DIGITAL CULTURE
ITMO UNIVERSITY

Text Classification

High School of Digital Culture
ITMO University
dc@itmo.ru

Contents

1 Formulation of the Classification Problem

One of the common tasks in computational linguistics and Natural Language Processing (NLP) is text classification. In general, text classification is the task of assigning categories from a finite set to the analyzed text based on its content.

But before we dig deep into this and consider respective machine-learning methods, let's take a look at the real-life examples of text classification.

1.1 Spam Detection

Every day we receive dozens of emails, but only some of them are relevant, because, along with important messages, we also get newsletters, phishing emails, and chain letters.

Most mail servers have built-in spam filters that process incoming emails to prevent junk from reaching a user's inbox. Instead, the filter moves junk to the Spam folder. So, how does it work?

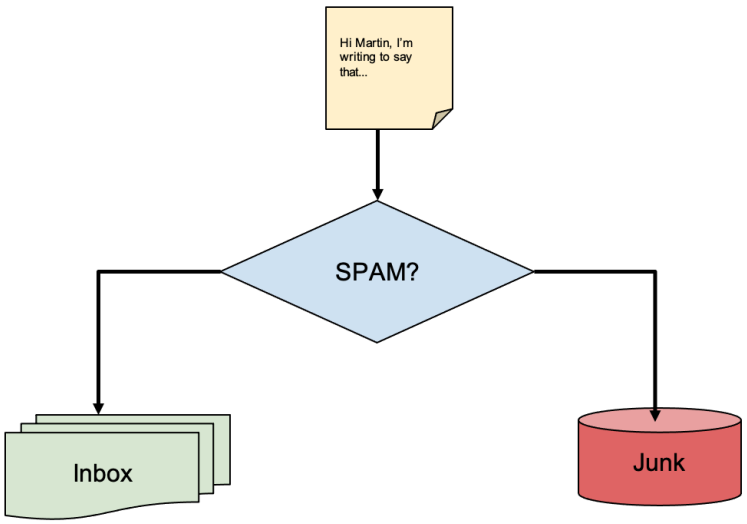

Figure 1: SPAM or HAM?

Assume that a user receives such an email. An advanced user can easily recognize spam. But how? Well, this email looks suspicious. First of all, it contains multiple references to prize or money, for example, 'Cash Certificate' or '$500 Gift Certificate'. Secondly, the email urges the user to get the money fast and stresses the phrase 'get your money' multiple times. On top of that, the recipient may notice the absence of the direct personal form of address ('Congratulations!'), a suspicious link, strange sender's address, as well as the sending time ('11:00'). All of these things combined allow classifying the email as junk.

1.2 Sentiment Analysis

Another common classification problem is called sentiment analysis. Sentiment analysis is the interpretation and classification of a general tone of the author's statement (positive, negative, or neutral). Such texts include movie reviews, customer reviews, or even comments on a website.

For example, the slide shows the mobile phone reviews. It's clear that the first review was written by a satisfied customer because it contains such words and phrases as 'great', 'awesome', 'high performance', 'latest firmware', 'at a high level'. And there's no doubt that the second review is negative. The customer expresses a desire to throw the purchase out of the window and doesn't recommend people to buy the phone. Classification of such reviews can be done automatically.

1.3 News Topic Analysis

The next text-classification technique we are going to discuss is called news topic analysis. Let's look at the titles. Based on the context, a person can easily tell what the article is about (sports or cryptocurrency news). Keywords are the easiest thing to use, for example, figure skating, fans, or altcoins. Keywords are not limited to words. They can include country names, brands, or personalities relevant to the topic (for example, Evgenia Tarasova or Craig Wright). This task is called named entity recognition (NER), and it deserves a separate discussion.

1.4 Examples Of Classification Problems

What else can a classifier do? It turns out that the number of tasks that classification can solve is very high. In general, classification is a process of text categorization based on the content. We can use it to identify the author of a text (for example, to find out who wrote the comment) or to predict an author's age and gender. We can also classify texts according to their language or encoding. Search engines use classification to limit search results, find targeted ads, and detect bot messages in online discussions.

1.5 Formal Description Of A Problem

Well, let's formally describe a classification problem in machine learning. One thing to remember is that text classification and text clustering are not the same. Text classification uses predefined document categories (for example, news topics or sentiment of reviews). Clustering, however, is performed when no information about possible text categories and their number is available.

Let's formally describe this task. There is a set of documents D that includes all texts in the sample, and there is a set of categories C that includes all possible categories assigned to documents. Moreover, there is an unknown objective function F that predicts text categories. The task is to use the pro-

vided data and construct such a classifier F' that is close to F and that can predict categories of text which the model hasn't observed during training.

It's important to note that we have no information about the text except that extracted from the texts, for example, specific words or their frequency.

Classification is a supervised learning task. It means that a subsample of texts with predefined classes is used to build a classifier. Such subsamples are used to train the classifier and define the parameters that provide the best result. The dataset is split into a training sample and a test sample. The system uses the training sample to develop rules of document classification and then applies them to the test sample to check the splitting accuracy.

1.6 Single-Label Classification And Multi-Label Classification

There are two types of classification. They are called single-label classification (or just 'classification') and multi-label classification. Single-label classification is a problem of assigning each document exactly one label.

Single-label classification can be multi-class or binary. For example, the task is to identify the author of a text from the list of different authors.

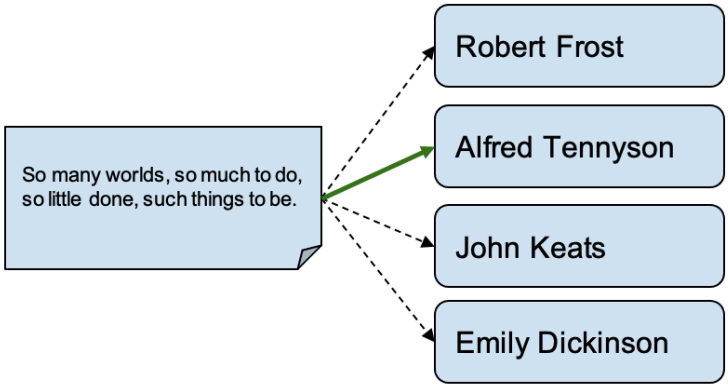

Figure 2: Multi-class classification

Another example of multi-class classification is assigning categories to texts to be placed in different sections of the website. Binary classification deals with two categories that are distinct from one another, for example, spam or non-spam, positive reviews or negative reviews.

Multi-label classification assumes that one text can have several different tags at the same time. For example, based on the defined topic, the website can offer the user personalized content. The article about airline tickets can present interest for travelers and those who read healthcare-related news.

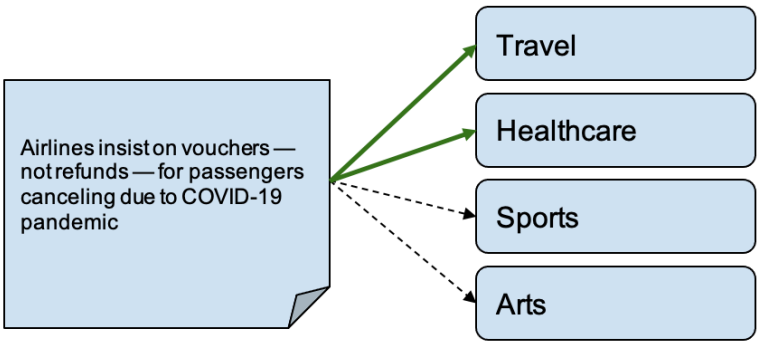

Figure 3: Multi-class classification

2 Classification Methods

2.1 The Main Steps In Solving A Problem

The main steps in solving a problem are document preprocessing and indexing, dimensionality reduction of the feature space, building and training the classifier, as well as quality evaluation.

We've already discussed text preprocessing, and, as you know, it includes tokenization, stop-word removal (too frequent words, prepositions, articles, and so on), and normalization. Such preprocessing helps to reduce dimensionality to some extent.

2.2 Document Indexing

The process of creating a numerical model for the conversion of a text into a preprocessed form is often called document indexing.

We are already familiar with one of such methods. It's called a bag of words. A vector size equals a dictionary size, and nonzero values are term frequencies. It's one of the simplest ways to converta text into numbers. However, the information about word order is lost in this case.

The second way of document indexing is called n-gram. In this case, we count not only standalone words and symbols but also pairs, triplets, and so on. Unlike a bag of words, n-gram allows taking the word order into account and differentiate 'will like this' from 'will not like this'. However, we encounter another problem. When we consider all n-grams, the resulting number of vectors can be too high. In such a case, our advice is to remove too frequent or rare n-grams and carefully choose N.

Some more advanced text processing methods use neural networks and prediction-based models of distributional semantics. A family of word2vec methods is a good example. word2vec represents each word as a vector that encodes the meaning based on the context words. By combining such vectors in different

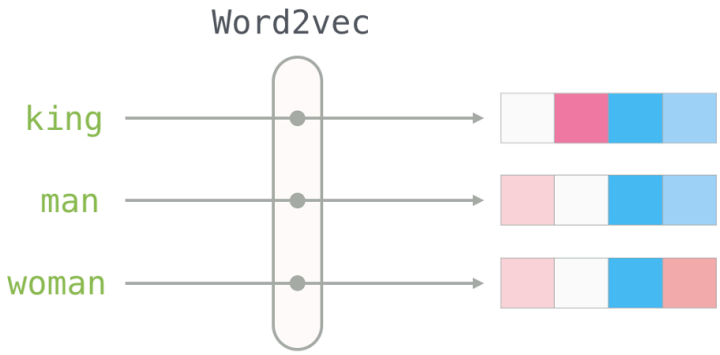

Figure 4: Word2vec

ways, we can obtain a text representation that can easily represent synonyms and implicitly re-uses language information derived from another, maybe even larger, text collection. It usually provides better results. Distributional semantics has become particularly important recently due to neural network breakthroughs, but it's a totally different story.

2.3 Classification Algorithms

There are many classification methods, including linear (for example, logistic regression), probabilistic (such as naive Bayes), metric (k-nearest neighbors), logical (decision trees), as well as neural network methods. Instead of considering each method one by one, we will focus only on some of them (logistic regression, support vector machine, and a naive Bayes classifier).

2.4 Linear Algorithms

Let's consider linear algorithms. Such algorithms assume that all objects in the training sample are points in a multidimensional space. Our task is to construct such a surface that will separate points of one class from those of another. A linear algorithm's goal is to find such separating hyperplane. In a two-dimensional case, a hyperplane is a straight line.

A linear plane is defined by the equation of the following form: a dot product of w and x minus b equals zero, where x are features, and w and b are tunable parameters. Next, we need to figure out, where the point that corresponds to the object with respect to the linear space is located in the multidimensional space. To do that, we need to look at the sign of the expression on the left of the equality. This is how classification is performed. Thus, the goal of linear methods is to find coefficients w and the constant b that set the hyperplane.

The advantages of linear algorithms are fast learning, coefficient interpretability, and high performance at prediction stage. However, their major disadvantage is low accuracy when the sample is linearly inseparable and the

relationship between features and predicted values is not a simple one.

2.5 Logistic Regression

Another linear algorithm is called logistic regression. It is a special case of a generalized linear regression. The good thing about this method is that it predicts not only the 'Yes' or 'No' type response but also the probability of assigning it to the class. In logistic regression, weights are adjusted using gradient descent. It minimizes errors given a training sample. Sometimes, despite that the number of errors on a training sample is small, the algorithm accuracy is low on a test sample. The reason is overfitting, which means that the trained model corresponds too closely to the provided training data. Due to that, the algorithm cannot demonstrate the same or even reasonable performance on new data. To avoid overfitting, a term is added to the minimized function. The term depends only on the vector W. This term called a regularizer helps to prevent overfitting by introducing a penalty for large W values, which, roughly speaking, are a certain sign that the model relies on certain features too much.

The advantage of logistic regression is that we obtain the estimated probability of assigning an object to a particular class. Moreover, the algorithm is relatively easy to code. The disadvantage of logistic regression is a complicated interpretation of the algorithm and non-robustness to input-data outliers.

2.6 Support Vector Machine

The next algorithm we are going to discuss is called Support Vector Machine (SVM). Support Vector Machine is a classification approach that was designed to find a hyperplane that separates classes in training data in the best way. The algorithm key idea is a search for such points on the graph that are the closest to the separating line. These points are called support vectors. After that, the algorithm calculates the distance between the support vectors and the separating plane. This distance is called a margin. The goal of the algorithm is to maximize the margin distance. The hyperplane is selected as the best one if it has the largest margin.

SVM has its pros and cons. One of the pros is a relatively high accuracy on small datasets. However, SVM's drawbacks are also important to remember. Firstly, it is a complicated interpretation of the algorithm parameters. Secondly, the algorithm is not robust to outliers in input data.

SVM was popular once. Nowadays, you can find many of its extensions that overcome the discussed disadvantages. The kernel trick deserves special attention. It allows using kernel functions for support vector machines.

2.7 Naive Bayes

The last method we will cover today is called naive Bayes. When the text is available, the task boils down to finding the conditional probability that the text belongs to a particular category given that the text has the selected features. For example, what is the probability that the email containing the word 'prize' is junk? Such conditional probabilities are hard to calculate directly. That's why the Bayes' formula is used. Since the denominator doesn't depend on the class Y used for optimization, we will disregard it. The formula for the probability of text classification is reduced to a simpler formula using the conditional independence hypothesis. We assume that all features(words in our case) don't depend on each other. The fact that the text contains a particular token doesn't affect the probability of the occurrence of another token. For example, we assume that the word 'prize' in an email doesn't affect the probability that the word 'money' will be in the email. Thus, we obtain the following formula. The variables are adjusted according to the training sample that consists of the labeled texts. The probability of the class Y is defined by the relative frequency in the training sample, and the probability of the ith token is modeled in different ways. In the simplest case, the probability is considered to be equal to the relative frequency of the ith token in the class Y.

The advantage of the algorithm is that the algorithm parameters are calculated using a small training dataset and that no optimization is carried out, just counting the simple statistics. The algorithm also demonstrates high performance. Moreover, it is not sensitive to the size of the training sample and resistant to overfitting. The disadvantages are low classification accuracy and the absence of ways to consider that features combinations affect the result.

3 Classification Quality Evaluation

3.1 Metrics

Let's consider a binary-classification example and discuss approaches to classification quality evaluation. The easiest way to do it is to calculate accuracy. In this case, we consider the ratio of correctly classified classes to the total number of objects. This easy-to-understand metric has disadvantages.

The first problem occurs when a dataset is not balanced. It relates to such cases when, for example, only 10% in the sample is junk. Thus, 90 percent accuracy is given by the constant value (the prediction that an email is not spam). In this case, accuracy, being the only number for quality assessment, is not informative.

The second problem is the inability to consider different types of errors.

Suppose one is interested in detecting spam, and the cost of misclassification when an important email is moved to junk is very high. There can be the opposite case when spam is not detected and appears in the inbox. Also bad but not critical. We cannot separate such cases when calculating accuracy only.

That's why two different metrics could and should be considered. They are called precision and recall. There are four types of predictions. The first one is true positives (TP). It is the desired class of correctly classified objects. For example, we wanted to find spam, and it was detected. The second one is false negatives (FN), which means that spam was not detected and appeared in the inbox. The third case is false positives (FP). For example, an ordinary email was considered spam. True negatives (TN) are correct predictions, for example, when a non-spam email was not classified as spam.

Precision is a ratio of correctly classified emails to all emails classified as spam. This metric reflects false alarms of the algorithm. recall, in its turn, shows how large is the percentage of actual spam emails the model has retrieved from the entire sample. Such metrics are more useful for real-life tasks than a simple calculation of prediction accuracy.

These two measures are sometimes used together to provide a single measurement called the F1 Score. The F1 Score is the harmonic mean of precision and recall. When precision and recall approach zero, it also approaches zero.

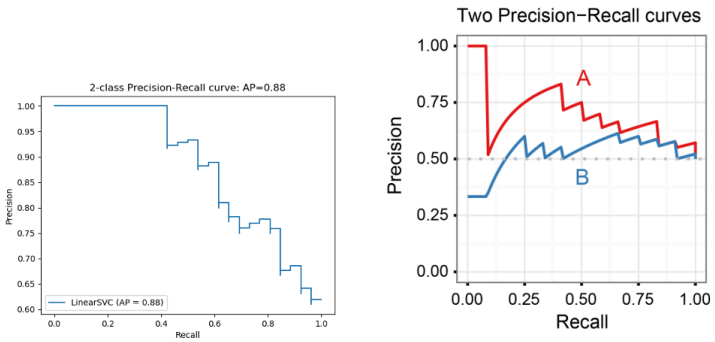

Figure 5: Metrics

The choice of the model often depends on which of the measures is most important (precision or recall). A model that predicts 'Yes' with no confidence will have high recall and low precision, and a model that confidently makes predictions will have low recall and high precision.

Multiclass classification allows calculating all the considered quality estimators for each class. It means that, for one class, we merge the remaining classes into one and consider predictions as a binary classification problem. Thus, we see how well the classification is performed for each class in terms of precision and recall. There are other ways of classification evaluation that we

	precision	recall	f1-score	support
class0	0.888	0.877	0.882	308
class1	0.958	0.535	0.687	43
class2	0.712	0.806	0.756	175
accuracy			0.825	526
macro avg	0.853	0.739	0.775	526
weighted avg	0.835	0.825	0.824	526

Figure 6: Classification results

can use in natural language processing. For more information about classification methods, including those applied to texts, please refer to the respective textbooks and machine learning courses.