There are many approaches to automatic text classification, which can be grouped into three different types of systems:

* Rule-based systems
* Machine Learning based systems
* Hybrid systems

1. Rule-based Systems

Rule-based approaches classify text into organized groups by using a set of handcrafted linguistic rules. These rules instruct the system to use semantically relevant elements of a text to identify relevant categories based on its content. Each rule consists of an antecedent or pattern and a predicted category.

Say that you want to classify news articles into 2 groups, namely, Sports and Politics. First, you’ll need to define two lists of words that characterize each group (e.g. words related to sports such as football, basketball, LeBron James, etc., and words related to politics such as Donald Trump, Hillary Clinton, Putin, etc.). Next, when you want to classify a new incoming text, you’ll need to count the number of sport-related words that appear in the text and do the same for politics-related words. If the number of sport-related word appearances is greater than the number of politics-related word count, then the text is classified as sports and vice versa.

Notes :

Rule-based systems are human comprehensible and can be improved over time.

Difficult , time consumning, complex and difficult to maintain

### Machine Learning Based Systems

Instead of relying on manually crafted rules, text classification with machine learning learns to make classifications based on past observations.

By using pre-labeled examples as training data, a machine learning algorithm can learn the different associations between pieces of text and that a particular output (i.e. tags) is expected for a particular input (i.e. text).

(training Data )

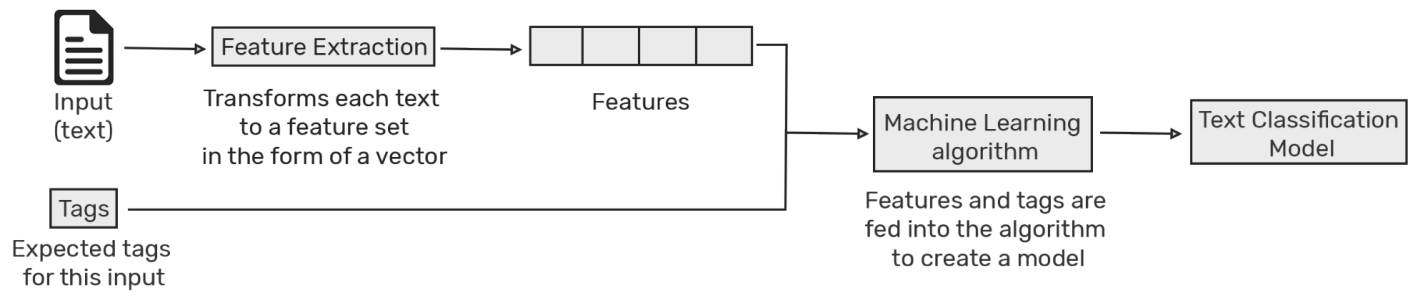
Steps:

1. Feature Extraction :

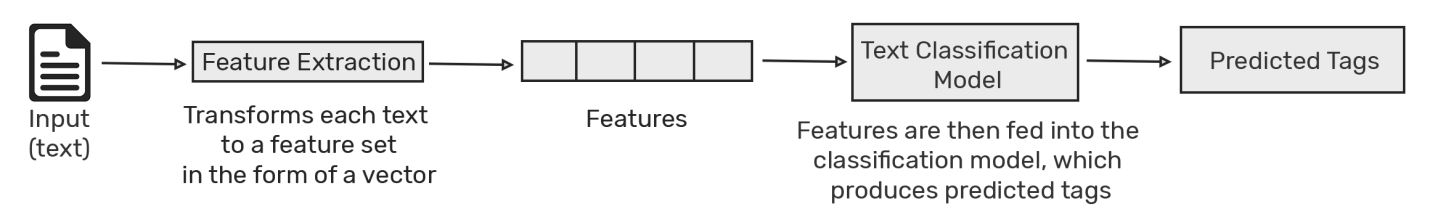
a method is used to [transform each text into a numerical representation](https://monkeylearn.com/blog/beginners-guide-text-vectorization/) in the form of a vector. One of the most frequently used approaches is [bag of words](https://machinelearningmastery.com/gentle-introduction-bag-words-model/), where a vector represents the frequency of a word in a predefined dictionary of words.

For example, if we have defined our dictionary to have the following words {This, is, the, not, awesome, bad, basketball}, and we wanted to vectorize the text *“This is awesome”*, we would have the following vector representation of that text: (1, 1, 0, 0, 1, 0, 0).

Training



Testing



Text classification with machine learning is usually much more accurate than human-crafted rule systems, especially on complex classification tasks. Also, classifiers with machine learning are easier to maintain and you can always tag new examples to learn new tasks.

#### Text Classification Algorithms

the naive bayes family of algorithms, support vector machines, and deep learning.

* 1. Naïve bayes
     + [Naive Bayes](https://monkeylearn.com/text-classification-naive-bayes/) is a family of statistical algorithms we can make use of when doing text classification. One of the members of that family is Multinomial Naive Bayes (MNB). One of its main advantages is that you can get really good results when data available is not much (~ a couple of thousand tagged samples) and computational resources are scarce.
     + This algorithm is build on bayes theory , This means that any vector that represents a text will have to contain information about the probabilities of appearance of the words of the text within the texts of a given category so that the algorithm can compute the likelihood of that text’s belonging to the category
  2. Support vector machines

[Support Vector Machines](https://monkeylearn.com/text-classification-support-vector-machines-svm/) (SVM) is just one out of many algorithms we can choose from when doing text classification. Like naive bayes, SVM doesn’t need much training data to start providing accurate results. Although it needs more computational resources than Naive Bayes, SVM can achieve more accurate results.

##### **Deep Learning**

[Deep learning](https://medium.com/dair-ai/deep-learning-for-nlp-an-overview-of-recent-trends-d0d8f40a776d) is a set of algorithms and techniques inspired by how the human brain works. Text classification has benefited from the recent resurgence of deep learning architectures due to their potential to reach high accuracy with less need of engineered features. The two main deep learning architectures used in text classification are [Convolutional Neural Networks](https://machinelearningmastery.com/crash-course-convolutional-neural-networks/) (CNN) and [Recurrent Neural Networks](https://machinelearningmastery.com/crash-course-recurrent-neural-networks-deep-learning/) (RNN).

Deep learning algorithms such as [Word2Vec](https://code.google.com/archive/p/word2vec/) or [GloVe](https://nlp.stanford.edu/projects/glove/) are also used in order to obtain better vector representations for words and improve the accuracy of classifiers trained with traditional machine learning algorithms.

DrawBack : Data

### 3-Hybrid Systems

Hybrids systems combine a base classifier trained with machine learning and a rule-based system, which is used to further improve the results. These hybrid systems can be easily fine-tuned by adding specific rules for those conflicting tags that haven’t been correctly modeled by the base classifier

### 4-Metrics and Evaluation

[Cross-validation](https://en.wikipedia.org/wiki/Cross-validation_(statistics)) is a common method to evaluate the performance of a text classifier. It consists in splitting the training dataset randomly into equal-length sets of examples (e.g. 4 sets with 25% of the data). For each set, a text classifier is trained with the remaining samples (e.g. 75% of the samples). Next, the classifiers make predictions on their respective sets and the results are compared against the human-annotated tags. This allows finding when a prediction was right (true positives and true negatives) and when it made a mistake (false positives, false negatives).

## Text Classification Applications

Some of the most well-known examples of text classification include sentiment analysis, topic labeling, language detection, and intent detection.

1. Sentiment analysis

#### Topic Labeling

Another common example of text classification is topic labeling, that is, understanding what a given text is talking about. It’s often used for structuring and organizing data such as organizing customer feedback by its topic or organizing news articles according to their subject.

#### Language Detection

Language detection is another great example of text classification, that is, the process of classifying incoming text according to its language.

## Text Classification Resources

. Building your first text classifier can be simple and straightforward. You just need two things:

1. A dataset to provide examples for training the classifier.
2. A tool for generating and consuming the classifier.
3. Dataset :
   * 1. A text classifier is worthless without accurate training data to power it. Just like humans, machine learning algorithms can make predictions by learning from previous examples. By telling the algorithm that you expect a specific set of tags as output for a particular text, it can learn to recognize patterns in text, like the sentiment expressed by a tweet, or the topic mentioned in a customer review.
     2. An accurate classifier depends entirely on getting the right training data, which means gathering examples that best represent the outcomes you want to predict. Say you want to predict the intent from chat conversations; you’ll need to identify and gather chat conversations that represent the different intents you want to predict. If you train your model with another type of data, the classifier will provide poor results.

HOW to Get the right data :

There are links to dataset in the article

### Tools

Alright. Now that you have training data, it's time to feed it to a machine learning algorithm and create a text classifier.

So, how do we do this?

Luckily, many resources can help you during the different phases of the process, i.e. transforming texts into vectors, training a machine learning algorithm, and using a model to make predictions. Broadly speaking, these tools can be classified into two different categories:

* Open Source libraries
* SaaS APIs

1. Open Source lib. :

##### Text Classification with Python

[Scikit-learn](http://scikit-learn.org/), [NLTK](https://www.nltk.org/) is a popular library focused on [Natural Language Processing](https://monkeylearn.com/blog/definitive-guide-natural-language-processing/) (NLP) ,  [SpaCy](https://spacy.io/),

deep learning libraries like Keras, TensorFlow, and PyTorch

[Keras](https://keras.io/) is probably the best starting point as its designed to simplify the creation of recurrent neural networks (RNNs) and convolutional neural networks (CNNs). [TensorFlow](https://www.tensorflow.org/) is the most popular open source library for implementing deep learning algorithms. Developed by Google and used by companies such as Dropbox, eBay, and Intel, this library is optimized for setting up, training, and deploying artificial neural networks with massive datasets. Although it’s harder to master than Keras, it’s the undisputed leader in the deep learning space. A reliable alternative to TensorFlow is [PyTorch](https://pytorch.org/), an extensive deep learning library primarily developed by Facebook and backed by Twitter, Nvidia, Salesforce, Stanford University, University of Oxford, and Uber

### Text Classification Tutorial