

# Dynamic Classification in Web Archiving Collections

Krutarth Patel, Cornelia Caragea, Mark E. Phillips

Kansas State University, University of Illinois at Chicago, University of North Texas

kipatel@ksu.edu, cornelia@uic.edu, Mark.Phillips@unt.edu

## Abstract

The Web archived data usually contains high-quality documents that are very useful for creating specialized collections of documents. To create such collections, there is a substantial need for automatic approaches that can distinguish the documents of interest for a collection out of the large collections (of millions in size) from Web Archiving institutions. However, the patterns of the documents of interest can differ substantially from one document to another, which makes the automatic classification task very challenging. In this paper, we explore dynamic fusion models to find, on the fly, the model or combination of models that performs best on a variety of document types. Our experimental results show that the approach that fuses different models outperforms individual models and other ensemble methods on three datasets.

**Keywords:** Web Archiving, Machine Learning, Deep Learning, Dynamic Classification

## 1. Introduction

A growing number of research libraries, museums, and archives around the world are embracing Web Archiving as a mechanism to collect born-digital material made available via the Web. Between the membership of the International Internet Preservation Consortium, which has 55 member institutions (Consortium, 2017), and the Internet Archive’s Archive-It Web Archiving platform with its 529 collecting organizations (Archive-It, 2017), there are well over 584 institutions currently engaged in building collections with Web Archiving tools. The amount of data that these Web Archiving initiatives generate is typically at levels that dwarf traditional digital library collections. As an example, in a recent impromptu analysis, Jefferson Bailey of the Internet Archive noted that there were 1.6 Billion PDF files in the Global Wayback Machine. If just 1% of these PDFs are of interest for collection organizations, that would result in a collection of 15+ million volumes in HathiTrust.

While the number of Web Archiving institutions increases, the technologies needed to provide access to these large collections have not improved significantly over the years. At this time, the standard way of accessing web archives is with known URL lookup using tools like the OpenWayback<sup>1</sup> or pywb<sup>2</sup>. The use of full-text search has increased in many web archives around the world, but often provides an experience that is challenging for users because of the vast amount of content and the limitations of strictly text-based searches for these large heterogeneous collections of content. Another avenue of access to web archived data that is of interest to Web Archiving institutions is the ability to extract high-quality, content-rich publications from the web archives in order to add them to their existing collections. In this paper, we explore machine learning and deep learning models to classify documents from Web Archiving collections into being in scope for a given collection (or collection policy) or out of scope. By identifying and extracting these documents, institutions will improve their ability to provide meaningful access to collections of materials harvested from

the Web that are complementary, but oftentimes more desirable than traditional Web archives.

In Web Archiving collections, usually, the documents are very diverse, with different types of documents having a different textual structure and covering different topics. As an example, consider a scholarly works repository, which contains publications that are typical for an institutional repository such as research articles, white papers, slide decks from presentations, and other scholarly publications. Documents not considered as part of the scholarly works repository include curriculum vitae, resumes, publications lists, and student manuals. The beginning and the end portions of a document might contain useful and sufficient information (either structural or topical) for deciding if a document is in scope of a collection or not. For example, research articles usually contain the abstract and the introduction in the beginning of the document, with the conclusion, acknowledgements, and references occurring towards the end of the document. Being on the proper subject (or in scope of a collection) can often also be inferred from the beginning and the end portions of a document.

Many traditional and neural network based approaches have been successfully explored for text classification. The “bag of words” (BoW) or *tf-idf* representation is commonly used for text classification (Caropreso et al., 2001; Sebastiani, 2002). Structural features (Str) designed based on the characteristics of a document have been successfully used to classify academic documents into six different categories such as papers, slides, theses, etc. (Caragea et al., 2016). In addition, for text classification tasks, Convolutional Neural Networks (CNNs) are extensively used in conjunction with word embeddings and achieve remarkable results (Goldberg, 2016; Johnson and Zhang, 2014; Kim, 2014).

However, we conjecture that simply using any of the above individual classifiers or combining them with static decision-level fusion (e.g., aggregating all three BoW, Str, and CNN classifiers’ confidences) may not always help in finding relevant documents to a particular repository from Web Archiving collections that contain a wide variety of documents. Figure 1 illustrates this phenomenon using a few examples sampled from one of our datasets (a scholarly

<sup>1</sup> <https://github.com/iipc/openwayback>

<sup>2</sup> <https://github.com/webrecorder/pywb>

Single classifier is correct				Two classifiers are correct			
Document	Class Label	Description	Prob.	Document	Class Label	Description	Prob.
	+ve	Slides	BoW: 0.15 Str: <b>0.74</b> CNN: 0.16 SEM: 0.35		+ve	Research article	BoW: 0.40 Str: <b>0.85</b> CNN: 0.63 SEM: 0.63
	(a)				(c)		
	-ve	CV	BoW: 0.55 Str: <b>0.40</b> CNN: 0.87 SEM: 0.61		-ve	Publication list	<b>BoW: 0.45</b> Str: 0.11 CNN: 0.97 SEM: 0.51
	(b)				(d)		

Figure 1: Example documents from a Web Archiving collection and classifiers’ confidences.

works repository). The figure contains the predicted probabilities for textual content based BoW and CNN classifiers, structural features classifier (Str), and static ensemble model (SEM) that averages the probability of each individual classifier (BoW, Str, and CNN, in our case). For the document (a) (slides) in Figure 1, only Str classifier correctly predicts the class label with a probability of 0.74. However, BoW, CNN and SEM miss-classified the document with very low probability of 0.15, 0.16, and 0.35, respectively. Similarly for document (b) (CV), only Str classifier correctly predicts the document as out of collection, while all others (BoW, CNN, and SEM) mistakenly classify the CV as being part of the repository. Interestingly, for the research article (document) (c), only BoW classifier miss-labeled the document with 0.40 probability, whereas Str, CNN, and SEM correctly classified the document. Similar observations can be seen for the document (d) (a publications list).

To this end, we explore dynamic decision-level classifier selection to identify on the fly the best suited classifier that can perform well on a given document. Our study includes an exploration of base classifiers including traditional machine learning models based on BoW (Caropreso et al., 2001; Sebastiani, 2002) and structural features (Str) (Caragea et al., 2016) that capture the characteristics of documents, as well as an exploration of convolutional neural networks (CNNs) for dynamically selecting the best classifier for specific document types on the fly. Our research focus is on three generic use cases that have been identified for the reuse of Web Archives and include populating an institutional repository from a Web archive of a university domain, the identification of state publications from a Web Archive of a state government, and the extraction of technical reports from a large federal agency. These three use cases were chosen because they have broad applicability and serve as a good test for classification using machine learning and deep learning models.

To our knowledge, mining Web Archiving collections is underexplored despite its importance in preserving the Web. As far as we know, our work is the first proposing a dynamic decision-level classifier selection model for finding relevant documents from these collections. In summary, our contributions are as follows:

- We built three datasets from three different Web

archives, each covering different domains: a scholarly works repository (UNT.edu), state publications (Texas.gov), and a federal agency publications (USDA.gov). Each dataset contains PDF documents along with their labels indicating whether a document is in scope of a collection or not. The datasets are available online<sup>3</sup> to further research in this area.

- We show that BoW and CNN classifiers that use only some portion of the documents outperform their counterparts that use the entire content of documents.
- We propose a dynamic classifier selection for document classification (DCSDC) to dynamically select an appropriate classifier to predict the probability of a target document as being in scope of a collection or not. To dynamically select the classifiers, we consider textual similarity along with the structural aspects of the documents.
- We show that DCSDC outperforms all the individual feature set models (base classifiers) and other strong baselines.

The rest of the paper is organized as follows: We summarize closely related work in Section 2. In Section 3, we describe three base classifiers used in our experiments: BoW classifiers, structural features classifiers, and CNN classifiers. Then, in Section 4, we discuss our proposed dynamic decision-level fusion approach for finding documents of interest from Web Archiving collections. We explain the process of creating the datasets in Section 5. We present our experimental setup and results in Section 6, followed by conclusions and future directions of our work in Section 7.

## 2. Related Work

**Web Archiving.** Web Archiving as a method for collecting content has been conducted by libraries and archives since the mid-1990’s. The most known Web Archiving is operated by the Internet Archive who began harvesting content in 1996. Recently, the Library of Congress (Dooley and Thomas, 2019) analyzed its Web Archiving holdings and has identified 42,188,995 unique PDF documents in its

<sup>3</sup> [https://www.cs.uic.edu/~cornelia/datasets/web\\_archive\\_data](https://www.cs.uic.edu/~cornelia/datasets/web_archive_data)

holdings. This shows interest in the PDF documents from the Web as being of interest to digital libraries. Thus, there is a high need for tools and techniques to help filter the desirable PDF content based on existing collections or collection development policies. Jacobs (2014) articulates the value and importance of web-published documents from the federal government that are often found in Web Archives. In this paper, we formulate the problem of classifying the PDF documents from a Web Archiving collection into being of scope for a given collection or not.

The Web Archiving can be used to gather documents or webpages belonging to a particular domain from the Web. Nwala et al. (2018) studied about bootstrapping the Web Archiving collections from the social media and showed that sources such as Reddit, Twitter, and Wikipedia produce collections that are similar to expert generated collections (i.e., Archive-It collections). McCoy et al. (2018) used the Twitter API for ranking 264 universities using two easily collected measurements as an alternative to university ranking lists published in U.S. News & World Report, Times Higher Education. To better understand a given Web Archiving collection, Alam et al. (2016) used CDX summarization for web archive profiling and AlNoamany et al. (2017) proposed the Dark and Stormy Archive (DSA) framework for summarizing and arranging these collections.

**Ensemble models and dynamic fusion.** Ensemble models have been used previously in many systems (Woźniak et al., 2014; Breiman, 1996; Skurichina and Duin, 1998). Bagging is an ensemble technique that builds a set of diverse classifiers, each trained on a random sample of the training data to improve the final (aggregated) classifiers' confidence (Breiman, 1996; Skurichina and Duin, 1998). Since the classifiers in an ensemble may learn very different patterns, dynamic ensembles that extend bagging have also been proposed (Cruz et al., 2018; Cruz et al., 2015; Cavalin et al., 2011). In dynamic ensembles, a pool of classifiers are trained on a single feature type (e.g., bag-of-words), each using a different subset of examples or features, within the bagging technique (Breiman, 1996; Skurichina and Duin, 1998) and the competence of the base classifiers is determined dynamically. Our work extends these approaches to dynamically select one of the various classifiers on the fly to perform document categorization.

Ensemble classifiers have also been used in a multi-modal setting (Guillaumin et al., 2010; Poria et al., 2016), in which different modalities are coupled, e.g., images and text for image retrieval (Kiros et al., 2014) and image classification (Guillaumin et al., 2010). Zahavy et al. (2018) urged the development of optimal unification methods to combine different classifiers trained on different modalities. Co-training approaches (Blum and Mitchell, 1998) use multiple views of the data to "guide" different classifiers in the learning process. However, co-training methods are semi-supervised and assume that all views are "sufficient" for learning. In contrast with the above approaches, we aim to capture different aspects of documents (structure and topicality), with each aspect having a different competence power, and perform dynamic selection of classifiers for classifying the textual documents from a Web Archiving collection into being in scope for the collection or not.

**Traditional Text Classification.** Text classification is a well-studied problem. The *BoW*, *binary*, *tf* or *tf-idf* representations are commonly used as input to machine learning classifiers, e.g., Support Vector Machine (Joachims, 1998) and Naïve Bayes Multinomial (McCallum and Nigam, 1998) for text classification. Feature selection is often applied to these representations to remove irrelevant or redundant features (Forman, 2003; Dumais et al., 1998). In the context of digital libraries, the classes for text classification are often document topics, e.g., papers classified as belonging to "machine learning" or "information retrieval" (Lu and Getoor, 2003; Caragea et al., 2015; Caragea et al., 2011). Caragea et al. (2016) proposed structural features for the classification of documents into six different categories, such as research papers, slides, and theses. These features extend the original structural features proposed for the automatic identification of research articles from crawled documents (Caragea et al., 2014).

Comprehensive reviews of the feature representations, methods, and results on various text classification problems are provided by Sebastiani (2002) and Manning (2008). Kodakateri Pudhiyaveetil et al. (2009) used the k-NN classifier to classify computer science papers into 268 different categories based on the ACM classification tree. Zhou et al. (2016) experimented with different classification methods such as unigram, bigram, and Sentence2Vec (Le and Mikolov, 2014) to identify the best classification method for classifying academic papers using the entire content of the scholarly documents.

**Deep Learning.** Deep learning models have achieved remarkable results in many NLP and text classification problems (Bengio et al., 2003; Collobert and Weston, 2008; Collobert et al., 2011; Mikolov et al., 2013; Kalchbrenner et al., 2014; Goldberg, 2016). Most of the works for text classification with deep learning methods have involved word embeddings. Among different deep learning architectures, Convolutional Neural Networks, Recurrent Neural Networks, and their variations are the most popular architectures for text applications. Kalchbrenner et al. (2014) proposed a deep learning architecture with multiple convolution layers and uses word embeddings initialized with random vectors. For text classification, Zhang et al. (2015) used encoded characters ("one-hot" encoding) as an input to the deep learning architecture with multiple convolution layers. Kim (2014) used a single convolution layer after extracting word embeddings for tokens in the input sequence. The author experimented with several variants of word embeddings, i.e., randomly initialized word vectors later tuned for a specific task, fixed pre-trained vectors or pre-trained vectors fine-tuned for a specific task, and a combination of two sets of word vectors. Zhang et al. (2016) and Yin et al. (2016) used the combination of diverse versions of pre-trained word embeddings followed by a CNN and a fully connected layer for the sentence classification problem.

### 3. Base Classifiers

Our goal in this paper is to study different types of features and learning models to accurately distinguish documents of interests from Web Archives for indexing in a specialized collection. In this section, we discuss different types of fea-

tures that are used in conjunction with traditional machine learning classifiers for finding documents of interests, and the CNN model that does not require any feature engineering. These models form our base classifiers.

### 3.1. Bag of Words (BoW)

BoW is a simple fixed-length vector representation of any variable length text based on the occurrence of different words within the text. It is called bag of words as the information about the positions of different words is discarded. First, a vocabulary from the words in the training documents is generated. Then, each document is represented as a vector based on the words in the vocabulary. The values in the vector representation are usually calculated as normalized term frequency (*tf*) or term frequency - inverse document frequency (*tf-idf*) of the corresponding words calculated based on the given document/text.

We extracted BoWs from the entire documents as well as using only some portions of documents. Our intuition behind using only some portion of the documents is that many types of documents contain discriminative words at the beginning and/or the end. For selecting these portions of documents, we considered first- $X$  words and first- $X$  words combined with last- $X$  words from each document before any preprocessing was performed (where  $X \in \{100, 300, 500, 700, 1000, 2000\}$ ). For documents with less than  $2 \cdot X$  words, we considered the entire document without repeating any parts/words from the document.

### 3.2. Structural Features

Structural features cover various structural aspects of documents and are grouped into four categories: file specific features, text specific features, section specific features, and containment features.

**File specific features** include the characteristics of a document such as the number of pages and the file size in kilobytes.

**Text specific features** include specifics of the text of a document: the length in characters; the number of words; the number of lines; the average number of words and lines per page; the average number of words per line; the count of reference mentions; the percentage of reference mentions, spaces, uppercase letters, symbols; the ratio of length of shortest to the longest line; the number of lines that start with uppercase letters; the number of lines starting with non-alphanumeric letters; the number of words that appear before the reference section.

**Section specific features** include section names and their position within a document. These features are boolean features indicating the appearance of “abstract”, “introduction”, “conclusion”, “acknowledgements”, “references” and “chapter,” respectively, as well as numeric features indicating position for each of these sections. These features also include two binary features indicating the appearance of “acknowledgment” before and after “introduction.”

**Containment features** include containment of specific words or phrases in a document. These features include binary features indicating the appearance of “this paper,” “this book,” “this thesis,” “this chapter,” “this document,” “this section,” “research interests,” “research experience,” “ed-

ucation,” and “publications,” respectively. These features also include three numeric features indicating the position of “this paper,” “this book,” and “this thesis” in a document.

### 3.3. Convolutional Neural Network (CNN)

CNNs (LeCun et al., 1998) are a special kind of neural networks. CNNs are associated with the idea of a “moving filter.” A convolution consists of a filter or a kernel that is applied in a sliding window fashion to extract features. The convolution layer consists of multiple filters of different region sizes that generate multiple feature maps for different region sizes. Pooling is usually used after the convolution layer to modify the output or reduce the dimensionality. The common practice is to extract the most important feature within each feature map (Collobert et al., 2011), called 1-max pooling. Max pooling is applied over each feature map and maximum values from all filter responses are selected. Maximum values from all feature maps are then concatenated and used as input to a fully connected layer for the classification task.

For CNN, we experimented by using the text from specific portions of a document. While selecting the portions of documents, as before, we considered first- $X$  words and first- $X$  words combined with last- $X$  words from each document before any preprocessing was performed (where  $X \in \{100, 300, 500, 700, 1000\}$ ). For the documents with less than  $2 \cdot X$  words, we considered the whole document without repeating any part/words from the document.

## 4. Proposed Model: Dynamic Classifier Selection

In contrast with the approaches that use a single model to identify relevant documents to a given collection, we propose an approach called “Dynamic Classifier Selection for Document Classification” (or DCSDC), that dynamically selects an appropriate classifier to identify if a given document is relevant to a collection by dynamically capturing different aspects of the document. The intuition behind using this approach is that there is a high variability in the type of the documents in each dataset.

The proposed approach consists of a three-step process for classifying a given document:

- **Step-1: Find its neighborhood documents.** We identify the neighborhood documents of the target document by considering textual similarity along with their structural aspects using K-Nearest Neighbors algorithm ( $K \in \{5, 10, 15, 20, 50\}$ ). In the first step, we consider the documents in the **Dev** set that fall under the range of  $X \pm L$  pages ( $X$  is the number of pages of the target document and  $L$  is a page range limit; we consider  $L = 3$ ). After that, we calculate the textual similarity between the target document and those documents from **Dev** that are within  $X \pm L$  page range from the target document. In order to calculate the textual similarity between the documents, we use two different methods by considering top  $N$  most frequent words ( $N \in \{25, 50, 100\}$ ): (a) tf-idf based cosine similarity, and (b) Word centroid based cosine similarity, where the centroid is calculated by a weighted average word

vectors. We consider pre-trained word embeddings trained on Google News for the word vectors.

- **Step-2: Find the most competent classifier.** Here we find the set of features and classifiers that perform best on neighborhood documents. The goal is to find the most competent classifier for a particular type of document. We apply each individual or the combination of different feature sets to neighborhood documents and find which classifier has the highest success rate for labeling them correctly.
- **Step-3: Use the most competent classifier on the test example.** The classifier with the highest success rate based on step-2 is used to classify the given test document. In case of multiple classifiers with the highest success rate, we selected the majority vote.

## 5. Data

For this research, we constructed three datasets from three Web archives collected by the UNT Libraries. For each of the datasets we extracted all PDF documents within each of the Web archives. Next, we randomly sampled 2,000 PDF files from each collection that we used as the basis for our labeled datasets. Each of the 2,000 PDF documents were then labeled in scope or out of scope by subject matter experts who are responsible for collecting publications from the Web for their collections. Each dataset includes PDF files along with their labels (in scope/out of scope or relevant/irrelevant). Further description of the datasets is provided below.

**UNT.edu:** The first dataset was created from the Web archive containing university scholarly works of the unt.edu domain. This archive was created in May 2017 as part of a bi-yearly crawl of the unt.edu domain by the UNT Libraries for the University Archives. A total of 92,327 PDF documents that returned an HTTP response of 200 are present in the archive. A total of 3,141,886 URIs are present in the entire Web archive with PDF content making up just 3% of the total number of URIs. Out of the 2,000 documents sampled for labeling, 445 documents (22%) were identified as being of interest for the repository and 1,555 not being of interest.

**Texas.gov:** The next dataset was created from a Web archive of websites that constitute the State of Texas web presence. The data was crawled from 2002 to 2011 and is housed as a collection in the UNT Digital Library. A total of 1,752,366 PDF documents that returned an HTTP response of 200 are present in the archive. A total of 26,305,347 URIs are present in the entire web archive with PDF content making up 6.7% of the total number of URIs. Out of the 2,000 documents sampled for labeling, 136 documents (7%) were identified as being of interest for the repository and 1,864 not being of interest.

**USDA.gov:** The last dataset created for this study comes from the End of Term (EOT) 2008 Web archive. This Web archive was created as a collaborative project between a number of institutions at the transition between the second term of George W. Bush and first term of Barack Obama. The entire EOT Web archive contains 160,212,141 URIs. For this dataset we selected the United States Department

of Agriculture (USDA) and its primary domain of usda.gov. This usda.gov subset of the EOT archive contains 2,892,923 URIs with 282,203 (9.6%) of those being PDF files that returned an HTTP 200 response code. Out of the 2,000 documents sampled for labeling, 234 documents (12%) were marked as of interest (technical reports) and 1,766 identified as not being of interest (not technical reports).

The three datasets represent a wide variety of publications that would be considered for inclusion into their target collections. The Texas.gov content is the most diverse as the publications range from strategic plans and financial audit reports (which are many pages in lengths) to pamphlets and posters (which are generally very short). The UNT.edu dataset contains publications that are typical for an institutional repository such as research articles, white papers, slide decks from presentations, and other scholarly publications. The publications from the USDA.gov dataset are similarly scoped as the UNT.edu content, but they also contain a wider variety of content that might be identified as a “technical report.” A goal in the creation of the datasets used in this research was to have a true representative sample of the types of content that are held in collections of this kind.

**Supplemental Datasets.** From the original datasets of 2,000 PDF files, we divided each dataset into three parts by randomly sampling training set (**Train-1**), development set (**Dev**), and test set (**Test**) from each dataset. All **Train-1**, **Dev**, and **Test** follow a similar distribution as the original datasets. Because the original datasets were very skewed, with only around 22%, 7%, and 12% of the PDF documents being part of the positive class, we asked the subject matter experts of each Web archive collection to further identify more positive examples to enlarge the training set. For the training set, we sampled from the newly labeled set of positive examples so that the number of negative examples is doubled as compared with the number of positive examples. We experimented with other ratios of positive to negative examples in the training set, but the 2:1 ratio showed better results. We denote this set as **Train-2**. Table 1 contains the summaries regarding the total number of documents in each dataset for which we were able to extract the text from a given PDF document. To extract the text from the PDF documents, we used PDFBox.<sup>4</sup> The scanned documents and other documents for which the text was not correctly extracted were ignored.

	UNT.edu		Texas.gov		USDA.gov	
	-ve	+ve	-ve	+ve	-ve	+ve
Train-1	869	250	981	72	907	121
Dev/Test	290	83	327	24	300	40
Train-2	869	434	981	490	907	453

Table 1: Datasets description.

## 6. Experimental Setup and Results

In this section, we first discuss the baselines and then the experimental setup for our document classification task. Next, we present the set of experiments to determine the base

<sup>4</sup> <http://pdfbox.apache.org/>

classifier hyper-parameters and to find which portion of a document to use on different classifiers. We then present an analysis to highlight the potential of the proposed algorithm. At last, we compare our proposed approach with the individual base classifiers and strong baselines.

## 6.1. Baselines

The baselines used for comparison are described below.

**1. KNN:** For a target document  $T$ , we calculate its  $K$ -nearest neighbors from the Train-2 set and then select the majority class label from the neighbors set. We experiment with  $K \in \{1, 5, 9, 15\}$  and select the best value of  $K$  using the **Dev** set. We used the Weka<sup>5</sup> implementation of KNN.

**2. Dynamic Ensemble Model (DEM):** We created this baseline by simplifying Step-2 and Step-3 of our proposed approach (Section 4.). We use the correctly classified neighborhood documents to calculate the score for each classifier and then use the scores for doing weighted sum of predicted probabilities of each base classifier similar to last stage of Step-3 in our approach as described below:

- **Find the score for each base classifier.** Here we compute the score for each base classifier by applying each base classifier to neighborhood documents. Then the score becomes:

$$Score = \frac{\#correctly\_classified\_neighbors}{\#neighbors} \quad (1)$$

- **Predict the class label for target document  $T$ .** We use weighted sum (similar to our main approach) of predicted probabilities of each base classifiers by using scores calculated as above as a weight for each base classifiers.

**3. Static Ensemble Model (SEM):** In this baseline, to make a final prediction, probability for each class label is averaged among all the base classifiers.

**4. Majority Vote:** We consider a majority vote as another baseline. We predict the document label by a label predicted by a majority of the base classifiers.

**5. Bagging (Skurichina and Duin, 1998):** In Bagging, a bag of classifiers, each trained on a random sample of examples from the training set are used to predict the class label for a test example. To predict the target document  $T$ , the individual class probabilities assigned by classifiers in the bag are averaged and the class label with the highest probability is selected. We used BoW classifier for the bagging.

**6. META-DES (Cruz et al., 2018):** Here, similar to bagging, a pool of classifiers are trained and then the competence or meta-classifier learning is performed to select the competent classifiers out of a pool of classifiers. The majority vote rule is applied over the selected competent classifiers. For META-DES, we used BoW classifiers to generate a pool of classifiers. Note that this baseline includes the competence learning component, but unlike our approach, it uses only one feature type (BoW).

## 6.2. Experimental Setup

As base classifiers, we experiment with the “bag of words” (BoW) extracted from the entire documents as well as from some portion of the documents, 43 structural features, and the convolutional neural networks (CNN). For the preprocessing step of the BoW, we remove stop words and punctuation, and perform stemming. In addition, we keep only words that appear in at least 5 documents.

For the traditional base classifiers, we experiment with several traditional machine learning classifiers: Naive Bayes (NB), Naive Bayes Multinomial (NBM), Random Forest (RF), and Support Vector Machines with a linear kernel (SVM). We used the Weka<sup>5</sup> implementation of these classifiers. Our CNN classifier comprises of mainly two layers: a convolution layer followed by a max pooling and a fully connected layer for the classification. For the CNN input, we consider a document (partial) as a sequence of words and use pre-trained word2vec. For CNN, we used TensorFlow.<sup>6</sup> We use the **Dev** set for the hyper-parameter tuning for the classifiers, and for the best classifier selection. In experiments, we tuned hyper-parameters for different classifiers as follows: the  $C$  parameter in SVM  $\in \{0.01, 0.05, 0.1\}$ ; the number of trees in RF  $\in \{20, 23, 25, 27, 30\}$ ; in CNN, single as well as three types of filters with region sizes  $\in \{1, 3, 4, 5, 7\}$ ; the number of each type of filters in CNN  $\in \{100, 128, 200\}$ . For CNN, we used ADAM optimizer with a learning rate of 0.0005.

**Evaluation Measures.** To evaluate the performance of classifiers, we use precision, recall, F1-score for the positive class, and the accuracy. All performance measures are averaged over the test set by repeating each experiment three times with three different seeds. We first discuss the results for the base classifiers in terms of F1-score using bar plots. Then we compare the proposed model with base classifiers and baseline models in terms of all the measures: precision, recall, F1-score, and accuracy in a table.

## 6.3. Experiments with Base Classifiers

Here, we report the performance of the base classifiers when we train on **Train-2** (2:1 distribution) and evaluate on **Dev**.

### 6.3.1. The Performance of the BoW Classifier

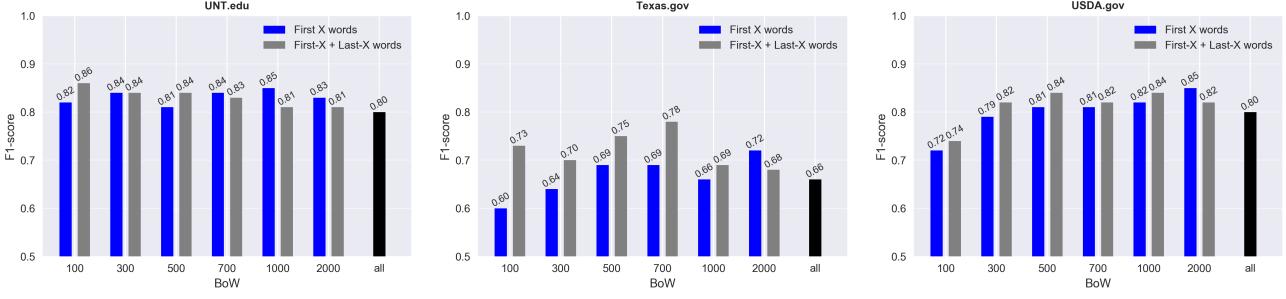
We compare the performance of the BoW classifiers in Figure 2 when we use different portions of the documents as well as the entire text of the documents. Random Forest performs better than any other classifier for the BoW features, and hence, we show the results using Random Forest. On the UNT.edu, the BoW using the first-100 words combined with last-100 words from each document performs best and achieves an F1 of 0.86. On the Texas.gov, the BoW classifier that uses the first-700 words combined with last-700 words performs best and achieves an F1 of 0.78. On the USDA.gov, the BoW classifier that uses the first-2000 words performs best and achieves an F1 of 0.85.

### 6.3.2. The Performance of the Str Classifier

Figure 4 shows the performance of the Str features extracted from the entire document for all three datasets. Random Forest classifier performs better than other classifiers for

<sup>5</sup> <https://www.cs.waikato.ac.nz/ml/weka/>

<sup>6</sup> <https://www.tensorflow.org/>

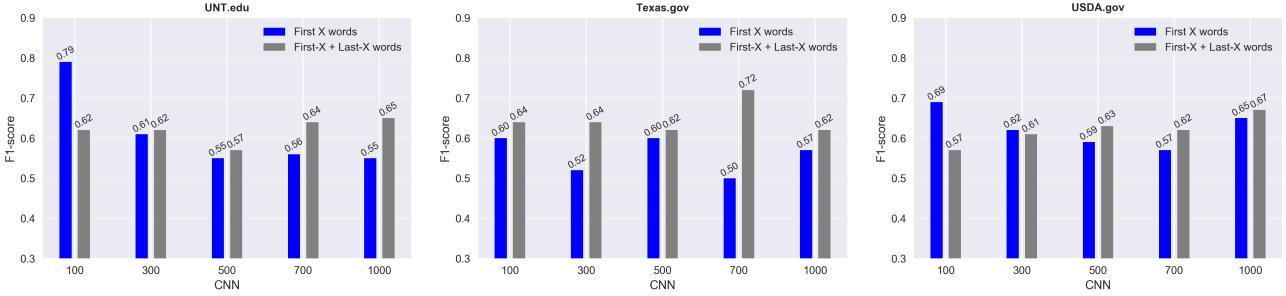


(a) UNT.edu dataset

(b) Texas.gov dataset

(c) USDA.gov dataset

Figure 2: Performance of BoW using different portions of the documents on different datasets



(a) UNT.edu dataset

(b) Texas.gov dataset

(c) USDA.gov dataset

Figure 3: Performance of the CNN using different portions of the documents on different datasets

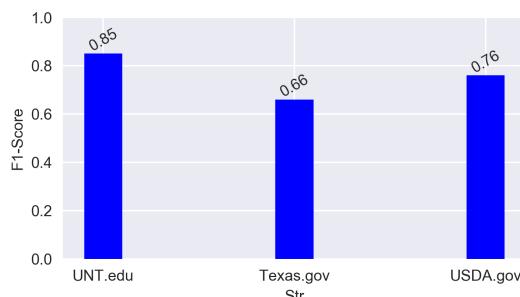


Figure 4: Performance of the Str classifiers using entire documents on different datasets

the Str features. The Str classifiers achieve an F1 of 0.85, 0.66, and 0.76 on UNT.edu, Texas.gov, and USDA.gov, respectively.

### 6.3.3. The Performance of the CNN Classifier

Next, we compare the performance of the CNN classifier when we consider the text from different regions of the documents in Figure 3. On the UNT.edu, the CNN classifier that uses the first-100 words from each document performs best and achieves an F1 of 0.79. On the Texas.gov, the CNN classifier that uses first-700 + last-700 words performs best and achieves an F1 of 0.72. On the USDA.gov, the CNN classifier that uses the first-100 words from each document performs best and achieves an F1 of 0.69.

## 6.4. Exploratory Analysis

We perform an exploratory analysis in Table 2 to highlight the potential of using our algorithm for selecting best classifier for classifying a document. We predict the class label for a given document by using different individual classifiers and obtain the coverage of the positive class (+ve) and

	UNT.edu (%)		Texas.gov (%)		USDA.gov (%)	
	+ve	All	+ve	All	+ve	All
BoW is correct	78	92	85	96	84	94
Str is correct	84	93	88	93	83	94
CNN is correct	80	90	83	94	78	91
All are correct	63	91	71	97	66	95
All are wrong	4	1.2	2.7	1	5.8	1.4
At least one	96	90	97	89	94	90

Table 2: Exploratory analysis.

the overall accuracy (All) for the following cases when: (1) an individual feature set is correct, (2) all feature sets are correct, (3) all feature sets are wrong, and (4) at least one feature set is correct. It can be seen from the table that the last row has the highest value regarding the coverage of the positive class (+ve) among all the datasets. This shows that there is room for improvement for the classifiers that use an individual feature set (first three rows), i.e., “At least one” covers 96%, 97%, and 94% of the positive examples as compared with 84% (Str), 88% (Str), and 84% (BoW) for the UNT.edu, Texas.gov, and USDA.gov, respectively. The large gap between the coverage of individual feature set (first three rows) and the “At least one” (last row) for the coverage of the positive class (+ve) showcases the potential of dynamically selecting a best performing classifier. Next, we evaluate the performance of our proposed DCSDC for the positive class.

## 6.5. Proposed Model vs. Individual Models and Baselines

We contrast the performance of our proposed model DCSDC with that of the three base classifiers (Section 3.: BoW, CNN, and Str), late fusion of the base classifiers

Classifier	UNT.edu				Texas.gov				USDA.gov			
	Pr(+)	Re(+)	F1(+)	Acc. (%)	Pr(+)	Re(+)	F1(+)	Acc. (%)	Pr(+)	Re(+)	F1(+)	Acc. (%)
BoW	0.87	0.78	0.82	92.4	0.64	0.85	0.73	95.7	0.75	0.84	0.79	94.7
Str	0.86	0.84	0.85	93.2	0.50	0.88	0.64	93.3	0.70	0.83	0.76	93.8
CNN	0.75	0.80	0.77	89.5	0.53	0.83	0.65	93.7	0.62	0.78	0.68	91.5
DCSDC <sub>3</sub>	0.88	0.85	0.87	94.2	0.69	0.88	0.77	96.5	0.77	0.86	0.81	95.2
BoW+Str	<b>0.94</b>	0.83	0.88	95.0	0.61	0.89	0.72	95.4	0.74	0.88	0.80	94.8
BoW+CNN	0.83	0.81	0.82	92.1	0.64	0.89	0.74	95.7	0.75	0.84	0.79	94.7
Str+CNN	0.88	<b>0.86</b>	0.87	94.2	0.65	0.93	0.77	96.1	0.76	0.85	0.80	95.0
BoW+Str+CNN	0.91	0.85	0.88	94.8	0.69	0.93	0.79	96.6	0.78	0.87	0.82	95.5
DCSDC <sub>7</sub>	<b>0.94</b>	<b>0.86</b>	<b>0.90</b>	<b>95.5</b>	<b>0.74</b>	<b>0.94</b>	<b>0.83</b>	<b>97.3</b>	0.81	<b>0.89</b>	<b>0.85</b>	96.3
KNN	0.85	0.17	0.29	80.9	0.34	0.49	0.35	88.5	0.51	0.12	0.19	88.6
DEM	0.93	0.84	0.88	95.1	0.59	0.88	0.70	95.0	0.74	0.88	0.80	94.9
SEM	0.91	0.85	0.88	94.8	0.69	0.93	0.79	96.6	0.78	0.87	0.82	95.5
Majority Vote <sub>3</sub>	0.90	0.83	0.86	94.1	0.70	0.88	0.78	96.6	0.76	0.85	0.80	95.0
Majority Vote <sub>7</sub>	0.91	0.85	0.88	94.8	0.69	0.93	0.79	96.6	0.78	0.87	0.82	95.5
Bagging	0.90	0.78	0.84	92.7	0.67	0.86	0.75	95.9	0.76	0.86	0.80	95.1
META-DES	0.92	0.82	0.86	94.2	0.73	0.85	0.78	96.8	<b>0.84</b>	0.88	<b>0.85</b>	<b>96.5</b>

Table 3: Performance of different features/models on our datasets.

(BoW+Str, BoW+CNN, Str+CNN, and BoW+Str+CNN) and six baselines (Section 6.1.: KNN, DEM, SEM, Majority Vote, Bagging, and META-DES) in terms of all compared measures, precision, recall and F1-score for the positive class, Pr(+), Re(+), and F1(+), and the overall accuracy of the classifier on the **Test** set. DCSDC<sub>3</sub> considers only the three base classifiers (BoW, CNN, and Str) and DCSDC<sub>7</sub> considers the three base classifiers along with their four late fusions (BoW+Str, BoW+CNN, Str+CNN, and BoW+Str+CNN). For DEM, SEM and Majority Vote, we experiment with considering only three base classifiers as well as base classifiers along with their late fusion, but the performance for DEM and SEM did not change. Majority Vote<sub>3</sub> and Majority Vote<sub>7</sub> indicate Majority Vote baseline by considering only three base classifiers and base classifiers along with their late fusion, respectively.

As we can see from Table 3, the DCSDC<sub>3</sub> outperforms the individual base classifiers (BoW, Str, and CNN). Moreover, we can see that the DCSDC<sub>7</sub> is the highest performing model across all three datasets in terms of all compared measures except the precision and accuracy on USDA.gov. On the other hand, the performance of the KNN classifier is worst as compared with all other compared classifiers.

On UNT.edu, Str outperforms the other two base classifiers, i.e., Str achieves an F1 of 0.85 as compared with 0.82 and 0.77 achieved by BoW and CNN, respectively. Late fusion of base classifiers outperforms the corresponding base classifiers, i.e., Str+CNN achieves an F1 of 0.87 as compared with 0.85 and 0.77 achieved by Str and CNN, respectively. KNN, DEM, and both Majority Vote baselines outperform individual base classifiers. DCSDC<sub>7</sub> achieves the highest values among all the measures. Furthermore, BoW+Str and Str+CNN also achieve highest precision and highest recall, respectively.

On Texas.gov, BoW outperforms the other two base classifiers, i.e., BoW achieves an F1 of 0.73 as compared with 0.64 and 0.65 achieved by Str and CNN, respectively. Late fusion of base classifiers outperforms the corresponding base classifiers except BoW+Str. All baselines except KNN,

and DEM outperform individual base classifiers. DCSDC<sub>7</sub> achieves again the highest values overall.

On USDA.gov, BoW outperforms the other two base classifiers, i.e., BoW achieves an F1 of 0.79 as compared with 0.76 and 0.68 achieved by Str and CNN, respectively. Late fusion of base classifiers outperforms the corresponding base classifiers, i.e., Str+CNN achieves an F1 of 0.80 as compared with 0.76 and 0.68 achieved by Str and CNN, respectively. All baselines except KNN outperform individual base classifiers. Surprisingly, META-DES achieves the highest recall, F1 and accuracy. DCSDC<sub>7</sub> achieves the highest values for the recall and F1.

## 7. Conclusion and Future Directions

In this paper, we proposed a dynamic decision-level fusion model for finding documents of interest from Web Archiving collections. We used a BoW classifier, a structural features-based classifier, and a CNN classifier as base classifiers. Experimental results show that BoW features extracted using only some portions of the documents outperform BoW features extracted using the entire document content. Thus, our conclusion was that the text from specific portions of documents is more useful than the text from the entire content for finding documents of interest to a collection. Furthermore, we proposed an approach that dynamically selects a classifier to identify if a document is relevant to a collection by dynamically capturing relevant aspects of the given document. Our experimental results show that the proposed dynamic selection approach performs better than the individual classifiers and other strong baseline models for finding documents of interest to a given collection as the collections may include documents of diverse types. In the future, it will be interesting to explore domain adaptation by training classifiers on one Web Archiving collection and testing on another Web Archiving collection.

## 8. Acknowledgements

This work is funded by the Institute of Museum and Library Services (IMLS) under award LG-71-17-0202-17.

## 9. Bibliographical References

- Alam, S., Nelson, M. L., Van de Sompel, H., Balakireva, L. L., Shankar, H., and Rosenthal, D. S. (2016). Web archive profiling through cdx summarization. *International Journal on Digital Libraries*, 17(3):223–238.
- AlNoamany, Y., Weigle, M. C., and Nelson, M. L. (2017). Generating stories from archived collections. In Proceedings of the 2017 ACM on Web Science Conference, pages 309–318. ACM.
- Archive-It. (2017). Archive-it homepage. <https://archive-it.org/>.
- Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *JMLR*, 3(Feb):1137–1155.
- Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT' 98, pages 92–100, New York, NY, USA. ACM.
- Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.
- Caragea, C., Silvescu, A., Kataria, S., Caragea, D., and Mitra, P. (2011). Classifying scientific publications using abstract features. In SARA.
- Caragea, C., Wu, J., Williams, K., Gollapalli, S. D., Khabsa, M., Teregowda, P., and Giles, C. L. (2014). Automatic identification of research articles from crawled documents. In Proceedings of the Workshop: Web-Scale Classification: Classifying Big Data from the Web, New York, NY.
- Caragea, C., Bulgarov, F., and Mihalcea, R. (2015). Co-training for topic classification of scholarly data. In EMNLP.
- Caragea, C., Wu, J., Gollapalli, S. D., and Giles, C. L. (2016). Document type classification in online digital libraries. In AAAI, pages 3997–4002.
- Caropreso, M. F., Matwin, S., and Sebastiani, F. (2001). A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. *Text databases and document management: Theory and practice*, 5478:78–102.
- Cavalin, P. R., Sabourin, R., and Suen, C. Y. (2011). Dynamic selection approaches for multiple classifier systems. *Neural Computing and Applications*, 22:673–688.
- Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th international conference on Machine learning, pages 160–167. ACM.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *JMLR*, 12(Aug):2493–2537.
- Consortium, I. I. P. (2017). Members. <http://netpreserve.org/about-us/members>.
- Cruz, R., Sabourin, R., Cavalcanti, G., and Ing Ren, T. (2015). Meta-des: A dynamic ensemble selection framework using meta-learning. *Pattern Recognition*, 48, 05.
- Cruz, R., Sabourin, R., and Cavalcanti, G. (2018). Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, 41, 05.
- Dooley, C. and Thomas, G. (2019). The library of congress web archives: Dipping a toe in a lake of data. <https://blogs.loc.gov/thesignal/2019/01/the-library-of-congress-web-archives-dipping-a-toe-in-a-lake-of-data/>.
- Dumais, S., Platt, J., Heckerman, D., and Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In CIKM, CIKM '98, pages 148–155.
- Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305.
- Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.
- Guillaumin, M., Verbeek, J., and Schmid, C. (2010). Multi-modal semi-supervised learning for image classification. In 2010 IEEE Computer society conference on computer vision and pattern recognition, pages 902–909. IEEE.
- Jacobs, J. A. (2014). Born-digital us federal government information: Preservation and access. *Leviathan: Libraries and Government Information in the Age of Big Data*.
- Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In Proc. of the 10th ECML, pages 137–142.
- Johnson, R. and Zhang, T. (2014). Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058*.
- Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.
- Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.
- Kiros, R., Salakhutdinov, R., and Zemel, R. (2014). Multimodal neural language models. In Proceedings of the 31st International Conf. on ML, volume 32, 22–24 Jun.
- Kodakateri Pudhiyaveetil, A., Gauch, S., Luong, H., and Eno, J. (2009). Conceptual recommender system for citeseerx. In RecSys, pages 241–244. ACM.
- Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In International Conference on Machine Learning, pages 1188–1196.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- Lu, Q. and Getoor, L. (2003). Link-based classification. In ICML.
- Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.
- McCallum, A. and Nigam, K. (1998). A comparison of event models for naive bayes text classification. In AAAI-98 Workshop on Learning for Text Categorization.
- McCoy, C. G., Nelson, M. L., and Weigle, M. C. (2018). Mining the web to approximate university rankings. *Information Discovery and Delivery*, 46(3):173–183.

- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pages 3111–3119.
- Nwala, A. C., Weigle, M. C., and Nelson, M. L. (2018). Bootstrapping web archive collections from social media. In Proceedings of the 29th on Hypertext and Social Media, pages 64–72. ACM.
- Poria, S., Cambria, E., Howard, N., Huang, G.-B., and Hussain, A. (2016). Fusing audio, visual and textual clues for sentiment analysis from multimodal content. *Neurocomput.*, 174(PA):50–59, January.
- Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.
- Skurichina, M. and Duin, R. P. (1998). Bagging for linear classifiers. *Pattern Recognition*, 31(7):909–930.
- Woźniak, M., Graña, M., and Corchado, E. (2014). A survey of multiple classifier systems as hybrid systems. *Information Fusion*, 16:3–17.
- Yin, W. and Schütze, H. (2016). Multichannel variable-size convolution for sentence classification. *arXiv preprint arXiv:1603.04513*.
- Zahavy, T., Krishnan, A., Magnani, A., and Mannor, S. (2018). Is a picture worth a thousand words? A deep multi-modal architecture for product classification in e-commerce. In AAAI. AAAI Press.
- Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In Advances in neural information processing systems, pages 649–657.
- Zhang, Y., Roller, S., and Wallace, B. (2016). Mgncnn: A simple approach to exploiting multiple word embeddings for sentence classification. *arXiv preprint arXiv:1603.00968*.
- Zhou, T., Zhang, Y., and Lu, J. (2016). Classifying computer science papers. In IJCAI.