BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2023
http://excel.fit.vutbr.cz

Automatic Webpage Content Categorisation and Extraction

Michal Rein*

Abstract
The aim of this work is to create a versatile and efficient system for the content categorisation and extraction from webpages, with the focus on the topic domain of darknet forums and markets. This domain lacks the availability for sufficient data sets which could be used for the training of models in a traditional supervised fashion. We used state-of-the-art models based on the transformer deep neural network architecture, trained on the textual entailment task, which can be used as highly flexible classifiers, and other models for the content segmentation and named entity recognition. The output of this work is a scalable and fully containerised application with a web client, which allows its users to categorise the content with custom labels based on their needs. We believe that our solution can be useful for the analysis of the content from non-standard and niche domains, such as the darknet environment, generating data for further analysis.

*xreinm00@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

The categorisation of the webpage content itself is a well-managed area if a sufficient annotated data set containing information about the content category is available for training the machine learning or deep learning algorithms in a supervised fashion. However, if no such data set is available, the manual creation of such data set is a highly time-consuming task, which scales with the number of desired categories to be recognised by the algorithm. We looked for ways to overcome the need for highly specialised data sets for niche domains, like the darknet environment, providing flexibility to define new categories with ease.

We also aspire to create a fully automatic system, which disposes of a set of tools and methods that will process any web page represented by HTML code, segmentate its parts to find a desired content, categorise this content, and, in addition, look for named entities within the text, which enables the possibility to search for specific events, locations, users, and more.

To make our system available to a wider audience, we have also implemented a fully interactive web application, which provides significant control over most of the features that our system has to offer.

2. Poster Commentary

The poster contains three main sections, which are dedicated to the key parts of our work. The following section is organised into parts reflecting the layout of the poster.

2.1 Textual Entailment

This section describes the main idea behind the method used for the construction of flexible classifiers. We found a group of pre-trained neural network models based on the transformer architecture, fine-tuned for the Textual Entailment task. The common data set used to train these models is the Standford Natural Language Inference (SNLI) Corpus [1], which includes triplets of the premise, hypothesis, and label for each entry. For each premise in the data set, there are a total of 3 premise-hypothesis pairs for each of the following labels ($entailment, neutral, contradiction$). Labels represent a relationship between the premise and the hypothesis. An example taken directly from the SNLI Corpus can be seen in the figure located on the right side of the Textual Entailment section present in the poster.

If the hypothesis is cleverly constructed, the model can serve as a classifier to predict the categories.

The main figure in the poster under Textual Entailment section demonstrates this use-case. However, this approach has drawbacks. Although the common classifier would contain a part called `classification head`, which contains the outputs of all defined categories, the method we use can output the prediction for only one category at a time. This leads to the fact that the classification of a single premise requires inference to be made N times, where N is the number of categories. This drawback is compensated for the ability to choose any label the user wants and even swap categories at runtime.

As for the model, we used the model already pretrained from a Ph.D. candidate researcher Moritz Laurer (VU Amsterdam) [2], which is based on the Microsoft `DeBERTA-V3-large` architecture [3].

2.2 Templating

Templating is a method that we use to identify content fragments in the DOM representation of the web page. The content fragment can be defined as a repeating entity present on the web page, but in our specific case, we understand them as an analogy to forum posts or comments. The content fragment can be divided into parts that we call attributes. Each attribute has an identification vector of the element node and a type, denoting, for example, post authors, titles, or message content. Multiple linked attributes form a template that the `parser` component uses to extract specific parts from the web page and send them to further analysis.

We let users define their own templates either manually with our special template creation tool present in the client application, or in an automatic way, which uses the OpenAI GPT-3.5 model, to perform fully automatic segmentation and identification of the repeating content fragments. For now, the prompts crafted for the GPT model serve the specific need to identify the authors, titles, and message content for our use case. However, we plan to implement a much more generic way of prompt construction, which will allow users to specify their needs. We are still researching the different ways of how these prompts can be crafted in the most compact ways possible, in order to minimise overall cost of the template creation process.

The whole process of an automatic template creation with the GPT-3.5 model can be seen in the diagram present right at the start of the Templating section in the poster. The figure below visualises the process of minimising the amount of tokens sufficient for the GPT model to still identify the desired segments without the need of sending the entire content of the web page, which greatly reduces the overall cost of the process.

2.3 System

The system was built following the principles of microservice architecture to provide high flexibility, scalability, and maintainability. The first diagram of the System section in the poster provides an overview of the system composition. The whole system runs in a containerised environment that uses `Docker` technology and the `Docker Compose` tool to orchestrate the application. Each entity (except the devices) visible in the diagram is an independent containerised application.

The main processing unit of the system is the `Worker` service, which implements the processing pipeline. There can be multiple worker instances present in the system, managed by the `Scheduler` service, which reserves `Worker` instances for incoming processing requests.

The processing pipeline is visualised by the diagram right under the overview of the system in the poster. All results are stored in the relational database, and it is possible to view and filter the results through the client application.

For completeness and respect for the authors, we also used the model for the Named Entity Recognition (NER) task provided by a tool `T-NER` [4]. This model is based on `RoBERTa` architecture and trained on the specialised `TweetNER7` [5] data set.

References

[1] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2015.

[2] Wouter van Atteveldt Andreu Salleras Casas Laurer, Moritz and Kasper Welbers. Less annotating, more classifying – addressing the data scarcity issue of supervised machine learning with deep transfer learning and bert - nli, June 2022.

[3] Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2021.

[4] Asahi Ushio and Jose Camacho-Collados. T-NER: An all-round python library for transformer-based

named entity recognition. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 53–62, Online, April 2021. Association for Computational Linguistics.

[5] Asahi Ushio, Leonardo Neves, Vitor Silva, Francesco. Barbieri, and Jose Camacho-Collados. Named Entity Recognition in Twitter: A Dataset and Analysis on Short-Term Temporal Shifts. In *The 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, Online, November 2022. Association for Computational Linguistics.