You have to develop an algorithm for sentiment classification in tweets and compare your results with participants of the competition **SentiRuEval-2016** http://www.dialog-21.ru/media/3410/loukachevitchnvrubtsovayv.pdf

Data are available in the Google Drive: https://goo.gl/VpnPo7 This task divides by two subtasks: processing of tweets about banks and about telecommunication companies. Training set for banks is the file bank_train_2016.xml, and test set for banks is the file banks_test_etalon_2016.xml. Correspondingly, for telecoms the file ttk_train_2016.xml contains a training set, and the file ttk_test_etalon.xl contains a test set.

Each tweet can contain one of three sentiments:

1) positive sentiment, for example, Эм. Почта фиг с ней, но вот Сбер и РЖД пусть будут, а? Они ж исправляются в последнее время. Особенно Сбер;

2) negative sentiment, for example, многа ДМ) спойлер - сбербанк говно,но мисшн комплитд;

3) neutral sentiment, for example, Я просто зашла в Альфа банк и увидела этот журнал)Мило.

Accordingly, you will need to create a system that would be able to correctly determine the sentiment of the tweets from the test set after its training on training data. Such task relates to a **three-class classification problem**, but you have to evaluate the final quality on the test set using only two classes - positive and negative emotions, without taking into account neutral ones. We will have two quality criteria: **F1-measure with macro averaging** and **F1-measure with micro-averaging**. The task has more detailed description in this article: http://www.dialog-21.ru/media/3410/loukachevitchnvrubtsovayv.pdf

All files contain data in the **XML format**. These data are structured as a table, each row of which corresponds to a training / test sample, and the column is one of the characteristics of this sample (ID, date, text, etc.). We need, firstly, a column with text, i.e. what is inside <column name = "text"> some text </ column>, and secondly, "supervisor's instruction", i.e. tag about the emotional fullness (sentiment) of tweet. In the xml-file for banks, this is 1, 0 or -1 for one of eight banks (these are the sberbank, vtb, etc.) fields, and we evaluate the sentiment of the tweet as a whole, not taking into account about which bank is being talked about. In the xml-file for telecoms, this is also a mark of 1, 0 or -1, but for one of the six telecommunication companies (the fields tele2, rostelecom, etc.). The procedure of parsing all these files using the Python language will not be a problem for you. Better not try to write your own parser, but use the Python library lxml https://lxml.de/.

After preparing data you have to create three deep learning algorithms for solving this task

1. Character-level convolutional neural network based on representation of texts as a character sequences. You can be guided by this paper: **"Character-level Convolutional Networks for Text Classification"** https://goo.gl/fkYCZd I recommend you to use the Keras library: https://keras.io for developing of neural network, but you can use any another library, such as Tensorflow or PyTorch.

2. Word-level convolutional neural network, based on representation of texts as a word sequences, like it is described in Yoon Kim's paper **"Convolutional Neural Networks for Sentence Classification"** https://goo.gl/GsqyMn. I recommend you to not use simple Word2Vec embeddings, because they

are not suit for such inflexional language as Russian. Instead of the Word2Vec you can use the **FastText embeddings** based on subword representation. Pretrained FastText models for Russian can be available in website of the RusVectores project https://rusvectores.org/ru/models/ or in "native" FastText website https://fasttext.cc/docs/en/crawl-vectors.html (see *.bin files). For working with the FastText in Python you have to use special **Gensim library**:
https://radimrehurek.com/gensim/models/fasttext.html and
https://radimrehurek.com/gensim/models/_fasttext_bin.html

3. Finally, the third part is almost the same as the second, but you should not use pretrained word2vec models, but *you will train them yourself on a large unlabeled tweet corpus*, compiled by Julia Rubtsova http://study.mokoron.com/ (you should use the largest unlabeled corpus of 17,639,674 tweets https://www.dropbox.com/s/9egqjszeicki4ho/db.sql, since it is on it that your FastText will learn best)