Full Content-based Web Page Classification Methods by Using Deep Neural Networks

Suleyman Suleymanzade *, Fargana Abdullayeva

Institute of Information Technology, Azerbaijan National Academy of Sciences, Baku, Azerbaijan

Abstract  The quality of the web page classification process has a huge impact on information retrieval systems. In this paper, we proposed to combine the results of text and image data classifiers to get an accurate representation of the web pages. To get and analyse the data we created the complicated classifier system with data miner, text classifier, and aggregator. The process of image and text data classification has been achieved by the deep learning models. In order to represent the common view onto the web pages, we proposed three aggregation techniques that combine the data from the classifiers.

Keywords  Web page classification, LSTM, web crawler, deep learning, data aggregation

AMS 2010 subject Information retrieval 68P20, 62H30

DOI: 10.19139/soic-2310-5070-1056

1. Introduction

Information retrieval (IR) systems play an important role in modern-day society [1]. The goal of an information retrieval system is to collect, store, and provide an efficient search mechanism for the client. During the last decades, information retrieval systems have come a long way from the Boolean model [2] systems for Artificial Intelligence (AI) based [3] complicated models. The client wants to get the relevant data from the search system. Organizing the users queries into the set of target categories, belonging to the area of query categorization, which is important for the search relevance. The quality of the indexing and classification process plays a crucial role in the information retrieval process.

To perform relevant information retrieval information should be classified effectively. The most common web page classification methods are based on text [4][5] and graph data [6] analysing. This approach is explained by the fact that the classification of the rest of the embedded media data such as images, audio, and video data is a time-consuming and computationally expensive process. Because the power of the computing systems was dramatically increased during the last five years [7] it gave the new capability for data scientists to develop new methods for webpage categorisation. In this work, we proposed the models that we called aggregation strategies for merging different classification algorithms in order to achieve more accurate results and to discover the new web page categories. Discovering the new web pages classes allow for the retrieval systems (built on top of the categorization system) to find additional materials as the results on queries.

The article organized as follows: The problem definition in Section 2 where we explain the reason why do we need to use different web page classification algorithms and combine them to get the consistent representation of the target classes, next we discuss some related works in Section 3. The classifiers system discussed in Section 4 where we also cover the work principles of text classifier and image caption generator. We also include the
aggregation strategies and algorithms to discover the new web page target classes in Section 4, the results of a process combining the target data are shown in Section 5, after what we discussed the future works in Section 6, and conclude the article in Section 6.

2. Problem definition

The amount of data on a web page must be enough to classify it efficiently. The data in a page presented as text, image, URLs, or meta tags. Each of these data types must be analysed with different algorithms. If some of the data are not presented in a web page, then the process of classification must be laid down on the rest of the data. For example, some web pages may include many images without (or with small amount) of text data, some web pages don't have a metadata in meta tags. The process of web page categorization required to include mechanisms for aggregating different classifier results in order to increase the classifiers accuracy and find the new target classes that are not shown in the metadata. In results of discovering the new class tags, the search engine build upon the target class data will produce more relevant data. To solve this problem, we created the web crawler [8] system with two different classifiers subsystems that classifies text and image data separately and then aggregate the results of both subsystems. For the aggregation process, we modelled three aggregation strategies that are shown in combiners part.

3. Related Works

There are many studies in developing a web page classifier. Some of them based on one aspect of the data that existed on a page another based on hybrid approach and include more than one method. Some of these methods give priority to classification speed and not the accuracy, and these methods generally based on analysing the meta tag combinations and do not use the content-based information that requires machine learning technique to discover the new class labels. One of the interesting content-free method that includes machine learning technique discussed in [9] which is based URLs analysing only to classify the content of the link itself without analysing the full web page content. The content-based methods can be organized as a text based or image-based web page classification techniques. In [10], there was discussed the technique of page categorization of images data with support of CNN deep learning model. Another approach based on meta tag information was discussed in [11] where RNN was used during the test phase. There is also a hybrid method to classify data based on content and link-based (URLs) [12] method. The scientific works that are mentioned above requires in aggregation strategy to combine these methods in order to discover the new target classes. Some classification techniques that are related to relational data was discussed in [13]. In general, the aggregation technique can be separated by features based and class-based results [14]. In our work we used the technique based on the class prediction results.

4. Proposed system Architecture

The architecture of the classifier includes several blocks: 1. Miner 2. Image caption generator [15], 3. text classifier and 4. combiner. Each of these blocks responsible for the next tasks: miner includes web crawler that gathers text and image data from the internet, it also estimates the weights of text and image data then stores it in separated repositories.

Image caption generator generates the text related to the images, then the text classifier classifies text data from the miner and the image caption generator. The last block is a combiner that aggregates results from both text classifiers.
4.1. Miner

The component that includes a web crawler and storage system for these data structures we called a miner. The loosely coupled architecture of the system allows us to use other approaches [5] for data mining as well. The web page categorization process starts from the mining of the web pages. The mining process of text and media data achieved by a web crawler (spider) [2] that traverses over external links by Breath first search or Depth first search strategy [3]. While crawler traverses through the web pages it stores data in associated key-value principles [4]. For each gathered web page, the key represents a hash code of the web page address, the value represents the references of three data components: text, images and meta tags from a web page that contained meta data keywords these references located in separated data structures for storing text and binary data. The picture below shows the structure of the saved image and text components for each webpage.

Each webpage paragraph and image stores in a separated bucket with associated weight, this paired data we call a data component. The weights represent the priority of each data component that later used in the summary of category computation. Initially, the weights that related to each text paragraph in the web page are equal to one, the flexibility of loosely coupled architecture of the system allows to calculate the weights for each data component separately based on various algorithms The algorithm for weights computation can be based on the next properties:

1. Text appearance: font styles, colours and size of the text data in each paragraph. This approach, computational cheap and fast [16],
2. Paragraphs and images location. This method includes: tag hierarchy analysing in DOM [17] [18].
3. Numerical statistics, where algorithms such as TF-IDF [19], Okapi BM25 [20] [21].
4. Combining techniques where one or more methods can be used to compute the weights for each data component.

In this work for analysing the weights we used the method based on text appearance where text paragraphs that include more than a half of the text in bold or italic style, the weights are taken as 1.5 instead of 1.

4.2. Image classifier

The image classifier includes a deep neural network for an image caption generation. It receives the image data components from the miner and generates the feature by using YOLO [22][23] algorithm. The general principles of image caption generator based on [24] consist of two neural networks: YOLO based CNN, for feature extraction and LSTM [25] for generating the text sequence, which is similar to [15] model but instead of RNN [26], the LSTM was used because it carries relevant data during the training process and excludes non-relevant information by forget gate.

![Image Caption Generator Diagram](image)

Figure 3. Image caption generator

Figure 3 shows the Merge Architecture for Encoder-Decoder Model from [15]. For training the image caption generator we used the flickr_8K dataset.

4.3. Text classifier

For a more complex dataset with more than five classes such as 20 newsgroups, in order to achieve more accurate results, there are other options such as a convolutional neural network for text classification [27], a combination of multichannel neural networks [29] or more complex solutions with RNN [30]. In our case, for the real-time system that gathers webpages and works continuously in the background the simple and efficient class computation with accuracy more than 95% is enough. More complex extensions that require more computational resources can be achieved with the help of HPC [31] platform and continuous deployment DevOps [32] methods.
4.4. Combiner

The goal of the combiner is to aggregate the results from the text and imagedata component. The data that the combiner receives is represented by the data structure shown in Figure 6. The data structure contains the header and the data part (separated with a dashed line). The header includes a web pages hash code and two global weights: $W_i$ represents the global image weight and $W_T$ global text weight respectively. The data part includes the set data components with three parameters: 1. Tag $(I,T)$ that shows to which data type the component belongs to, 2. Local weight (float number on a picture after the tag) and ordered list data with the numeric representation of class labels (five float numbers).

![Confusion Matrix](image)

**Figure 5. Confusion Matrix**

This data is enough for the combiner to aggregate the set of data components for each web page separately. We decided to model three aggregation strategies for class combining to produce and then analyse the results of each strategy. Combiner generates the result tags $T$ by aggregating the results of separated images $C_{image}$ and texts $C_{text}$ data components. Where $C_{image} = \{c_{image}^i : c_{image}^i \in \mathbb{R}^n\}$. And aggregation function $A$ can be defined as

$$A = \frac{W_i \cdot c_{image}^i + W_T \cdot c_{text}^i}{W_i + W_T}$$
\[ A : O \left( \{ C^{\text{text}} \rightarrow w^{\text{text}}, W^{\text{text}}, C^{\text{image}} \rightarrow w^{\text{image}}, W^{\text{image}} \} \right) \rightarrow T \]  

(1)

That includes mapped image and text components with respect to their local weights.

\[ c^{\text{image}} \rightarrow w^{\text{image}} : \{ \forall i : c_i^{\text{image}} \rightarrow w_i^{\text{image}} \} \]  

(2)

\[ c^{\text{text}} \rightarrow w^{\text{text}} : \{ \forall i : c_i^{\text{text}} \rightarrow w_i^{\text{text}} \} \]  

(3)

Two global weights parameters \( W^{\text{image}}, W^{\text{text}} \) regulate the classification priority for each component type. In our case, there are only two component types (text and image). Output function \( O \), may differ depending on the objectives presented below:

1. To get only one class from the aggregator, the \( O \) must be taken as the \text{argmax} function.
2. If the goal is to get the \( n \) constant number of categories, then \( O \) function must choose the first \( n \) highest number of the categories (if \( n \) is not higher than the total number of classes)
3. To discover the new \( n \) classes from the web page then \( O \) function must process as the pseudocode that presented in Algorithm 1 below.

\begin{algorithm}
\SetAlgoLined
\DontPrintSemicolon
\textbf{function} FIND\_N\_NEW\_CLASSES(excited\_tags, n, classes = \{class: value\})
\State result $\leftarrow$ set()
\State inserted\_classes $\leftarrow$ 0
\State sorted\_classes $\leftarrow$ sort(classes.values, ascending)
\For {class \textbf{in} sorted\_classes}
\If {len(inserted\_classes) < n and class not in excited\_tags}
\State result.insert(class)
\State \( n \leftarrow n + 1 \)
\EndIf
\EndFor
\State \textbf{return} result
\Endfunction
\end{algorithm}

Algorithm 1

We defined three aggregating strategies for combining the data components.

\[ T = O \left( \sum_{i=1}^{n} w_i^{\text{image}} c_i^{\text{image}} + \sum_{j=1}^{m} w_j^{\text{text}} c_j^{\text{text}} \right) \]  

(4)

The first equation (Equation 4) shows the \text{local weights-based aggregation strategy} that achieved by the class addition of each data component with its local weight. The local weights characterise the priority of each data component on the web page. If there is no strategy to calculate the weights, the weight parameters are initialized as 1.

\[ T = O \left( W^{\text{image}} \sum_{i=1}^{n} w_i^{\text{image}} c_i^{\text{image}} + W^{\text{text}} \sum_{j=1}^{m} w_j^{\text{text}} c_j^{\text{text}} \right) \]  

(5)

The second strategy (Equation 5) shows the \text{aggregation with global weights} \((W^{\text{image}}, W^{\text{text}})\) where each global weigh gives priority to one of the data components types. This strategy is good for the web pages where the number of data components of one type is more than another. For example, there are some web pages with many images and a few text data of vice-versa and the global weights must be selected according to the difference of image and text data number.

\[ T = O \left( W^{\text{image}} \sum_{i=1}^{n} w_i^{\text{image}} c_i^{\text{image}} + W^{\text{text}} \sum_{j=1}^{m} w_j^{\text{text}} c_j^{\text{text}} \right) \]  

(6)
The third (Equation 6) aggregation strategy uses global weights with normalization function $S$ that scales the result of each component. In our experiment, we used the softmax function to achieve a similar scaling factor between image and text component sets.

5. Experimental data and results

5.1. Experiment 1

For the experiment as an example, we took two web pages. In the first example, we took the http://www.bbc.com/travel website and gather seven text paragraphs with seven images data, next to aggregate them, we used three (Equation 4-6) strategies to get the combined results. In the first website we didn’t calculate the global and local weights, by default the weights have been selected as parameter 1. The text and image classifiers have classified the gathered data as shown in Table 1-2. The distribution of target data shown in Figure 7.

![Figure 7. The target classes distribution from the (http://www.bbc.com/travel)](image)

Table 1. The text-based target class distribution (http://www.bbc.com/travel)

| C1: business | C2: entertainment | C3: politics | C4: sport | C5: tech | Labels (argmax) |
|--------------|-------------------|--------------|-----------|----------|-----------------|
| 0.2083       | **0.2207**        | 0.1988       | 0.20034   | 0.1717   | entertainment   |
| 0.2329       | **0.2422**        | 0.2016       | 0.1968    | 0.1262   | entertainment   |
| 0.2329       | **0.2422**        | 0.2016       | 0.1968    | 0.1262   | entertainment   |
| **0.2422**   | 0.2329            | 0.2016       | 0.1968    | 0.1262   | business        |
| 0.2047       | 0.1676            | 0.2341       | **0.2915**| 0.1019   | sport           |
| 0.2057       | **0.2890**        | 0.1600       | 0.2163    | 0.1287   | entertainment   |
| 0.1874       | **0.2901**        | 0.1627       | 0.2144    | 0.1451   | entertainment   |

After the classification, the next phase is combining the results from both classifiers. Figures 8 shows three data components.
Table 2. The Image data-based target class distribution (http://www.bbc.com/travel)

| C1: business | C2: entertainment | C3: politics | C4: sport | C5: tech | Labels (argmax) |
|--------------|-------------------|--------------|-----------|----------|----------------|
| 0.1990       | 0.2428            | 0.2018       | 0.2183    | 0.1378   | entertainment  |
| 0.2106       | 0.20272           | 0.2189       | 0.1998    | 0.1678   | politics       |
| 0.2197       | 0.2060            | 0.1934       | 0.2329    | 0.1478   | sport          |
| 0.1976       | 0.2575            | 0.1796       | 0.1854    | 0.1797   | entertainment  |
| 0.1889       | 0.1954            | 0.2512       | 0.2330    | 0.1313   | politics       |
| 0.2093       | 0.2356            | 0.1850       | 0.2078    | 0.1621   | entertainment  |
| 0.2219       | 0.1866            | 0.1622       | 0.2586    | 0.1704   | sport          |

In a result we get three answers based on Strategy 1-3 (Equation 4-6) shown in Table 3 below.

Table 3. The Image data-based target class distribution (http://www.bbc.com/travel)

```
| Strategy   | C1: business | C2: entertainment | C3: politics | C4: sport | C5: tech | Labels (argmax) |
|------------|--------------|-------------------|--------------|-----------|----------|----------------|
| Strategy 1 | 5.7000       | 10.5379           | 4.5729       | 6.0963    | 4.0926   | entertainment  |
| Strategy 2 | 4.4657       | 8.4958            | 3.4136       | 4.9322    | 3.2925   | entertainment  |
| Strategy 3 | 0.1828       | 1.3787            | 0.1552       | 0.1818    | 0.1014   | entertainment  |
```

In a result we get three answers based on Strategy 1-3 (Equation 4-6) shown in Table 3 below.

We compared the results from three strategies with the meta data of the web page and discover the new topic, the *Sport* that wasn't presented in the meta data. The discovering process that we used here were based on idea to find only the first label that wasn't presented on the web page.

5.2. Experiment 2

For the second experiment, we chose https://www.espn.com/ website. At this time, we calculated parameters with global weights that were calculated according to the relation between the number of text paragraphs to the images number on the web page.

The class distribution shown in Figure 9. As shown in Table 4 image classifier gave priority to *politics* classes instead of the *sport*. The *ESPN* website includes many links, adverts, and images that are not related to the webpage topics. The one way to solve this problem is to use filter functions during the crawling process that will ignore cookies, but in our work, we wanted to show the power of global weights that give priority to the one classifiers decision instead of other. Table 5 shows that the most of paragraphs have been classified as the *sport*. Because of the amount of text data is more than the image data the global weights will give more priority to the text classifier than in image. In our example the global weight for the image classifier was given as 0.2 and for the text classifier 1.2.
Table 4. The image-based target class distribution [(https://www.espn.com/)](https://www.espn.com/)

| C1: business | C2: entertainment | C3: politics | C4: sport | C5: tech | Labels (argmax) |
|--------------|-------------------|-------------|----------|----------|---------------|
| 0.2009       | 0.2337            | 0.2768      | 0.1404   | 0.1479   | politics      |
| 0.2009       | 0.2337            | 0.2768      | 0.1404   | 0.1479   | politics      |
| 0.1747       | 0.1674            | 0.1253      | 0.3938   | 0.1385   | sport         |

Table 5. text data-based target class distribution [(https://www.espn.com/)](https://www.espn.com/)

| C1: business | C2: entertainment | C3: politics | C4: sport | C5: tech | Labels (argmax) |
|--------------|-------------------|-------------|----------|----------|---------------|
| 0.1275       | 0.1353            | 0.1479      | 0.4733   | 0.1157   | sport         |
| 0.1510       | 0.2002            | 0.3027      | 0.1816   | 0.1642   | politics      |
| 0.2165       | 0.1914            | 0.2515      | 0.1929   | 0.1474   | politics      |
| 0.1067       | 0.1986            | 0.2166      | 0.3498   | 0.1280   | sport         |
| 0.1327       | 0.2613            | 0.1817      | 0.2956   | 0.1285   | sport         |
| 0.1540       | 0.1873            | 0.1926      | 0.3210   | 0.1449   | sport         |
| 0.1623       | 0.1984            | 0.1354      | 0.4048   | 0.0989   | sport         |
| 0.2478       | 0.1633            | 0.2061      | 0.2654   | 0.1172   | sport         |
| 0.2472       | **0.2476**        | 0.1762      | 0.1909   | 0.1378   | entertainment |
| 0.0717       | 0.3311            | 0.1202      | **0.4209** | 0.0558   | sport         |

Table 6. The aggregation results [(https://www.espn.com/)](https://www.espn.com/)

| Strategy | C1: business | C2: entertainment | C3: politics | C4: sport | C5: tech | Labels (argmax) |
|----------|--------------|-------------------|-------------|----------|----------|---------------|
| Strategy 1 | 2.3061 | 3.7155 | 3.0506 | **3.9119** | 2.0157 | sport         |
| Strategy 2 | 1.7889 | 3.3561 | 2.4278 | **3.7386** | 1.6883 | sport         |
| Strategy 3 | 0.1361 | 0.3963 | 0.2090 | **0.5348** | 0.1236 | sport         |

Figure 10 shows three aggregation strategies-based images and text components class parameters distribution. The results of experiment 2 shown in Table 6.

In this example the new discovered label that wasnt presented in the meta data was politics.
6. Future Works

The architecture of the classifier system can be modified: instead of chain with image caption generator and text classifier use direct image classifier that classifies images by activity. This method will allow to use image and text classifiers with different label sets and generate the tags, based on Boolean, continuous or combinational results of two classifiers.

The crawling process [33] can be optimized by filtering advertisement data on images and text data [34]. The process of finding advertisements in image and text data can be achieved by comparison the distance between text and image labels, the numerical distribution between class probabilities must be relatively similar. The outlier [35] can be detected by using z-score [36], Dbscan [37], isolation forests [38] algorithms.

The process of generating the global and local weights between image and text components in class combiners can be achieved by the deep learning neural network where text and image classifiers compute categories for the web pages and train the weights according to the categories in metadata. The control over global and local weights and selecting the right aggregation strategy may be considered in future studies.

7. Conclusion

The improvements in web page classification effects to the performance of retrieval systems built on the top of it. The newly discovered categories in the web pages allow for search engines to sort and find more relevant results on queries. In this work, we improved the web page classification process by combining the results of text and image data classifiers. To achieve this goal, we built the loosely coupled categorization system to gather, store, and process text and image data. To combine the target summary of each data element we modeled three aggregation strategies. During the experiments we discovered the new categories of the web pages that have not been presented in the metadata.

REFERENCES

1. Shashidhara Hr, Gt Raju, Prakash Shivanna The Role of an Information Retrieval in the Current Era of Vast Computer Science Stream, International Journal of Soft Computing and Engineering, vol. 3, no. 3, 2013.
2. Arash Habibi Lashkari, Fereshteh Mahdavi, Vahid Gholami, in A Boolean Model in Information Retrieval for Search Engines, Information Management and Engineering, ICIME, Kuala Lumpur, Malaysia, 2009.
3. Thomas Mandl Artificial Intelligence for Information Retrieval, Encyclopedia of Artificial Intelligence, 2008.
4. Jochen Hartmann, Juliana Huppertz, Christina Schamp, Mark Heitmann Comparing automated text classification methods, International Journal of Research in Marketing, vol. 36, no. 1, pp. 20-38, 2019.
