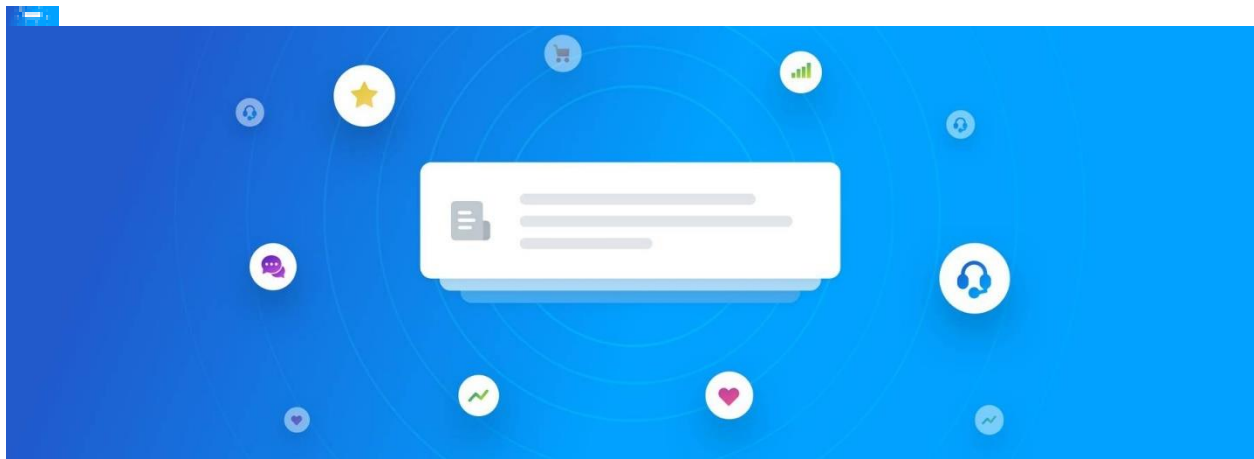
Many natural language processing tasks involve syntactic and semantic analysis, used to break down human language into machine-readable chunks.

Syntactic analysis, also known as parsing or syntax analysis, identifies the syntactic structure of a text and the dependency relationships between words, represented on a diagram called a parse tree.

Semantic analysis focuses on identifying the meaning of language. However, since language is polysemic and ambiguous, semantics is considered one of the most challenging areas in NLP.

Semantic tasks analyze the structure of sentences, word interactions, and related concepts, in an attempt to discover the meaning of words, as well as understand the topic of a text.

Below, we've listed some of the main sub-tasks of both semantic and syntactic analysis:

Tokenization

Tokenization is an essential task in natural language processing used to break up a string of words into semantically useful units called *tokens*.

Sentence tokenization splits sentences within a text, and word tokenization splits words within a sentence. Generally, word tokens are separated by blank spaces, and sentence tokens by stops. However, you can perform high-level tokenization for more complex structures, like words that often go together, otherwise known as collocations (e.g., *New York*).

An example of how word tokenization simplifies text:

Here's an example of how word tokenization simplifies text:

Customer service couldn't be better! = "customer service" "could" "not" "be" "better".

Part-of-speech tagging

Part-of-speech tagging (abbreviated as PoS tagging) involves adding a part of speech category to each token within a text. Some common PoS tags are *verb*, *adjective*, *noun*, *pronoun*, *conjunction*, *preposition*, *intersection*, among others. In this case, the example above would look like this:

"Customer service": NOUN, "could": VERB, "not": ADVERB, be": VERB, "better": ADJECTIVE, "!": PUNCTUATION

PoS tagging is useful for identifying relationships between words and, therefore, understand the meaning of sentences.
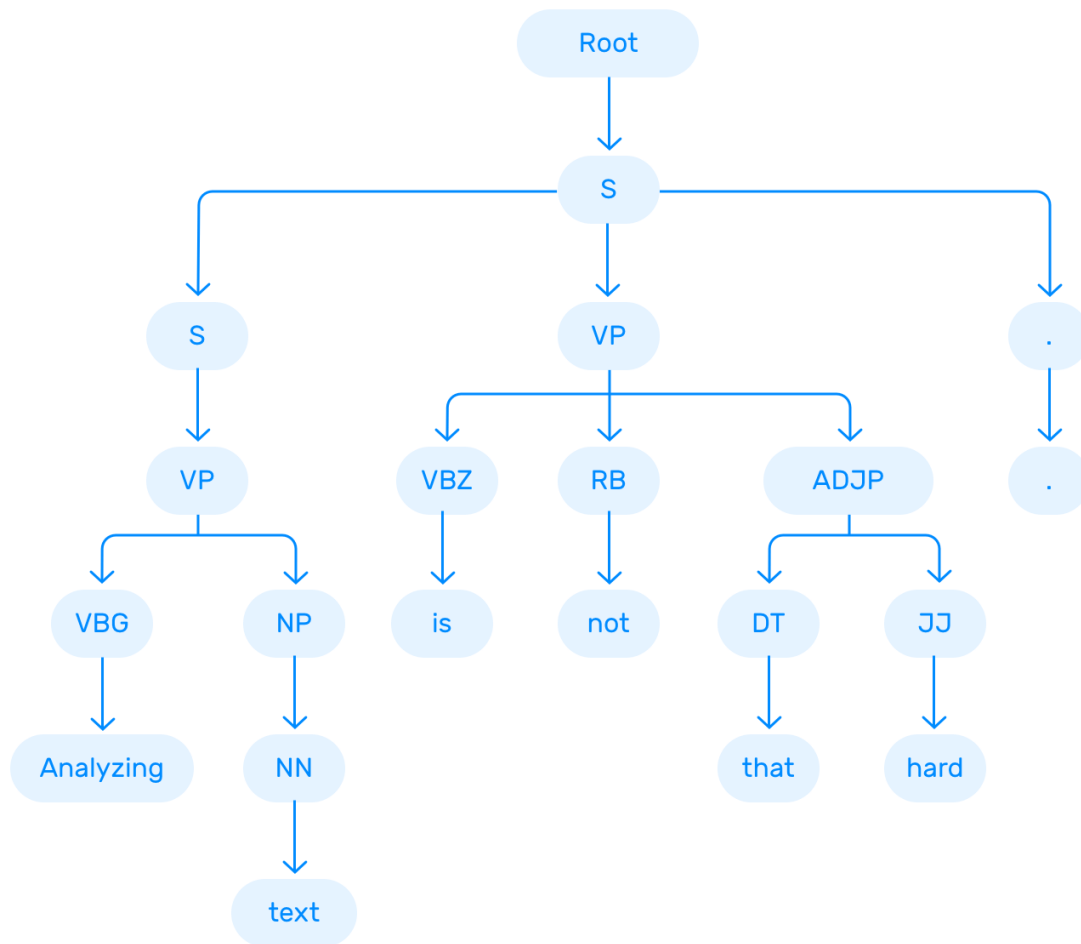
Dependency Parsing

[Dependency grammar](#) refers to the way the words in a sentence are connected. A dependency parser, therefore, analyzes how 'head words' are related and modified by other words too understand the syntactic structure of a sentence:
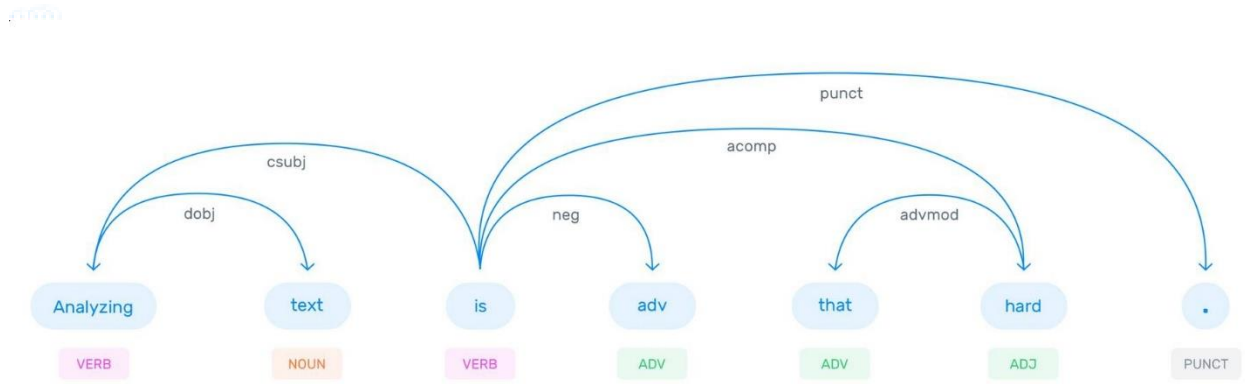
Constituency Parsing

Constituency Parsing aims to visualize the entire syntactic structure of a sentence by identifying [phrase structure grammar](). It consists of using abstract terminal and non-terminal nodes associated to words, as shown in this example:

```
                          Root
                           |
                           v
                           S
         +-----------------+-----------------+
         v                 v                 v
         S                 VP                .
         |       +---------+---------+       |
         v       v         v         v       v
         VP      VBZ       RB       ADJP     .
      +--+--+     |         |      +---+---+
      v     v     v         v      v       v
     VBG    NP    is       not     DT      JJ
      |     |                      |       |
      v     v                      v       v
  Analyzing NN                    that    hard
            |
            v
           text
```

You can try different parsing algorithms and strategies depending on the nature of the text you intend to analyze, and the level of complexity you'd like to achieve.

Lemmatization & Stemming

When we speak or write, we tend to use inflected forms of a word (words in their different grammatical forms). To make these words easier for computers to understand, NLP uses lemmatization and stemming to transform them back to their root form.

The word as it appears in the dictionary – its root form – is called a lemma. For example, the terms *"is, are, am, were, and been,"* are grouped under the lemma *'be.'* So, if we apply this lemmatization to *"African elephants have four nails on their front feet,"* the result will look something like this:

African elephants have four nails on their front feet = "African," "elephant," "have," "4", "nail," "on," "their," "foot"]

This example is useful to see how the lemmatization changes the sentence using its base form (e.g., the word "feet"" was changed to "foot").

When we refer to stemming, the root form of a word is called a stem. Stemming "trims" words, so word stems may not always be semantically correct.

For example, stemming the words "consult," "consultant," "consulting," and "consultants" would result in the root form "consult."

While lemmatization is dictionary-based and chooses the appropriate lemma based on context, stemming operates on single words without considering the context. For example, in the sentence:

"This is better"

The word "better" is transformed into the word "good" by a lemmatizer but is unchanged by stemming. Even though stemmers can lead to less-accurate results, they are easier to build and perform faster than lemmatizers. But lemmatizers are recommended if you're seeking more precise linguistic rules.

Stopword Removal

Removing stop words is an essential step in NLP text processing. It involves filtering out high-frequency words that add little or no semantic value to a sentence, for example, *which, to, at, for, is,* etc.

You can even customize lists of stopwords to include words that you want to ignore.

Let's say you want to classify customer service tickets based on their topics. In this example: *"Hello, I'm having trouble logging in with my new password"*, it may be useful to remove stop words like *"hello"*, *"I"*, *"am"*, *"with"*, *"my"*, so you're left with the words that help you understand the topic of the ticket: *"trouble"*, *"logging in"*, *"new"*, *"password"*.

Word Sense Disambiguation

Depending on their context, words can have different meanings. Take the word *"book"*, for example:

- *You should read this **book**; it's a great novel!*
- *You should **book** the flights as soon as possible.*
- *You should close the **books** by the end of the year.*
- *You should do everything by the **book** to avoid potential complications.*

There are two main techniques that can be used for word sense disambiguation (WSD): *knowledge-based (or dictionary approach)* or *supervised approach*. The first one tries to infer meaning by observing the dictionary definitions of ambiguous terms within a text, while the latter is based on natural language processing algorithms that learn from training data.

Named Entity Recognition (NER)

[Named entity recognition](#) is one of the most popular tasks in semantic analysis and involves extracting entities from within a text. Entities can be names, places, organizations, email addresses, and more.

Relationship extraction, another sub-task of NLP, goes one step further and finds relationships between two nouns. For example, in the phrase *"Susan lives in Los Angeles,"* a person *(Susan)* is related to a place *(Los Angeles)* by the semantic category "lives in."

Text Classification

[Text classification](#) is the process of understanding the meaning of unstructured text and organizing it into predefined categories (tags). One of the most popular text classification tasks is sentiment analysis, which aims to categorize unstructured data by sentiment.

Other classification tasks include intent detection, [topic modeling](#), and language detection.

Challenges of Natural Language Processing

There are many [challenges in Natural language processing](#) but one of the main reasons NLP is difficult is simply because human language is ambiguous.

Even humans struggle to analyze and classify human language correctly.

Take sarcasm, for example. How do you teach a machine to understand an expression that's used to say the opposite of what's true? While humans would easily detect sarcasm in this comment, below, it would be challenging to teach a machine how to interpret this phrase:

"If I had a dollar for every smart thing you say, I'd be poor."

To fully comprehend human language, data scientists need to teach NLP tools to look beyond definitions and word order, to understand context, word ambiguities, and other complex concepts connected to messages. But, they also need to consider other aspects, like culture, background, and gender, when fine-tuning natural language processing models. Sarcasm and humor, for example, can vary greatly from one country to the next.

Natural language processing and powerful machine learning algorithms (often multiple used in collaboration) are improving, and bringing order to the chaos of human language, right down to concepts like sarcasm. We are also starting to see [new trends in NLP](#), so we can expect NLP to revolutionize the way humans and technology collaborate in the near future and beyond.

Natural Language Processing Examples
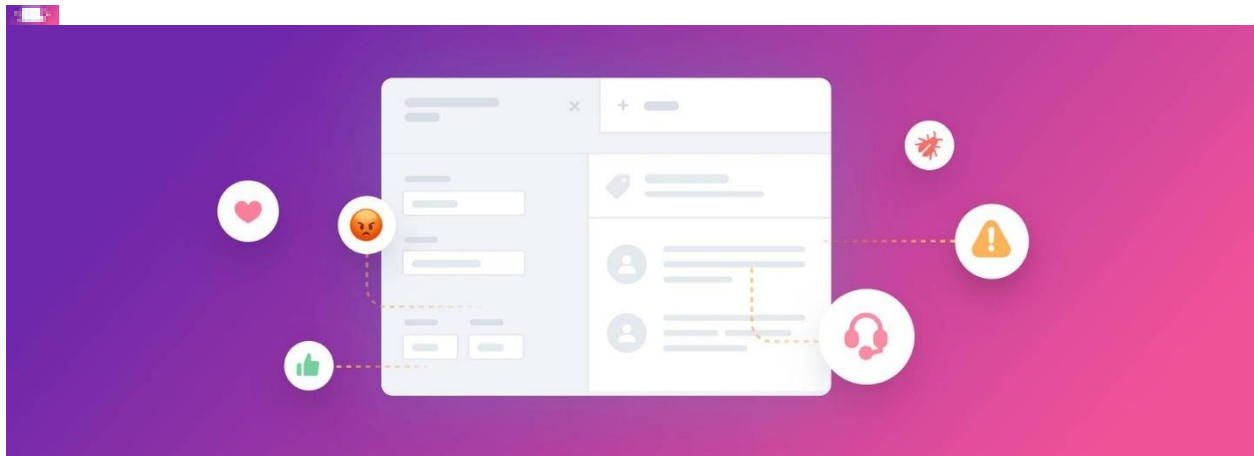
Although natural language processing continues to evolve, there are already many ways in which it is being used today. Most of the time you'll be exposed to natural language processing without even realizing it.

Often, NLP is running in the background of the tools and applications we use everyday, helping businesses improve our experiences. Below, we've highlighted

some of the most common and most powerful uses of natural language processing in everyday life:

11 Common Examples of NLP

1. [Email filters](#)
2. [Virtual assistants, voice assistants, or smart speakers](#)
3. [Online search engines](#)
4. [Predictive text and autocorrect](#)
5. [Monitor brand sentiment on social media](#)
6. [Sorting customer feedback](#)
7. [Automating processes in customer support](#)
8. [Chatbots](#)
9. [Automatic summarization](#)
10. [Machine translation](#)
11. [Natural language generation](#)

Email filters

As mentioned above, email filters are one of the most common and most basic uses of NLP. When they were first introduced, they weren't entirely accurate, but with years of machine learning training on millions of data samples, emails rarely slip into the wrong inbox these days.

Virtual assistants, voice assistants, or smart speakers

The most common being [Apple's Siri](#) and [Amazon's Alexa](#), virtual assistants use NLP machine learning technology to understand and automatically process voice requests. Natural language processing algorithms allow the assistants to be custom-trained by individual users with no additional input, to learn from previous interactions, recall related queries, and connect to other apps.

The use of voice assistants is expected to continue to grow exponentially as they are used to control home security systems, thermostats, lights, and cars – even let you know what you're running low on in the refrigerator.

Online search engines

Whenever you do a simple Google search, you're using NLP machine learning. They use highly trained algorithms that, not only search for related words, but for the intent of the searcher. Results often change on a daily basis, following trending

queries and morphing right along with human language. They even learn to suggest topics and subjects related to your query that you may not have even realized you were interested in.

Predictive text

Every time you type a text on your smartphone, you see NLP in action. You often only have to type a few letters of a word, and the texting app will suggest the correct one for you. And the more you text, the more accurate it becomes, often recognizing commonly used words and names faster than you can type them.

Predictive text, autocorrect, and autocomplete have become so accurate in word processing programs, like MS Word and Google Docs, that they can make us feel like we need to go back to grammar school.

Monitor brand sentiment on social media

[Sentiment analysis](#) is the automated process of classifying opinions in a text as positive, negative, or neutral. It's often used to [monitor sentiments on social media.](#) You can track and analyze sentiment in comments about your overall brand, a product, particular feature, or compare your brand to your competition.

Imagine you've just released a new product and want to detect your customers' initial reactions. Maybe a customer tweeted discontent about your customer service. By tracking sentiment analysis, you can spot these negative comments right away and respond immediately.

Quickly sorting customer feedback

[Text classification](#) is a core NLP task that assigns predefined categories (tags) to a text, based on its content. It's great for organizing qualitative feedback (product reviews, social media conversations, surveys, etc.) into appropriate subjects or department categories.

[Retently](#), a SaaS platform, [used NLP tools to classify NPS responses](#) and gain actionable insights in next to no time:

NPS Feedback Classification

- Onboarding
- Product Performance
- Product UI
- Product UX
- Product Features
- Ease of Use
- Integrations
- Reporting
- Customer Support
- Pricing

Retently discovered the most relevant topics mentioned by customers, and which ones they valued most. Below, you can see that most of the responses referred to "Product Features," followed by "Product UX" and "Customer Support" (the last two topics were mentioned mostly by Promoters).

Response Tags Analysis

| Promoters | Passives | Detractors |

Topic	Promoters	Passives	Detractors
Product Features	50	160	90
Product UX	220	30	20
Customer Support	190	10	20
Integrations	30	60	10
Pricing	60	10	
Ease of Use	40		
Product UI	40		
Product Performance	20		10

Automating processes in customer service

Other interesting applications of NLP revolve around [customer service automation](#). This concept uses AI-based technology to eliminate or reduce routine manual tasks in customer support, saving agents valuable time, and making processes more efficient.

According to the [Zendesk benchmark](#), a tech company receives +2600 support inquiries per month. Receiving large amounts of support tickets from different channels (email, social media, live chat, etc), means companies need to have a strategy in place to categorize each incoming ticket.

Text classification allows companies to [automatically tag incoming customer support tickets](#) according to their topic, language, sentiment, or urgency. Then, based on these tags, they can instantly [route tickets to the most appropriate pool of agents](#).

Uber designed its own [ticket routing workflow](#), which involves tagging tickets by *Country, Language, and Type* (this category includes the sub-tags *Driver-Partner, Questions about Payments, Lost Items, etc*), and following some prioritization rules, like sending requests from new customers (*New Driver-Partners*) are sent to the top of the list.

Chatbots

A chatbot is a computer program that simulates human conversation. Chatbots use NLP to recognize the intent behind a sentence, identify relevant topics and keywords, even emotions, and come up with the best response based on their interpretation of data.

As customers crave fast, personalized, and around-the-clock support experiences, chatbots have become the heroes of customer service strategies. Chatbots reduce customer waiting times by providing immediate responses and especially excel at handling routine queries (which usually represent the highest volume of customer support requests), allowing agents to focus on solving more complex issues. In fact, [chatbots can solve up to 80% of routine customer support tickets](#).

Besides providing customer support, chatbots can be used to recommend products, offer discounts, and make reservations, among many other tasks. In order to do that, most chatbots follow a simple 'if/then' logic (they are programmed to identify intents and associate them with a certain action), or provide a selection of options to choose from.

Automatic summarization

Automatic summarization consists of reducing a text and creating a concise new version that contains its most relevant information. It can be particularly useful to summarize large pieces of unstructured data, such as academic papers.

There are two different ways to use NLP for summarization:

1. To extract the most important information within a text and use it to create a summary **(extraction-based summarization)**
2. Apply deep learning techniques to paraphrase the text and produce sentences that are not present in the original source **(abstraction-based summarization)**.

Automatic summarization can be particularly useful for data entry, where relevant information is extracted from a product description, for example, and automatically entered into a database.

Machine translation

The possibility of translating text and speech to different languages has always been one of the main interests in the NLP field. From the first attempts to translate text from Russian to English in the 1950s to state-of-the-art deep learning neural systems, machine translation (MT) has seen significant improvements but still presents challenges.

Google Translate, Microsoft Translator, and Facebook Translation App are a few of the leading platforms for generic machine translation. In August 2019, Facebook AI [English-to-German machine translation model](#) received first place in the contest held by the Conference of Machine Learning (WMT). The translations obtained by this model were defined by the organizers as "superhuman" and considered highly superior to the ones performed by human experts.

Another interesting development in machine translation has to do with customizable machine translation systems, which are adapted to a specific domain and trained to understand the terminology associated with a particular field, such as medicine, law, and finance. [Lingua Custodia](#), for example, is a machine translation tool dedicated to translating technical financial documents.

Finally, one of the latest innovations in MT is adaptative machine translation, which consists of systems that can learn from corrections in real-time.

Natural language generation

Natural Language Generation (NLG) is a subfield of NLP designed to build computer systems or applications that can automatically produce all kinds of texts in natural language by using a semantic representation as input. Some of the applications of NLG are question answering and [text summarization](#).

In 2019, [artificial intelligence company Open AI released GPT-2](#), a text-generation system that represented a groundbreaking achievement in AI and has taken the NLG field to a whole new level. The system was trained with a massive dataset of 8 million web pages and it's able to generate coherent and high-quality pieces of text (like news articles, stories, or poems), given minimum prompts.

The model performs better when provided with popular topics which have a high representation in the data (such as Brexit, for example), while it offers poorer results when prompted with highly niched or technical content. Still, it's possibilities are only beginning to be explored.

Natural Language Processing with Python

Now that you've gained some insight into the basics of NLP and its current applications in business, you may be wondering how to put NLP into practice.

There are many open-source libraries designed to work with natural language processing. These libraries are free, flexible, and allow you to build a complete and customized NLP solution.

However, building a whole infrastructure from scratch requires years of data science and programming experience or you may have to hire whole teams of engineers.

SaaS tools, on the other hand, are ready-to-use solutions that allow you to incorporate NLP into tools you already use simply and with very little setup. Connecting SaaS tools to your favorite apps through their APIs is easy and only requires a few lines of code. It's an excellent alternative if you don't want to invest time and resources learning about machine learning or NLP.

Take a look at the Build vs. Buy Debate to learn more.

Here's a list of the top NLP tools:

- **MonkeyLearn** is a SaaS platform that lets you build customized natural language processing models to perform tasks like sentiment analysis and keyword extraction. Developers can connect NLP models via the API in Python, while those with no programming skills can upload datasets via the smart interface, or connect to everyday apps like Google Sheets, Excel, Zapier, Zendesk, and more.
- **Natural Language Toolkit (NLTK)** is a suite of libraries for building Python programs that can deal with a wide variety of NLP tasks. It is the most popular Python library for NLP, has a very active community behind it, and is often used for educational purposes. There is a handbook and tutorial for using NLTK, but it's a pretty steep learning curve.
- **SpaCy** is a free open-source library for advanced natural language processing in Python. It has been specifically designed to build NLP applications that can help you understand large volumes of text.
- **TextBlob** is a Python library with a simple interface to perform a variety of NLP tasks. Built on the shoulders of NLTK and another library called Pattern, it is intuitive and user-friendly, which makes it ideal for beginners. Learn more about how to use TextBlob and its features.

Natural Language Processing Tutorial
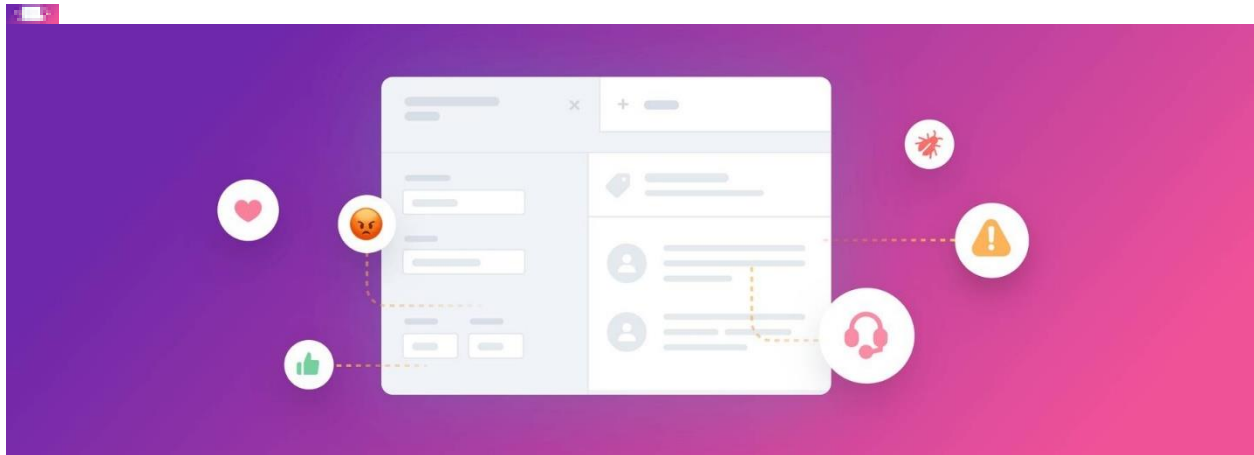
SaaS solutions like MonkeyLearn offer ready-to-use NLP templates for analyzing specific data types. In this tutorial, below, we'll take you through how to perform sentiment analysis combined with keyword extraction, using our customized template.
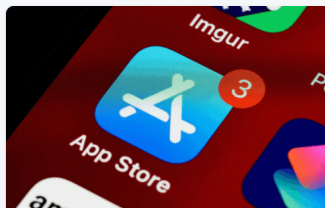
1. Choose Keyword + Sentiment Analysis template

2. Upload your text data

If you don't have a CSV, use our [sample dataset](#).

3. Match the CSV columns to the dashboard fields

In this template, there is only one field: *text*. If you have more than one column in your dataset, choose the column that has the text you would like to analyze.

4. Name your workflow

5. Wait for your data to import
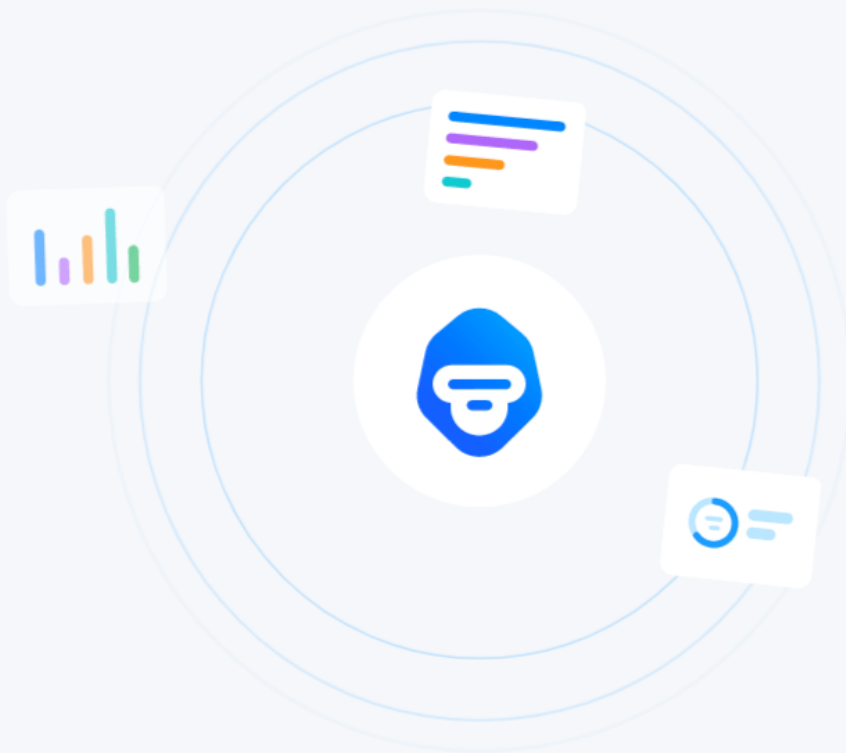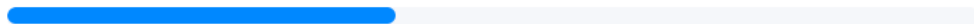
6. Explore your dashboard!

You can:

- Filter by sentiment or keyword.
- Share via email with other coworkers.

Final Words on Natural Language Processing

Natural language processing is transforming the way we analyze and interact with language-based data by training machines to make sense of text and speech, and perform automated tasks like translation, summarization, classification, and extraction.

Not long ago, the idea of computers capable of understanding human language seemed impossible. However, in a relatively short time — and fueled by research and developments in linguistics, computer science, and machine learning — NLP has become one of the most promising and fastest-growing fields within AI.

As technology advances, NLP is becoming more accessible. Thanks to plug-and-play NLP-based software like MonkeyLearn, it's becoming easier for companies to create customized solutions that help automate processes and better