Bursa Uludağ University Journal of The Faculty of Engineering, Vol. 27, No. 1, 2022 RESEARCH

DOI: 10.17482/uumfd.891038

WEB PAGE CLASSIFICATION WITH DEEP LEARNING METHODS

Mehmet Salih KURT [*]
Eylem YUCEL [**]

received: 31.03.2021 ; revised: 16.11.2021 ; accepted: 13.02.2022

Abstract: Today, millions of websites on the Internet are widely used to access information. For effective use of web pages with increasing numbers every day, they need to be well classified. In this study, binary and multi-class classification models have been created which can classify web pages with high accuracy. In our experiments, URLs and categories of English web pages in the Open Directory Project (ODP) were used. Training dataset was created by pulling web page texts from URL information. To our knowledge, this is the first comprehensive web page classification dataset for Turkish. In this study, Convolutional Neural Network (CNN), Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) deep learning methods which are effective in text classification are used. Word embedding was used instead of n-gram approaches commonly used for feature extraction in text classification studies. In this study, hyper-parameter optimization was performed for deep learning models. Binary and multi-class classification models were created with the best parameters. Binary classification models were compared with the results of another study, and multi-class classification models were compared with each other. The performances of all models were examined by considering their training time and f1 scores.

Keywords: Web Page Classification, Deep Learning, CNN, LSTM, GRU

Derin Öğrenme Yöntemleri ile Web Sayfası Sınıflandırma

Öz: Günümüzde bilgiye erişmek için internet ağı üzerinde milyonlarca web sitesi yaygın olarak kullanılmaktadır. Sayıları her geçen gün artan web sayfalarının daha etkin kullanılabilmesi için iyi bir şekilde kategorize edilmeleri önem kazanmıştır. Bu çalışmada, web sayfalarını yüksek doğrulukta sınıflandırabilen ikili ve çok sınıflı sınıflandırma modelleri oluşturulmuştur. Bu çalışmada, Açık Dizin Projesindeki (ODP) İngilizce web sayfalarının URL'leri ve kategorileri kullanıldı. Web sayfası metinleri URL bilgilerinden çekilerek eğitim veri kümesi oluşturuldu. Bildiğimiz kadarıyla bu, Türkçe için ilk kapsamlı web sayfası sınıflandırma veri setidir. Bu çalışmada, metin sınıflandırmada etkili olan Evrişimsel Sinir Ağı (CNN), Uzun Kısa Süreli Bellek (LSTM) ve Geçitli Tekrarlayan Birim (GRU) derin öğrenme yöntemleri kullanılmıştır. Metin sınıflandırma çalışmalarında özellik çıkarımı için yaygın olarak kullanılan n-gram yaklaşımları yerine kelime temsilleri kullanılmıştır. Bu çalışmada derin öğrenme modelleri için hiperparametre optimizasyonu yapılmıştır. En iyi parametrelerle ikili ve çok sınıflı sınıflandırma modelleri oluşturulmuştur. İkili sınıflandırma modelleri başka bir çalışmanın sonuçlarıyla ve çok sınıflı sınıflandırma modelleri kendi aralarında karşılaştırılmıştır. Tüm modellerin performansları eğitim süreleri ve f1 puanları dikkate alınarak incelenmiştir.

Anahtar Kelimeler: Web Sayfası Sınıflandırma, Derin Öğrenme, CNN, LSTM, GRU

[*] Department of Informatics, Hakkari University, Hakkari, 30100, Turkey
[**] Department of Computer Engineering, Istanbul University-Cerrahpaşa, Istanbul, 34320, Turkey
Correspondence Author: Mehmet Salih Kurt (dr.salihkurt@gmail.com)

1. INTRODUCTION

Today, millions of websites are easily accessible via the Internet, and the number of these websites is increasing day by day. These websites offer a huge amount of content every day, including millions of URLs.

The classification of websites according to the content provides great convenience to users who use search engines. The classification of contents allows users to reach what they want faster and easier. This is important in presenting web pages related to the category that users are looking for. In addition, determining the category of the web page is also important in the selection of advertisements to be placed on websites. Placing advertisements according to the category of the website will more affect users and will increase earnings from advertisements.

Millions of URLs on platforms such as social media sites (Facebook, Twitter, Instagram, etc.) and e-mail service providers (Gmail, Outlook, Yahoo Mail, etc.) can become more useful with an efficient classification and malicious websites can be detected in advance. The effective classification of websites is also important for children to surf the Internet more useful. In this way, it is possible to prevent the sites that can have a negative impact on children and to present the contents that may be more useful to them.

There are many studies on web page classification in the literature. These studies can be divided into two categories as content-based approaches and URL-based approaches. Content-based approaches use a variety of features in the content and structure of Web pages. They have better classification performances as they contain more information than approaches that use only URL information.

Table 1 summarizes URL-based and content-based proposals. In content-based approaches, Dumais and Chen (2000) used the title, description and keyword fields in the META tag beside the web page text. Then they applied the Support Vector Machine (SVM) method. Lai and Wu (2002) used meaningful term extraction and discriminative term selection approaches to get the features required for classification. In this way, they have classified by creating specific terms for a specific area. Tsukada et al. (2001) classified the web pages using the nouns which are in the texts of the web pages. A stop-list was used to remove useless words and apriori algorithm was used to select the words required for classification training. They applied the Decision Trees method to classify the web pages. Chen et al. (2006) proposed an algorithm to deal with the high dimensionality problem of vector space and generate a set of properties. They used the Adaptive Fuzzy Learning Network (AFLN) method for classification. Sun et al. (2011) developed a conceptual method of web classification using an entity-based co-training (EcT) algorithm. In this algorithm, all pages are categorized by considering the hidden entity semantics in web pages. Ozel (2011) used HTML tags of web pages and terms belonging to each tag as a feature. He used a Genetic Algorithm (GA) based approach for the automated web page classification model. Chen and Hsieh (2006) developed a model that uses a SVM to automatically classify webpages. In this model, they combined latent semantic analysis and web page feature selection to extract semantic and text properties. Chen et al. (2009) proposed two new feature selection approaches for web page classification. These are an efficient discriminating power measure for feature selection and a two-level promotion technique to improve the behavior of some relevance measures, often used in text classification. Kwon and Lee (2003) proposed a method comprising web page selection, web page classification and web site classification phases. In the first stage of this method, several representative web pages are selected using link analysis. Then it classifies each of the selected web pages using the k-NN classifier. In the last step, classified webpages are expanded into a classification of the entire website. Selamat and Omatu (2004) combined the features created from the most regular words in each class with the most suitable features selected from the Principal Component Analysis (PCA). They performed the classification by using these property vectors as input to neural networks. Lee et al. (2015) proposed a Simplified Swarm Optimization (SSO) algorithm for

automatic classification of web pages. They compared their algorithm performances with well-known classifiers based on their datasets.

Table 1. List of content-based and URL-based proposals

Content-based approaches	URL-based approaches
Dumais and Chen (2000)	Kan (2004)
Lai and Wu (2002)	Vonitsanou et al. (2011)
Tsukada et al. (2001)	Nicolov and Salvetti (2007)
Chen et al. (2006)	Baykan et al. (2009)
Sun et al. (2011)	Baykan et al. (2011)
Ozel (2011)	Baykan et al. (2013)
Chen and Hsieh (2006)	Chung et al. (2010)
Chen et al. (2009)	Lee et al. (2015)
Kwon and Lee (2003)	Hernandez et al. (2014)
Selamat and Omatu (2004)	Kan and Thi (2005)
Lee et al. (2015)	

In URL-based approaches, web pages are classified by using URLs instead of content. This method is much faster because it doesn't require getting page content or parsing text. This approach is also helpful when the page content is not suitable for processing. In the URL-based classification, in previous studies, URLs are divided by punctuation and the terms are used as the property set of the classifier (Kan, 2004). Researchers further divided URLs beyond punctuation using statistical or brute force approaches. Dictionary based tokenizes (Vonitsanou et al., 2011), information content (Kan, 2004), and symmetric / non-symmetric sliding windows (Nicolov and Salvetti, 2007) were used in non-brute force approaches. Brute force approaches tend to use all sub-strings, all grams, as the feature set of the classifier. Baykan et al. (2009, 2011, 2013); Chung et al. (2010) proposed URL-based web page classifiers that create a set of URL patterns representing different page classes on a website so that more pages can be classified by matching their URLs with patterns. Kan and Thi (2005) proposed a supervised web page classifier based exclusively on features derived from the URLs of web pages. Besides URLs tokens, they also used features such as the position of each token in the URL, the length of the URL. They created a web page classifier by giving these features as input to the entropy maximization algorithm. Baykan et al. (2009) extracted the feature vectors by using whole tokens and using letter n-gram of the tokens. They used the ODP library as a dataset and created separate classification models for 15 categories. Each binary classification model determines whether a given URL belongs to that category.

The main purpose of this study is to create web page classification models with high classification accuracy. The contents of web pages were obtained using URL. Web page texts in these contents are used for classification models. In short, a content-based approach was used.

Our proposed method relies on obtaining and processing instant web page texts using URL information. Word2vec was used to extract features from these texts. The effective tools used in the proposed model made it possible to quickly retrieve web page text from the URL. Afterwards, these texts were classified by using deep learning methods which are effective in text classification. Thus, the model we propose is faster than other content-based models and has better accuracy than URL-based models.

In this study, both binary and multiclass classification models were formed. Binary classification models were compared with the models of Baykan et al. (2009). The accuracy and

training time of multi-class classification models were compared and the advantages and disadvantages of the models were examined.

2. MODELS

In this study, deep learning methods which are effective in text classification are used. This section gives a brief introduction of models (CNN, LSTM and GRU) used in this study.

2.1. Convolutional Neural Networks

CNNs are considered the basic architecture of deep learning. The architecture of the convolutional neural network has one or more successive convolution layers and a pooling layer. These layers are combined with a fully connected layer and classification layer, respectively. In this study, the CNN model proposed by Kim was used (Kim, 2014). The architecture of this model is a slight variant of Collobert's CNN architecture (Collobert et al., 2011). In the CNN architecture shown in Figure 1, important features are extracted from the input data using the convolution layer, sub-sampling layer, fully connected layer and classification layer in this architecture. The categories of input data are determined by using these properties.

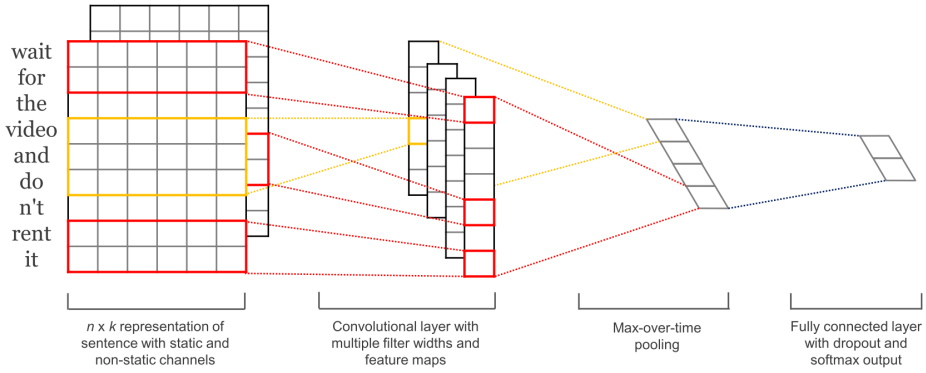

Figure 1:
Model architecture with two channels for an example sentence (Kim, 2014)

The input layer consists of n inputs where each input is represented by k-dimensional dense vector. Hence, the input x is represented by a $d\ x\ k$ dimensional feature map. Let $x_i \in \mathbb{R}^k$ be the k-dimensional word vector representing the i-th word in the input sentence. A sentence with length n is represented as:

$$x_{1:n} = x_1 \oplus x_2 \oplus \ldots \oplus x_n \tag{1}$$

Where $\oplus$ is the concatenation operator. A convolution operation involves applying $w \in \mathbb{R}^{hk}$ filter to a window of h words to generate a new feature. For example, using a window of $x_{i:i+h-1}$ words, a new property c_i feature is generated as follows:

$$c_i = f(\mathrm{w}.x_{i:i+h-1} + b) \tag{2}$$

In Eq. (2), f is a non-linear function such as the hyperbolic tangent and $b \in \mathbb{R}$ is a bias term. A feature map is generated by applying this convolution filter to every possible window of words in the sentence $x_{1:h}, x_{2:h+1}, \ldots, x_{n-h+1:n}$. This feature map is generated according to Eq. (3):

$$c = [\, c_1, c_2, \ldots, c_{n-h+1}] \tag{3}$$

Here $c \in \mathbb{R}^{n-h+1}$. We then take the maximum values corresponding to the filters by applying a max-overtime pooling operation on the feature map. The purpose of this process is to capture the most prominent features in feature maps.

The model aims to detect different features using multiple filters in varying window sizes. The outputs of the layer consisting of these features are transferred to the last layer, a fully connected layer. The probability distribution on the labels is calculated in a fully connected softmax layer.

2.2. Recurrent Neural Networks

RNNs have great potential in machine translation tasks (Sutskever et al., 2014; Auli et al., 2013; Liu et al., 2014). RNNs were initially used mostly for language models because of their ability to memorize long-term addictions. The main idea of RNN is that recurrent connections between hidden layers can affect outputs using the memory of previous inputs. However, there are two major problems of RNN, vanishing gradient and exploding gradient problems, which need to be solved during the training phase (Bengio 6 et al., 1994). The fact that long-term components in the derivatives of the activation function go exponentially to zero makes it difficult to learn the relationship between distant inputs. In order to solve this problem, specific RNN approaches such as LSTM and GRU using forget units were proposed. These models are designed to give memory cells the ability to optimize time delays and thus determine when specific information will be forgotten. Let's examine the structure of LSTM and GRU neural networks separately.

2.2.1. Long Short Term Memory

LSTM was first proposed in 1997 for language models and was known for its ability to memorize long-term dependencies (Hochreiter and Schmidhuber, 1997). The LSTM layers consist of memory blocks that are recurrently connected and each of these memory blocks includes three multiplicative gates. Gates perform a continuous a kind of write, read and reset operation to ensure that the temporary information is used for a specified period of time.

Figure 2 shows the typical structure of LSTM cells (Chung et al., 2014). Unlike from traditional recurrent unit, LSTM unit keeps the current memory $c_t \in \mathbb{R}^n$. The input of the unit, x_t, h_{t-1}, c_{t-1} and the output of the unit, h_t, c_t are updated as follows:

Gates:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{4}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{5}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{6}$$

Input transform:

$$g_t = \tanh(W_g x_t + U_g h_{t-1} + b_g) \tag{7}$$

State update:

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{8}$$

$$h_t = o_t \odot \tanh(c_t) \tag{9}$$

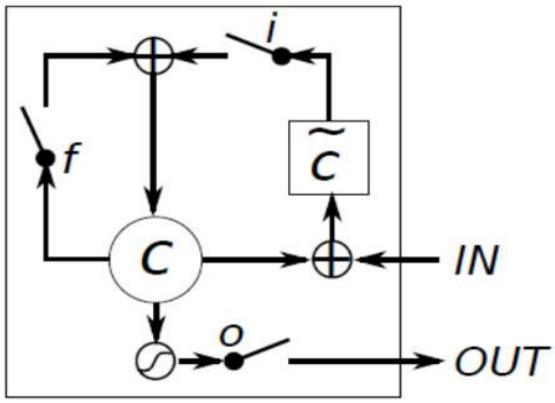

Figure 2:
LSTM single cell structure (Chung et al., 2014)

In the above equations, σ and $\odot$ denote the logistic sigmoid function and element-wise multiplication, respectively. The LSTM unit has an input gate i_t, a forget gate f_t, an output gate o_t, a hidden unit h_t and a memory cell c_t at each time step t. W, U are the learned parameters and b denotes the added bias. Intuitively, the input gate controls how much each unit is updated, the forget gate controls how much the memory cell is erased, and the output gate controls how much of the internal memory state is revealed.

2.2.2. Gated Recurrent Unit

The GRU proposed by Cho et al. (2014) is another type of RNN that uses memory cells. They are similar to LSTM but easier to compute and implement. The typical structure of GRU cells is shown in Figure 3. GRU also has gates that control the flow of information through cell states. It has fewer parameters than LSTM and does not have an output gate.

Gates:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{10}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{11}$$

State update:

$$\hat{\mathrm{h}}_t = \tanh(\mathrm{W}x_t + \mathrm{U}(r_t \odot h_{t-1}) + b_h) \tag{12}$$

$$h_t = (1 - z_t)h_{t-1} + z_t\hat{\mathrm{h}}_t \odot \hat{\mathrm{h}}_t \tag{13}$$

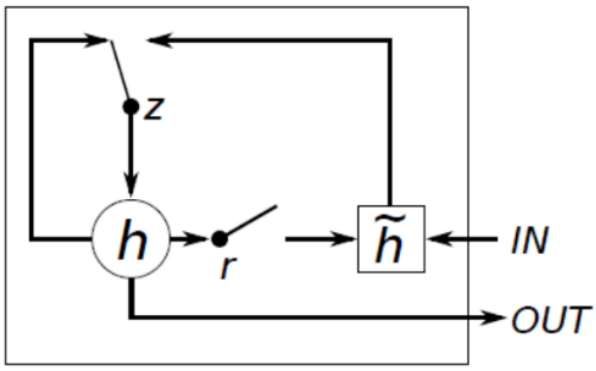

Figure 3:
GRU single cell structure (Chung et al., 2014)

In the above equations, r_t, z_t, x_t, h_t and $\hat{h}_t$ represent the reset gate, update gate, input vector, output vector and candidate activation, respectively. W denotes the weight matrix, and b denotes bias. Sigmoid function σ and hyperbolic tangent function tanh are used as activation functions.

3. DATASETS AND EXPERIMENTAL SETUP

In this study, Python programming language was used in the stages of obtaining web page contents from the URL information, preprocessing the data, preparing the data and creating the machine learning models. Python libraries such as Selenium and Urllib were used to extract web page content from URL information. The Pandas library was used in the pre-processing of the texts of the websites and Numpy library was used to make mathematical operations faster and more efficient.

In order to train the deep learning models, the Keras library, which runs the Tensorflow library in the background, was used. The Scikit-Learn library was used to measure the success of the models and to create complexity matrices.

More efficient classification models have been created by embedding vectors of words in web pages. This section also provides information about the word representations used in this study.

3.1. Dataset

In this study, ODP dataset, which is known as the world's largest online website directory project, has been used. In the ODP dataset, also known as DMOZ dataset, millions of websites have been examined by volunteer editors and classified according to their topics. All activities of the ODP project ended on 14 March 2017.

The categories “World” and “Regional” containing non-English websites in the ODP dataset were excluded from the training dataset. In our study, we chose the categories used by Baykan et al. (2009). Thus, we could compare the performances of the binary classification models. The dataset to be used for the training of models is English web pages consisting of 15 categories.

Figure 4 shows the categories of English websites in the ODP dataset and the number of web pages in each category.

Figure 4: *Comparison of the number of URLs in the ODP dataset*

As shown in Figure 4, the English web sites in the ODP dataset comprise 15 categories and each category has a different number of web pages.

That the sample numbers of the categories in test data differ from each other has a negative impact on performing learning models. Therefore, the dataset has been balanced by reducing the number of samples in categories with many web pages. 150 thousand web pages were selected, including 10 thousand samples for each category. The selenium library was used to obtain the text of web pages from the URL of 150 thousand web pages. The dataset used for each web page consists of a sequence of words, with a maximum of 1000 words. These words are created by sorting the words in the text of the web page according to the frequency of the words.

In this study, binary classification models were also created to compare with the study of Baykan et al. (2009). Their binary classification models determine whether a web page is a member of a category. Balanced datasets consisting of positive and negative samples were used for the training of binary classification models. For example, to determine whether a web page is a member of "Arts" category, 10K URL data from the "Arts" category were combined with 10k "Non-Arts" data received in balanced amounts from other categories. The binary classification model was trained with 80% of the 20K web page data. Then, the model was tested with the remaining 20% of the data and the f1 score was measured. Training and test datasets were created for all binary classification models in the same way.

3.2. Implementation of Word Embedding

Word embedding is basically word representation in which the human understanding of language is adapted to the machine. In this model, words with similar meaning have a similar representation in an n-dimensional space.

In this study, word embedding libraries commonly used in the literature were examined and the Glove library developed by Pennington et al. (2014) was used. In our experiments, all word vectors are 300-dimensional vectors provided by Glove. The word embedding vectors used in this study were pre-trained on an unlabeled corpus of about 840 billion tokens.

3.3. Hyperparameter Optimization and Training

In this study, hyper-parameter optimization was performed for CNN, LSTM and GRU deep learning models. This section provides information about the hyper-parameter optimization using 20% of the dataset.

The model and parameters used by Kim were used for the CNN model (Kim, 2014). In the CNN model, 3, 4 and 5 filters applied to the input vectors. Then, functions such as maximum pooling, dropout and activation were applied to the resulting matrices. Finally, the classification was carried out by applying the softmax function.

Table 2 shows the hyper-parameter optimization of the CNN model and the f1 scores of the resulting models. 20% of the dataset was used for all models. The training of all CNN models in this study was performed with the best parameters (parameters in rows marked with red in Table 2).

Table 2. Parameters applied to CNN model and their results

Batch Size	Filter Size	Dropout Rate	Learning Rate	F1 Score
32	128	0.2	0.0001	0.70
32	128	0.5	0.0001	0.71
32	256	0.2	0.0001	0.70
32	256	0.5	0.0001	0.72
64	128	0.2	0.001	0.71
64	128	0.2	0.0001	0.72
64	128	0.5	0.001	0.71

64	128	0.5	0.0001	0.73
64	256	0.2	0.001	0.72
64	256	0.2	0.0001	0.71
64	256	0.5	0.001	0.72
64	256	0.5	0.0001	0.72
128	128	0.2	0.0001	0.70
128	128	0.5	0.0001	0.72
128	256	0.2	0.0001	0.71
128	256	0.5	0.0001	0.72

Table 3 and Table 4 show the hyperparameter optimization of LSTM and GRU models, respectively. 20% of the dataset was used for all models and the resulting models were evaluated with f1 scores. The training of all LSTM and GRU models in this study was performed with the best parameters (parameters in rows marked with red in tables).

Table 3. Parameters and results of LSTM model

Batch Size	Dropout Rate	Learning Rate	F1 Score
32	0.2	0.001	0.71
32	0.2	0.0001	0.59
32	0.3	0.001	0.71
32	0.5	0.001	0.70
32	0.5	0.0001	0.63
64	0.2	0.001	0.72
64	0.2	0.0001	0.58
64	0.3	0.001	0.70
64	0.3	0.0001	0.68
64	0.5	0.001	0.71
64	0.5	0.0001	0.63
128	0.2	0.001	0.71
128	0.2	0.0001	0.62
128	0.3	0.001	0.72
128	0.5	0.001	0.71
128	0.5	0.0001	0.69

Table 4. Parameters and results of GRU model

Batch Size	Dropout Rate	Learning Rate	F1 Score
32	0.2	0.001	0.71
32	0.2	0.0001	0.69
32	0.3	0.001	0.70
32	0.5	0.001	0.72
32	0.5	0.0001	0.69
64	0.2	0.001	0.71

64	0.2	0.0001	0.67
64	0.3	0.001	0.71
64	0.3	0.0001	0.68
64	0.5	0.001	0.72
64	0.5	0.0001	0.66
128	0.2	0.001	0.70
128	0.2	0.0001	0.68
128	0.3	0.001	0.71
128	0.5	0.001	0.70
128	0.5	0.0001	0.66

4. RESULTS AND ANALYSIS

In this study, in addition to the multi-class classification models, binary classification models were created to compare the results of Baykan et al. in terms of accuracy and speed (Baykan et al., 2009). For the training of each binary classification model, 10K positive and 10K negative samples were used.

The negative training samples were created by combining the samples from the other 14 classes in a balanced way. Binary classification models were trained with 80% of web page data and tested with 20%. Table 5 lists the f1 scores of all binary classification models.

Table 5. Results of binary classification models

	Previous Study	**Proposed Models**		
Category	**(URL Based/N-gram/SVM)**	**CNN**	**LSTM**	**GRU**
Adult	0.87	0.93	0.92	0.92
Arts	0.81	0.92	0.88	0.89
Business	0.82	0.89	0.87	0.87
Computers	0.82	0.90	0.89	0.90
Games	0.86	0.91	0.90	0.89
Health	0.82	0.93	0.92	0.92
Home	0.81	0.90	0.89	0.89
KidsTeens	0.80	0.87	0.83	0.85
News	0.80	0.91	0.89	0.91
Recreation	0.79	0.91	0.88	0.90
Reference	0.84	0.91	0.90	0.91
Science	0.80	0.93	0.92	0.92
Shopping	0.83	0.91	0.89	0.89
Society	0.80	0.87	0.85	0.84
Sports	0.84	0.94	0.93	0.93
Average	**0.82**	**0.90**	**0.89**	**0.89**

According to the model scores in Table 5, all binary classification models created by CNN, GRU and LSTM have a better classification performance than the models in the study of Baykan et al. (2009). Since some URL information in the ODP dataset belongs to closed

websites, we could not pull the contents of web pages from these URLs. Therefore, much fewer data was used than the data used in the study of Baykan et al. (2009). Despite this, better classification models have been obtained through embedded words and deep learning models.

The models proposed by Baykan et al. have a URL-based approach, while the models we propose in this study have a content-based approach (Baykan et al., 2009). Features in the models proposed by Baykan et al. are extracted from URLs with n-grams and classify with SVM. The features we propose in this study have been extracted from the words in the content of websites using word2vec and classified with deep learning models. Since we use content-based approach, the models we propose are likely to be slower than their URL-based models. Classification using URLs has some advantages, such as faster classification, classification without downloading pages, classification when content hides in images. However, the main purpose of this study is the performance of classification rather than the speed of the models. For this purpose, much more accurate classification models have been obtained than Baykan et al. (2009).

In this study, multiple classification models were created for 15 categories using CNN, LSTM and GRU deep learning models. 150 thousand web pages were used, including 10 thousand samples for each category. 80% of the data set was used in model training and 20% was used in model testing. This study is one of the most comprehensive web page classification studies in Turkish language, with both dataset and classification models.

Table 6 shows the f1 scores of multi-class classification models that are trained with the text of the web pages. The word embedding has been applied to the input data of all models. Hyper-parameter optimization was performed on all models and f1 scores were obtained from test data. According to the f1 scores of the models, it can be said that the CNN model has a better classification performance than the LSTM and GRU models.

Table 6. Results of multi-class classification models

Model (15 Categories)	**F1 Score**
CNN	0.78
LSTM	0.76
GRU	0.77

Without considering the training times of the models, evaluating these results can mislead us. Let's examine the training time of the models to make a better evaluation. Figure 5 shows a comparison of the training time of their models. Considering both the f1 scores and training times of the models, it can be said that the CNN model is much better. From the Recurrent Neural Networks, the GRU model is better than the LSTM model because it has a shorter training time and a better f1 score. The LSTM model performed worse than the other models, both in terms of training time and f1 score.

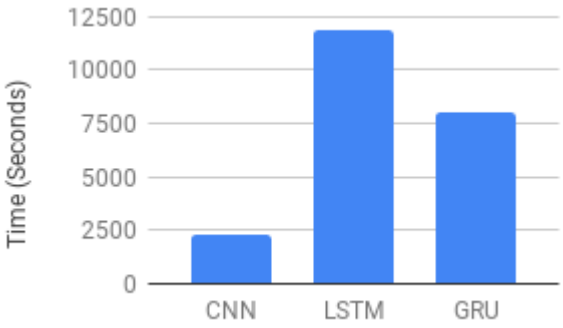

Figure 5:
Comparison of training times of models

When evaluating the results, we should also mention the impact of the dataset on the performance of the models. The dataset used here is a list of words on web pages. The sequence of words in this list does not consist of meaningful sentences because it is sorted according to the frequency of words. Therefore, it is not suitable for GRU and LSTM models that use the memory feature.

5. CONCLUSIONS

In this study, automatic web page classification models with high classification accuracy were created. The URL information of 15 categories with English content in the ODP dataset were used. The content-based method was used for this. Web page text is obtained from URL information and classification is made through these texts. To our knowledge, this is the first comprehensive web page classification dataset for Turkish.

Deep learning approaches CNN, GRU and LSTM which give effective results in text classification were used. Binary and multi-class classification models were created by using these approaches. In order to increase the performance of classification, word embedding was used instead of n-gram approaches commonly used in text classification studies. Hyper-parameter optimization has been done for deep learning models.

Multi-class classification models were compared by considering the training times and f1 scores. According to this, CNN model has the best classification since CNN model has less training time and more f1 score than LSTM and GRU models.

Besides the multi-class classification models, binary classification models were created to compare the results of Baykan et al. (2009). Although all the binary classification models we have created by CNN, GRU and LSTM use fewer datasets, they have better classification performance than the models in the study of Baykan et al. (2009). Since Baykan et al. (2009) propose a URL-based approach, their models are likely to be faster than our 15 content-based models.

In the future web page classification studies, screenshots of web pages can be used instead of the texts of web pages. Since the CNN approach performs well in image processing, the training set created with screenshots can be used. The CNN classification model created with web page images is likely to be both faster and more successful.

CONFLICT OF INTEREST

The authors acknowledge that there is no known conflict of interest or common interest with any institution / organization or person.

AUTHOR CONTRIBUTION

In this paper, Mehmet Salih Kurt contributed to all stages of the article, including data collection, determining the concept and design process of the research, data analysis and interpretation of the results, preparation of the manuscript. Eylem YUCEL contributed in all stages of manuscript except data collection.

REFERENCES

1. Auli, M., Galley, M., Quirk, C., and Zweig, G. (2013). Joint language and translation modeling with recurrent neural networks. In Proceedings of EMNLP, pages 1044–1054.

2. Baykan, E., Henzinger, M., Marian, L., and Weber, I. (2009). Purely url based topic classification. In Proceedings of the 18th international conference on World wide web, pages 1109–1110. doi:10.1145/1526709.1526880.

3. Baykan, E., Henzinger, M., Marian, L., and Weber, I. (2011). A comprehensive study of features and algorithms for url-based topic classification. ACM Transactions on the Web. doi:10.1145/1993053.1993057

4. Baykan, E., Henzinger, M., Marian, L., and Weber, I. (2013). A comprehensive study of techniques for url-based web page language classification. ACM Transactions on the Web. doi:10.1145/2435215.2435218

5. Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, pages 157–166. doi:10.1109/72.279181

6. Chen, C. M., Lee, H. M., and Chang, Y. J. (2009). Two novel feature selection approaches for web page classification. Expert Systems with Applications. doi:10.1016/j.eswa.2007.09.008

7. Chen, C. M., Lee, H. M., and Tan, C. C. (2006). An intelligent web-page classifier with fair feature-subset selection. Engineering Applications of Artificial Intelligence. doi:10.1109/NAFIPS.2001.944285

8. Chen, R. C. and Hsieh, C. H. (2006). Web page classification based on a support vector machine using a weighted vote schema. Expert Systems with Applications. doi:10.1016/j.eswa.2005.09.079

9. Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. Syntax, Semantics and Structure in Statistical Translation.

10. Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. NIPS Deep Learning and Representation Learning Workshop.

11. Chung, Y., Toyoda, M., and Kitsugeregawa, M. (2010). Topic classification of spam host based on urls. In Proceedings of the Forum on Data Engineering and Information Management (DEIM).

12. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. Journal of Machine Learning Research 12, pages 2493–2537.

13. Dumais, S. and Chen, H. (2000). Hierarchical classification of web content. In Proceedings of SIGIR-00, 23rd ACM International Conference on Research and Development in Information Retrieval, pages 256–263. doi:10.1145/345508.345593

14. Hernandez, I., Rivero, C. R., Ruiz, D., and Corchuelo, R. (2014). Cala: An unsupervised url-based web page classification system. Knowledge-Based Systems. doi:10.1016/j.knosys.2013.12.019

15. Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), pages 1735–1780. doi:10.1162/neco.1997.9.8.1735

16. Kan, M. Y. (2004). Web page classification without the web page. Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers and Posters, pages 262–263. doi:10.1145/1013367.1013426

17. Kan, M. Y. and Thi, H. O. N. (2005). Fast web page classification using url features. In Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM '05), pages 325–326. doi:10.1145/1099554.1099649

18. Kim, Y. (2014). Convolutional neural networks for sentence classification. In Proceedings of EMNLP. doi:10.3115/v1/D14-1181

19. Kwon, O. W. and Lee, J. H. (2003). Text categorization based on k-nearest neighbor approach for web site classification. Information Processing and Management 39. doi:10.1016/S0306-4573(02)00022-5

20. Lai, Y. S. and Wu, C. H. (2002). Meaningful term extraction and discriminative term selection in text categorization via unknown-word methodology. ACM Transactions on Asian Language Information Processing (TALIP), pages 36–64. doi:10.1145/595576.595579

21. Lee, J. H., Yeh, W. C., and Chuang, M. C. (2015). Web page classification based on a simplified swarm optimization. Applied Mathematics and Computation. doi:10.1016/j.amc.2015.07.120

22. Liu, S., Yang, N., Li, M., and Zhou, M. (2014). A recursive recurrent neural network for statistical machine translation. In Proceedings of the Association for Computational Linguistics. doi:10.3115/v1/P14-1140

23. Nicolov, N. and Salvetti, F. (2007). Efficient spam analysis for weblogs through url segmentation. In RANLP, volume 292 of Current Issues in Linguistic Theory (CILT). doi:10.1075/cilt.292.17nic

24. Ozel, S. A. (2011). A web page classification system based on a genetic algorithm using tagged-terms as features. Expert Systems with Applications. doi:10.1016/j.eswa.2010.08.126

25. Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014). doi:10.3115/v1/D14-1162

26. Selamat, A. and Omatu, S. (2004). Web page feature selection and classification using neural networks. Information Sciences. doi:10.1016/j.ins.2003.03.003

27. Sun, A., Liu, Y., and Lim, E. P. (2011). Web classification of conceptual entities using co-training. Expert Systems with Applications. do:10.1016/j.eswa.2011.03.010

28. Sutskever, I., Vinyals, O., and Le, Q. (2014). Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems (NIPS 2014).

29. Tsukada, M., Washio, T., and Motoda, H. (2001). Automatic web-page classification by using machine learning methods. Web Intelligence: Research and Development, LNAI 2198, pages 303–313. doi:10.1007/3-540-45490-X_36

30. Vonitsanou, M., Kozanidis, L., and Stamou, S. (2011). Keywords identification within greek urls. Polibits 43, pages 75–80.