Natural Language Processing Course Project

First Defense

Natural Language Processing 2019/20, Faculty of Computer and Information Science, University of Ljubljana

Jernej Vivod
vivod.jernej@gmail.com

I. Introduction

Text classification is the process of assigning labels to text according to the information extracted by analysing it. It is a typical task in the field of Natural language processing with broad applications such as sentiment analysis, spam detection, intent detection among many others. Development in this field has been motivated by the ever increasing number of available unstructured data in the form of text due to the rise of digitally-based communication.

The goal of our project is to develop and implement a system that can serve as a means to automate or augment real-time discussion moderation using text classification methods. The developed system will be used to analyse the relevance of posts of elementary school-aged children participating in a discussion concerning their mandatory reading. The system will be developed using the data collected during book discussions collected during studies performed as part of the IMapBook project. The system will be able to help inform teachers about the quality of the ongoing discussion and possibly alert them to intervene.

II. Existing Methods

Text classification methods can be broadly divided into classical machine learning-based and deep learning-based with the latter currently being the topic of much research and the source of continuously improving state-of-the-art models [1]. Still, traditional machine learning-based models can function as a good baseline and sanity check for more complex models and are therefore still worth studying. Unlike with deep learning models that often take documents as they are, classical machine learning methods require the text data to be processed and converted into useful features before it can be used. Special characters, punctuations and contractions are removed and tokenization is performed. Stemming, the process of converting words to their base forms, is also performed to improve the density of training data and to reduce the size of the dictionary. Lemmatization also is performed for the same purpose. Statistical machine learning algorithms require the text to be encoded by a number of features which represent an abstract notion of the text's characteristics from which the algorithm can deduce information. There are many ways to extract features from preprocessed text. We can charactarise the text by the frequencies of words appearing in it, also called the Bag of words approach. We can further enhance this approach by taking into account only the words which are deemed significant using the so called term-frequency and inverse document-term frequency features. Shallow two-layer neural networks can also be used to extract features from a large corpus of text. This group of models is called *word2vec*. Extracted features are then fed into typical classifiers (Random forest classifier, support vector machines, etc.) which are then trained using a training set consisting of documents with known annotations/labels. The trained classifier can then be used to classify new documents [2].

Deep learning models used for text classification can take the form of convolutional neural networks that take work on an image-like representation of text documents [3], long short-term memory neural networks, which are especially suited for context-sensitive data analysis such as text, as they are capable of learning long-term dependencies. The long short-term recurrent neural network tries to remember all the past knowledge that the network has seen so far and to forget irrelevant data. This is achieved by introducing different activation layers called gates for specific purposes [4] and more recently transformers, which are highly parallelizable models that use a special attention mechanism using a set of encodings to incorporate context into a sequence of text [5].

III. Initial approach approach

The core part of the system used to analyse the book discussions will consist of a classification method for which we will use and evaluate the current state-of-the-art methods for this domain. We will implement a modular and comprehensive pipeline for evaluating the methods and visualising the results in a human-readable format, which will serve as a proof of concept and demonstration of the functionality. We will also implement and test more traditional approaches that will serve as a baseline for the more advanced methods.

We will use the Python programming languages since it allows for fast prototyping and supports interfacing with most of the well known deep-learning as well as traditional machine learning libraries. Due to previous experience, we will use the Keras library [6] to implement the deep-learning models and the Scikit-learn library [7] for classical machine learning algorithms. We will also use the NLTK (Natural Language Toolkit) library [8] for text preprocessing and feature extraction.

We will make all developed code publicly available and describe the results in a detailed written report.

References

[1] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, D. Brown, L. Id, and Barnes, "Text classification algorithms: A survey," *Information (Switzerland)*, vol. 10, 04 2019.

[2] I. K. in M. Robnik-Šikonja, *Inteligentni sistemi.* Založba FE in FRI, 2010. [Online]. Available: https://books.google.si/books?id=LxZDMwEACAAJ

[3] A. Jacovi, O. S. Shalom, and Y. Goldberg, "Understanding convolutional neural networks for text classification," 2018.

[4] J. Nowak, A. Taspinar, and R. Scherer, "Lstm recurrent neural networks for short text and sentiment classification," 05 2017, pp. 553–562.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[6] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.

[7] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[8] E. Loper and S. Bird, “Nltk: The natural language toolkit,” *CoRR*, vol. cs.CL/0205028, 2002. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr0205.html#cs-CL-0205028