### **To solve the problem of multiclass classification of news statements the next steps have been taken:**

### **1. Data Cleaning and Preprocessing**

* **Removal of Duplicates**: Duplicate entries in the dataset are identified and removed to ensure data quality and avoid redundancy.
* **Text Preprocessing**:
  + Conversion to lowercase to standardize text.
  + Tokenization and lemmatization using **spaCy**'s ru\_core\_news\_sm language model for russian language.
  + Removal of stopwords and non-alphabetic characters to focus on meaningful tokens.

### **2. Data Splitting**

* The dataset is split into training and testing subsets using **Scikit-learn's** train\_test\_split.

### **3. Embedding Representation**

* **Pre-trained Embeddings**: pre-trained word embeddings are used to represent the text numerically in a vector space. These embeddings capture semantic similarity between words. We use RuBERT, which is a layered language space transformation model developed for the russian language. This is the BERT model (Bidirectional Encoder Representations from Transformers), which is trained on a large corpus of russian-language literature.
* **Data Saving:** Additionally, the preprocessed data is saved and downloaded to the local machine for further use

### **4. Model Creation and Training**

After text preprocessing was performed, vectorized text data can be used for model training. Due to some data characteristics such as high dimensionality, class imbalance and specifics of task performed, neural network model was chosen over classic machine learning model, because:

* their performance improves as the size of the training data grows(we expect that dataset will become larger, due to additional data collection).
* can easily handle multi-class classification tasks using softmax activation in the output layer
* excellent at learning non-linear models through layers of neurons and activation functions.

**Neural network model created** using TensorFlow/Keras. It consists of:

* Input layer with 767 features.
* Two hidden layers with 128 and 64 units using ReLU activation.
* Output layer with the number of classes in the target variable using softmax activation.

Compiled with the Adam optimizer and cross-entropy loss.

**The model is trained** over 10 epochs with a batch size of 32. The class\_weight parameter adjusts the loss computation for each class based on the weights. Class weights are calculated to address imbalanced data, ensuring the model treats each class equally during training.

The same as preprocessed data, **the trained model is saved** for later use.

### **5. Evaluation Metrics**

* Metrics such as **accuracy**, **precision**, **recall**, and **F1-score** are calculated using Scikit-learn's tools to evaluate the classifier's performance. As we face Class Imbalance Problem, the key metrics are F1-score and precision/recall for each category
* A **confusion matrix** is generated and visualized for detailed analysis of predictions versus true labels.