Identifying Sustainability in the Private Sector Using Text Mining

Siemen Spinder[a], Flavius Frasincar[a,*], Vladyslav Matsiiako[a], David Boekestijn[a], Thomas Brandt[a]

[a]*Erasmus University Rotterdam, PO Box 1738, 3000 DR, Rotterdam, the Netherlands*

Abstract

Sustainability among companies is a topic that has gained enormous attention in recent years. Nonetheless, not much is known about the distribution of sustainable companies in the Netherlands. This paper explores text mining techniques to identify whether a company is sustainable or not by means of the information the companies provide on their websites. In addition, this paper introduces seven novel neural network architectures to perform this task. These make use of the hierarchical nature of Web domains, that are made up of layers of Web pages, sentences, and words. We either encode each of these layers by an attention mechanism, skip a layer, or use convolution layers. Among these models, the one that encodes pages, followed by a convolution layer appears to perform the best. It also performs better than both logistic regression and SVM approaches as well as convolutional neural networks. The attention model is used to gain insight into the distribution of sustainable industrial companies. The most sustainable activities identified are (i) production of chemical products, (ii) production of leather, leather products, and shoes, and (iii) production of food, with the least sustainable activities being (i) production of computers, (ii) printing, and (iii) production of cars, trailers, and semitrailers.

Keywords: company sustainability, neural networks, attention layers, hierarchical network, Web mining, classification

*Corresponding author; tel: +31 (0)10 408 1340; fax: +31 (0)10 408 9162

Email addresses: siemenspinder@hotmail.com (Siemen Spinder), frasincar@ese.eur.nl (Flavius Frasincar), matsiiako@gmail.com (Vladyslav Matsiiako), boekestijn@ese.eur.nl (David Boekestijn), tbrandt@student.eur.nl (Thomas Brandt)

1. Introduction

Recent years have seen the surge of a sustainability trend. Governments latch onto the topic, companies and academic journals publish sustainability reports, all showing goodwill [1, 2, 3, 4, 5]. According to a survey by KPMG [6], 75% of the N100 companies (the 100 largest companies for each of 49 countries) now report on corporate responsibility. The same holds for 93% of the Fortune 250 (the largest 250 companies in the world), though this statistic varies greatly between countries, with 82% of the 100 largest companies reporting in the Netherlands, but only 62% in Belgium.

[5] finds that high sustainability companies significantly outperform their counterparts over the long term, both in terms of stock market and accounting performance. In fact, [7] shows that the announcement of a step towards sustainability results in a positive jump in the stock prices of manufacturing companies, early adopters, and firms with large R&D spendings. Among the reporting of large multinationals, it is unclear how sustainable companies contribute to society economically, socially, and environmentally, and how sectors compare to each other in terms of sustainability. Such information may aid in guiding policy, with the possibility of rewarding companies that support a sustainable future and penalizing companies that do not. Moreover, [8] shows that various sustainability events receive much of financial analysts' attention and result in an increase in the number of shares held by long-term investors. This indicates that over time, professional investors pay more attention to firms with visible corporate social responsibility.

By classifying individual companies, the geographic spread of sustainability over a country can be found. To find a subset of sustainable companies, we may use indicators such as the amount of waste a company produces or what percentage of resources is reused. However, finding such information for small companies may be impractical, as these companies have no obligation to report on these values and it is costly to measure the values ourselves. Therefore, this paper proposes a different approach: use of text mining to identify sustainable companies. The vocabulary of a sustainable company can be inferred by looking at their websites; phrases such as 'biomass', 'solar panels', or 'sustainability' come to mind. This research focuses on finding such phrases and using them to classify a large number of companies. The research question is therefore:

How are sustainable companies in the private sector identified by means of text mining?

Sustainability is broad. E.g., the United Nations has set 17 Sustainable Development Goals (SDGs) [9] to establish priority areas for international development. These include 'quality education', 'gender equality', and 'peace and justice'. However, as the SDGs are diverse, proper classification is difficult. Therefore, the focus of this paper is on ecological sustainability, a narrower definition of sustainability closer to the layman's definition.

This narrow definition of sustainability allows for the use a particular set of companies as training data, namely those from `allesduurzaam.nl`, a website focused on collecting information of sustainable Dutch companies. A company listed on `allesduurzaam.nl` adheres to one of four main criteria: the company (i) tries to lessen its ecological footprint, (ii) closes the material cycle, e.g., by recycling, (iii) uses or generates renewable energy, or (iv) contributes to natural development or awareness.

The website `allesduurzaam.nl` lists 18,000 companies. We scrape the websites of the roughly 4,500 companies for which a website was available. Additionally, we scrape the websites of 4,500 companies with similar activities, but which are not listed on `allesduurzaam.nl`. For the purposes of this paper, these companies are therefore regarded as not sustainable. After annotating, we are left with 2,400 companies that are deemed to be sustainable and 2,400 that are not.

There is currently a good deal of research surrounding the use of text mining to obtain 'actionable' knowledge, such as, for instance, by [10]. In our paper, we make use of text classification, which is generally performed in two steps: (i) converting the text into informative quantitative features, and (ii) classifying the text. The first step has traditionally been performed by means of TF-IDF weighting, a weighting scheme where each word is assigned a score depending on its frequency in the document itself and the frequency of the word in the complete corpus of documents. TF-IDF weighting was, for instance, implemented as one of the techniques applied in the research of [11]. An alternative was proposed by [12], who embedded words into vectors by looking at the context of the word. Such embeddings incorporate syntactic information, which TF-IDF weighting is not able to capture. For the second step, classification machine learning algorithms are used. Common algorithms in text classification are Naive Bayes, Classification Trees, Random Forests, and Support Vector Machines (SVMs) [13].

Apart from these classification methods, recent years have seen an enormous growth in the use of neural networks for text classification [14, 15, 16]. A neural network is a machine learning model based on the human brain that processes data through a number of layers, each consisting of so-called neurons, the cells of the neural network. The hidden cells apply weights and

non-linear activation functions. Weights are traditionally optimized through a process called backpropagation, which calculates and uses gradients in a backwards fashion to move weights towards a more optimal value. Particular neural network architectures such as the convolutional neural network (CNN) [17] and the Bi-LSTM [18] have shown top performance in text classification [14].

However, according to [19], Recurrent Neural Networks (RNNs) such as the Bi-LSTM struggle with large sequences. This led [19] to introduce attention networks in text mining, networks that further emulate the human brain by focusing only on particular aspects of images or text, which was shown to give further benefits over the regular Bi-LSTM approach [20]. Such neural networks have been applied to texts mostly of a (semi-)structured and homogeneous nature. One may think of IMDB or Yelp reviews [15]. Our paper contributes to the literature by applying these neural network architectures on a noisier type of data, namely that of complete websites. Because it is impractical to scan every website manually, we use an attention mechanism to aid in the reliable identification of the salient details in the data, as was also done in e.g. [21].

In addition, we explore the hierarchical nature of text by implementing hierarchical attention networks described by [15]. Sentences are made up of words and documents are made up of sentences. Therefore, by first applying attention to particular words and then to sentences, further accuracy improvements can be achieved. However, the approach of [15] only considers two hierarchical layers, i.e., applying attention at word level and sentence level. In the particular case of Web domains, we can identify an additional level: the Web page level. We propose seven novel architectures that use this level. Although every level could simply be encoded by means of attention layers, some of the proposed architectures replace attention layers with CNN layers, as we hypothesize there is not always a logical ordering between components like Web pages. In this case, convolutional layers may more reliably encode the Web page level.

It appears that the best attention model is the one that encodes Web pages first by means of attention, followed by a convolution layer. This agrees with previous literature, such as [15], who suggest attention models can outperform simpler neural networks and provide more interpretation through attention weights, which correspond well with coefficients from traditional approaches. However, the difference in accuracy between attention models and convolutional neural networks is not as large as suggested by [15].

To show the efficacy of the best model, this model is applied to a separate unannotated data set: all companies that could be scraped in the produc-

tion industry. This list was obtained from Statistics Netherlands (CBS). We find that about 1,199 of 5,919 companies are sustainable. The most sustainable activities are deemed to be (i) production of chemical products, (ii) production of leather, leather products, and shoes, and (iii) production of food. The least sustainable activities are (i) production of computers and electronic and optical equipment, (ii) printing, and (iii) production of cars, trailers and semitrailers. Provinces with a high number of sustainable companies are Utrecht and Zeeland, while Noord-Brabant and Limburg relatively underperform.

All methods in this paper were implemented in Python, as it excels at text mining through packages such as *NLTK*, *gensim*, and *jusText*, and offers a plethora of packages related to deep learning, like *TensorFlow* and *Keras*. These packages will be expanded on in later sections.[1]

The remainder of this paper is structured as follows. Section 2 provides an overview of relevant literature. Section 3 shortly describes the websites considered in this paper and the way information can be obtained from them. Section 4 explains in detail existing ways to quantitatively represent text as well as methods for classification. Additionally, this section introduces our novel architectures for text classification. Results are presented in Section 5, containing model comparisons as well as an application to the considered branch of industry. Section 6 concludes this paper and provides suggestions for future work.

2. Literature

A comprehensive comparison of traditional machine learning methods used to categorize text is given by [13]. Among others, he describes Naive Bayes, C4.5, logistic regression, k-NN, and SVM. He compares these methods on 5 different datasets, all of which are related to news. He concludes that Adaboost using a one-level decision tree, k-NN, logistic regression, and SVM perform best, with one being better than the other depending on the dataset. Naive Bayes, popular in text classification, performs relatively poorly.

[14] provides a comparison of commonly used neural network methods for classification and uses them to classify the sentiment of 25,000 IMDB movie reviews. Among the standard network architectures, they use a network with a single LSTM layer, a network with a bidirectional LSTM layer, and a simple

[1]The scripts for this research are made publicly available at https://github.com/SiemenSpinder/HANVariants.

CNN. Additionally, they advocate in favour of two mixed networks: one where the CNN output is used as input for an LSTM layer and one where the bidirectional LSTM output is used as input for a CNN. The latter method, the Recurrent Convolutional Neural Network (RCNN), is based on work by [22]. RCNN performs best in both papers. [22] compares the methods to traditional approaches and shows that neural network architectures generally perform better.

[23] describes how the concept of attention may be used in text analysis. Attention in neural networks refers to putting more focus on certain aspects of an image or text. It was first proposed in image recognition [24] to detect salient elements, but [23] noted that it could also be used in neural machine translation, i.e., the use of neural networks to automatically translate sequences of text. The attention architecture showed a large improvement over a more conventional RNN encoder-decoder architecture such as the one mentioned by [25].

[15] provides the core reference work of this paper. Using the hierarchical structure of documents, the authors achieve (i) increased transparency of neural networks and (ii) improved accuracy. Building on attention models such as the one proposed by [23], the authors first create sentence representations by means of an RNN and an attention layer. They then input the sentence representation into another RNN and another attention layer to get a document representation. This representation is used as input for a final softmax layer to classify the current document. The attention weights from the separate attention layers can be taken to determine what words in the sentences and what sentences in the documents are most informative for classification.

Our paper has a number of significant improvements with respect to [15]. We build on their work by adding an additional layer and replacing some of the attention layers by convolutional layers. This layer should be appropriate to capture the Web page level that is not present in non-Web documents. The convolutional layers should perform better than attention layers when there is no temporal component in the data, which is possibly true for both the sentence layer and the Web page layer, as they do not necessarily have a (chrono)logical ordering. In the case of websites, sentences are often put separately from each other without being in a larger paragraph, while Web pages are better imagined in terms of a tree with the leaves as separate entities.

3. Data

The training data used in this paper concerns websites of companies that are labelled as either sustainable or not sustainable. The initial list of sustainable companies was obtained from `allesduurzaam.nl` and comprises 18,000 companies. As many of these companies did not have URLs listed on `allesduurzaam.nl`, we were left with a total of 4,427 sustainable companies after scraping and parsing. For their complement set, we ensure the companies are similar to their sustainable counterparts apart from the sustainability aspect, to prevent models recognizing differences in irrelevant characteristics. Ignoring this stage of data processing might result in words such as 'shop' ending up as important features if the list of sustainable companies has a large share of shops. To tackle this problem we find the Standard Business Indicator (SBI) codes of the sustainable companies, which expresses the activity of the company by matching zip code, house number, and URL of the companies to those in a dataset employed by CBS. We sample about 12,000 companies from the General Business Register, a list of all companies in the Netherlands, with the same proportions of activities as in the sustainable companies set. SBIs were found for a third of the companies. Thus, after scraping and parsing, 4,219 of these unlabeled companies are left.

Both the unlabeled list and the sustainable list were annotated by a group of annotators. Initially, it was assumed that the sustainable list consisted of only sustainable companies. This was refuted later on, causing the unlabeled list to be annotated by four annotators and the sustainable list by three annotators. The annotators consisted of employees of CBS and university students of different backgrounds, with ages ranging from 22 to 60. The annotators were presented the main sustainability criteria found on `allesduurzaam.nl` and asked to fill in either a 1 if the company is sustainable, a 0 if this is not the case, and a cross if the website was offline. After annotating, the companies determined to be sustainable by all annotators were put in the sustainable set, while the companies that were determined not to be sustainable by all annotators were put in the other set. All other companies are removed. Before removal, Fleiss' kappa was calculated, which measures the degree of inter-annotator agreement.

[26] describes ways to interpret the kappa values. The authors suggest a kappa value above 0.7 can be considered good. Because we had more annotators for the unlabeled set, we calculated two separate kappas. In our case, Fleiss' kappa was found to be equal to 0.52 in the case of the sustainable list and 0.37 for the non-sustainable one. According to [26], this corresponds to fair, but unsubstantial agreement.

The percent agreement is found to be 0.80 for both sets. The large discrepancy between Fleiss' kappa and the percent agreement can be explained by the fact that Fleiss' kappa heavily penalizes in the case of skewed data [27], e.g., data that contains a large number of non-sustainable companies but almost no sustainable companies. This is a problem in our case as the lists were annotated separately and we can therefore only compute a statistic on two skewed data sets. [27] proposes an alternative to Fleiss' kappa that is more representative of the quality of data: Gwet's AC1. Gwet's AC1 replaces the expression for the probability due to chance.

For our data, we find an AC1 of 0.67 for the sustainable list and 0.70 for the non-sustainable list. This corresponds to substantial agreement. After filtering, we are left with 2,340 sustainable companies and 2,408 non-sustainable ones, for a total of 4,748 companies.

3.1. Scraping and Parsing

The process of scraping, parsing, cleaning and saving the data is depicted in Fig. 1. By saving raw .html, the data can be used with other parsers in future research.

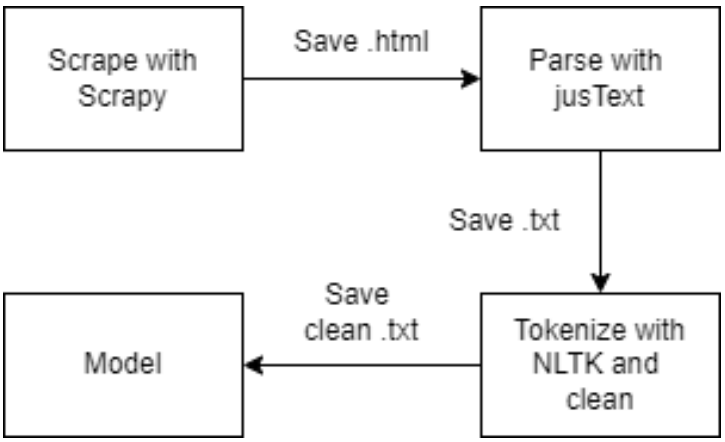

Figure 1: Obtaining the data.

To scrape the relevant websites, *Scrapy* is used, a Python library constructed to efficiently scrape and crawl a large number of websites. Scrapy uses so-called spiders, classes that define how to scrape, crawl, and process websites. One of the generic spiders supplied by Scrapy is the CrawlSpider, which enables one to efficiently crawl by defining a set of rules. Unlike other libraries, Scrapy incorporates features such as retrying requests, filtering

duplicated requests, and auto-throttling. Although Scrapy enables one to scrape only specific parts of websites by using CSS and XPath selectors, we write the complete .html file to disk to get as much information as possible. Scrapy can be used in conjunction with extensions such as Selenium and light headless browsers to get both standard content and JavaScript elements, but this severely slows down the scraping and preliminary analysis showed it does not provide much more actual text. Therefore, we limit ourselves to the regular content.

After obtaining the .html files, they are parsed using *jusText* [28], a Python library aimed at boilerplate removal, such as headers or navigation links. An alternative is *BeautifulSoup*, a library designed to turn .html content into a parse tree, from which one can easily get desired information by employing functions such as *.find.all(p)*, which finds all content between <p> tags, and *.get_text()*, which takes all text, including titles. However, *jusText* has the benefit of detecting the quality of content and whether a block of text is of a specified language. *jusText* was made by inspecting which block-level html tags, such as h1, p, or ul, often included non-grammatical text. After an initial division into 'good' and 'bad' tags, a machine learning model was made to classify the remaining tags based on surrounding tags. Remaining text is usually well-written, reducing noise considerably.

The text is tokenized by means of the Python package *NLTK*. *NLTK* can split sequences of characters in either words or sentences, after which the separate words can be cleaned. Some general steps are always taken to

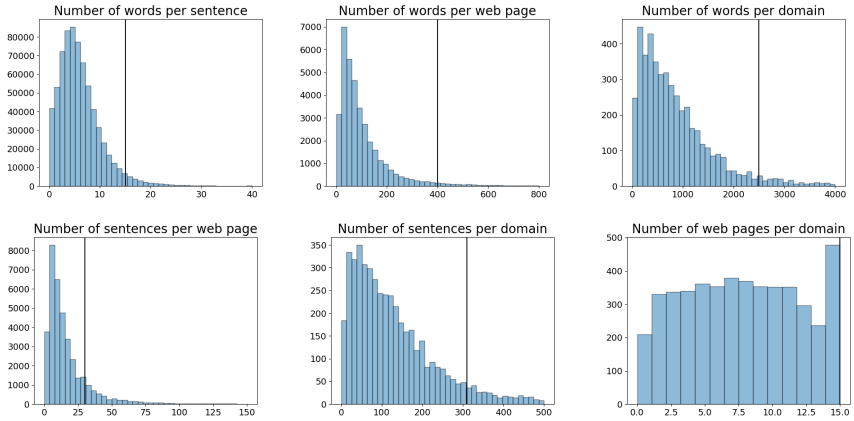

Figure 2: Histograms regarding (i) words per sentence, page and domain, (ii) sentences per page and domain, and (iii) pages per domain.

clean textual data. These are (i) making all tokens lowercase, (ii) removing non-alphanumeric tokens, (iii) removing tokens with a frequency below 2, and (iv) removing *stopwords* (tokens that are frequent but uninformative, such as 'the' and 'and'). The list of stopwords is constructed as a concatenation of a general Dutch stopword list by [29] and custom stopwords that are obtained after manually assessing the most informative features from a Logit model, based on coefficients (with TF-IDF scores as word representations).

3.2. Descriptive Statistics

Fig. 2 shows histograms of respectively the number of (i) words per sentence, page and domain, (ii) sentences per page and domain, and (iii) pages per domain. For efficient computation, neural networks require the same length of input. To achieve this, one can either pad the remainder of a sequence (sentence, page, or domain) with zeros, or cut the sequence. We decided to cut the sequences in such a way that a large part of every sequence is taken into account, while ensuring the sequences are not too long. This allows for the use of a GPU, which, although it has a limited amount of Video RAM (4 gigabytes), is more capable at matrix computations than a CPU because of its higher number of cores. In addition, we limit ourselves to only the 30,000 most frequent terms. The cut-off points are shown as lines in Fig. 2 and correspond roughly (but not exactly) to an adaptation of Tukey's fence, i.e., the median plus 1.5 times the interquartile range (IQR). Higher or lower values than the adapted Tukey's fence were chosen to optimize performance given the limited amount of Video RAM.

Tables 1 and 2 provide basic statistics of the text data used. The total number of unique words is comparable to that of the Yelp 2013 data set [15], but considerably lower than the vocabulary sizes of 1.5M and 1.9M of the Yahoo Answer and Amazon Review data sets [15], respectively. The total number of documents, 4,748, is lower than the number of documents in [15], as their data sets range between 0.3M and 3.7M documents. The low number of documents can be disadvantageous to our model, but the binary classification performed in our paper should be simpler than classification into 5 or 10 classes, as was done in [15].

Table 1: Total number of domains, Web pages, sentences, words, and unique words.

Domains	Pages	Sentences	Words	Unique words
4,748	36,905	700,208	4,561,395	217,373

The median number of words per sentence, 6, seems low, but could be caused by the fact that Web pages often contain simple sentences such as 'look here'. Note that the data sets in [15] contain, on average, 15-25 words per sentence. The median number of sentences per domain is 106, considerably higher than the roughly 5-15 sentences per review for the Yelp, Amazon, IMDB and Yahoo data sets. The higher number of sentences, but lower number of words per sentence results in an average number of words per domain of about 1,000, which is 3-11 times higher than the aforementioned data sets. This can be advantageous for the accuracy of our models.

Table 2: Descriptive statistics of text characteristics.

	Avg.	Std.	Med.	IQR	ATF[2]	Cut-off	Max.
WPS[1a]	6.5	4.7	6	4	12	15	226
WPP[1b]	123.6	276.2	73	95	215.5	400	26,747
WPD[1c]	960.7	1,168.4	688	867.3	1988.88	2500	27,481
SPP[1d]	19.1	40.9	11	15	33.5	30	3,724
SPD[1e]	147.5	176.1	106	133	305.5	310	3,810
PPD[1f]	7.8	4.0	8	7	18.5	15	15

1: (a) Words Per Sentence; (b) Words Per Page;
(c) Words Per Domain; (d) Sentences Per Page;
(e) Sentences Per Domain; (f) Pages Per Domain.
2: Adapted Tukey's Fence.

4. Identifying Sustainable Companies

The following sections propose a new framework to detect sustainable companies by means of the content on their websites. Based on the work of [15] we construct hierarchical neural networks that utilize the three layers contained in every Web domain, i.e., Web pages, sentences, and words. The architectures vary in the way they use respectively attention layers and convolutional layers.

4.1. Hierarchical Attention Networks

[15] describe a neural network in which words are first encoded as word vectors and attention is applied to specific words. Afterwards, complete sentences are encoded as sentence vectors and attention is applied to these. Fig. 3 gives an overview of such a network.

Figure 3: Hierarchical Attention Network.

We denote with w_{it} the one-hot encoding of the t-th word in sentence i. Then an embedding layer is applied, usually pretrained by means of Word2Vec, such as proposed by [12], or FastText, such as proposed by [30]. The resulting vectors can be calculated as:

$$\begin{aligned} &\mathbf{x}_{it} = \mathbf{W}_e w_{it},\ t \in [1, T] \\ &\mathbf{x}_{it} \in \mathbb{R}^{d_{emb}},\ \mathbf{W}_e \in \mathbb{R}^{d_{emb} \times d_{hw}}, \end{aligned} \tag{1}$$

where $\mathbf{W}_e$ are d_{emb} dimensional embedding weights obtained from Word2Vec or FastText. The embedded word, $\mathbf{x}_{it}$, can then be input into an encoder to obtain a representation of a sentence. A common encoder is the Gated Recurrent Unit (GRU) [25], that is defined as in (2).

$$\begin{aligned} \mathbf{h}_t &= (\mathbf{1} - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t \\ \mathbf{z}_t &= \sigma(\mathbf{x}_t \mathbf{W}^{xz} + \mathbf{h}_{t-1} \mathbf{W}^{sz}) \\ \mathbf{r}_t &= \sigma(\mathbf{x}_t \mathbf{W}^{xr} + \mathbf{h}_{t-1} \mathbf{W}^{sr}) \\ \tilde{\mathbf{h}}_t &= \tanh\left(\mathbf{x}_t \mathbf{W}^{xs} + (\mathbf{r} \odot \mathbf{h}_{t-1}) \mathbf{W}^{sg}\right) \\ \mathbf{h}_t, \tilde{\mathbf{h}}_t &\in \mathbb{R}^{d_s};\ \mathbf{x}_t \in \mathbb{R}^{d_x};\ \mathbf{z}_t, \mathbf{r}_t \in \mathbb{R}^{d_s}; \\ \mathbf{W}^{\mathbf{x}\circ} &\in \mathbb{R}^{d_x \times d_s};\ \mathbf{W}^{\mathbf{s}\circ} \in \mathbb{R}^{d_s \times d_s}. \end{aligned} \tag{2}$$

The GRU consists of several gates that are used to combat the vanishing gradient problem as described by [31]. This leads to an efficient encoding for a sequence and can be applied to words in both reading order and reverse reading order, leading to a forward representation $\overrightarrow{\mathbf{h}_{it}}$ and a backward representation $\overleftarrow{\mathbf{h}_{it}}$ of the word w_{it}. Combined, these give word representations $\mathbf{h}_{it}$.

$$\begin{aligned} \overrightarrow{\mathbf{h}_{it}} &= \overrightarrow{\mathrm{GRU}}(\mathbf{x}_{it}),\ t \in [1, T] \\ \overleftarrow{\mathbf{h}_{it}} &= \overleftarrow{\mathrm{GRU}}(\mathbf{x}_{it}),\ t \in [T, 1] \\ \mathbf{h}_{it} &= [\overrightarrow{\mathbf{h}_{it}}, \overleftarrow{\mathbf{h}_{it}}]. \end{aligned} \tag{3}$$

Afterwards, the first attention layer is applied. As described by [19], attention emulates human behaviour, where focus is put onto specific aspects of an image or text. In addition to $\mathbf{h}_{it}$, a source context vector $\mathbf{u}_w$ is trained that tells how informative words are. This layer ensures only informative

words are chosen to create a sentence vector $\mathbf{s}_i$.

$$\begin{aligned} \mathbf{u}_{it} &= \tanh(\mathbf{W}_w \mathbf{h}_{it} + \mathbf{b}_\mathbf{w}) \\ a_{it} &= \frac{\exp(\mathbf{u}_{it}^T \mathbf{u}_w)}{\sum_t \exp(\mathbf{u}_{it}^T \mathbf{u}_w)} \\ \mathbf{s}_i &= \sum_t a_{it} \mathbf{h}_{it} \end{aligned} \tag{4}$$

$$\mathbf{h}_{it}, \mathbf{s}_i \in \mathbb{R}^{d_{hw}};\ \mathbf{u}_{it}, \mathbf{u}_w, \mathbf{b}_w \in \mathbb{R}^{d_{uw}};\ \mathbf{W}_w \in \mathbb{R}^{d_{uw} \times d_{hw}},$$

where d_{uw} and d_{hw} correspond to, respectively, the dimensions of the input representation $\mathbf{u}_{it}$ and of the hidden state $\mathbf{h}_{it}$. The symbol w refers to the fact that words are being encoded at this stage. The resulting sentence embedding is again input into a bidirectional GRU, this time resulting in a sentence representation $\mathbf{h}_i$.

$$\begin{aligned} \overrightarrow{\mathbf{h}_i} &= \overrightarrow{\mathrm{GRU}}(\mathbf{s}_i),\ i \in [1, L] \\ \overleftarrow{\mathbf{h}_i} &= \overleftarrow{\mathrm{GRU}}(\mathbf{s}_i),\ i \in [L, 1] \\ \mathbf{h}_i &= [\overrightarrow{\mathbf{h}_i}, \overleftarrow{\mathbf{h}_i}]. \end{aligned} \tag{5}$$

Afterwards, another attention layer is applied, which selects informative sentences rather than informative words and results in one final document vector $\mathbf{v}$.

$$\begin{aligned} \mathbf{u}_i &= \tanh(\mathbf{W}_s \mathbf{h}_i + \mathbf{b}_\mathbf{s}) \\ a_i &= \frac{\exp(\mathbf{u}_i^T \mathbf{u}_s)}{\sum_t \exp(\mathbf{u}_i^T \mathbf{u}_s)} \\ \mathbf{v} &= \sum_t a_i \mathbf{h}_i \end{aligned} \tag{6}$$

$$\mathbf{h}_i, \mathbf{v} \in \mathbb{R}^{d_{hs}};\ \mathbf{u}_i, \mathbf{u}_s, \mathbf{b}_s \in \mathbb{R}^{d_{us}};\ \mathbf{W}_s \in \mathbb{R}^{d_{us} \times d_{hs}},$$

where d_{us} and d_{hs} correspond to respectively the dimensions of the input representation $\mathbf{u}_i$ and of the hidden state $\mathbf{h}_i$. The symbol s refers to the fact sentences are being encoded at this stage. The document vector can be used to classify a document by feeding it into a softmax layer or, in the case of binary classification, a sigmoid layer, and using an appropriate loss function which is usually chosen to be cross-entropy for classification. The added benefit of the hierarchical structure is that we can not only find the informative words in a document, but also the informative sentences, giving even more interpretation. In addition, [15] report a 3% accuracy improvement with respect to the second-best model, the LSTM-GRNN by [32]. This makes the Hierarchical Attention Network (HAN) one of the best performing models in document classification.

4.2. Proposed Architectures

Web domains, unlike documents, have more than just 2 hierarchical layers, as every domain is made up of Web pages, every Web page is made up of sentences and every sentence is made up of words (Fig. 4). To utilize this, we introduce novel 3-layer approaches. All these models utilize attention network (AN) encodings such as the one in (4), but at different levels. Attention can

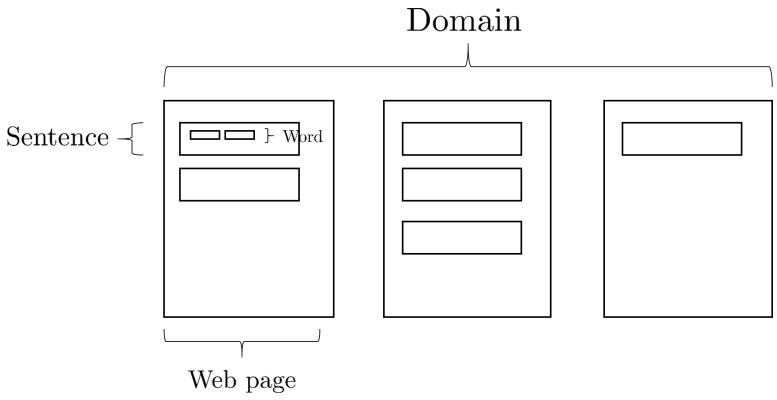

Figure 4: Structure of Domains.

be used to obtain three kinds of encodings: sentence encodings, Web page encodings, and domain encodings. There is just one way to obtain sentence encodings: by applying attention at the word level as in (4). To obtain Web page encodings, we can take two approaches: take sentence encodings and apply attention to them, or directly apply attention at the word level for the entire Web page. Domain encodings have the most options: we can apply attention to both kinds of Web page encodings, we can apply attention directly to the sentence encodings, omitting the Web page encodings, or we can even directly apply attention at the word level. In total, this gives 7 types of encodings.

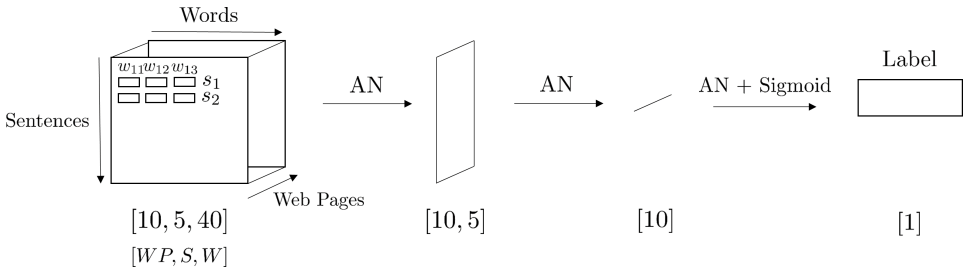

Figure 5: DomainAN3, in which domain level encodings are used to classify.

All the domain encodings can be directly used to classify when fed into a sigmoid or softmax layer. An example of using domain level encodings is shown in Fig. 5. To classify with Web page encodings or sentence encodings, we have to combine them in some way. We could simply concatenate the encodings to form one flat encoding. However, this creates long representations. To make this denser, we propose adding a CNN after the attention layers. For CNNs, the convention is to name a convolution after the number of directions that is moved in, i.e., if we move in only one direction this is called a 1-D convolution. This means that when moving from Web page encodings (a 2-D way to represent a domain) to a domain representation, we require 1-D convolutions as we generally only move over the Web page dimension and not over the encoding dimension.

Sentence encodings can be a 2-D way to represent a domain if we ignore the Web page layer or a 3-D way if we take the layer into account. In the former case, we require 1-D convolutions while in the second case 2-D convolutions are used (Fig. 6). The benefit of putting sentence encodings or Web page encodings into a CNN is that we omit the temporal aspect of RNNs, as this temporal aspect makes sense for words in sentences, that have logical ordering, but does not necessarily make sense for sentences in Web pages or Web pages in domains. To exemplify these structures, consider

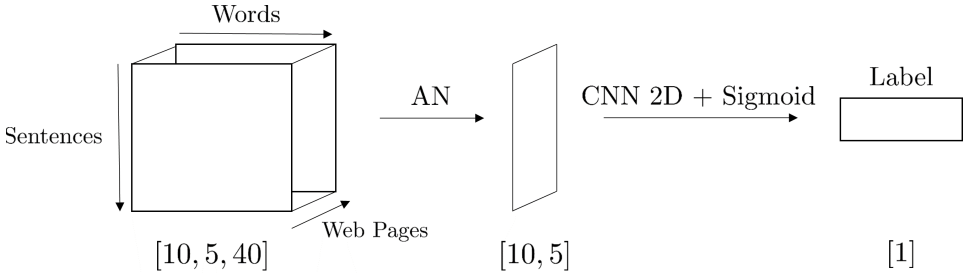

Figure 6: SentANConv2D, in which sentence level encodings are fed into a 2D CNN.

Table 3, which lists all different options and input dimensions for a particular example, with a domain consisting of 2,000 words and 10 Web pages, each Web page having 200 words and 5 sentences, and each sentence having 40 words. The word embeddings are also present, but not shown in the pictures or input dimensions. They can be considered a fourth or third input dimension, depending on the number of input dimensions (see Table 3). The abbreviations represent respectively the highest level attention encoding type (sentence level, Web page level, or domain level), the number of attention layers (1, 2, or 3) and the kind of convolutions (1-D or 2-D). In the case of

domain-level encodings with 2 attention layers, the abbreviation also shows which layer (sentence or Web page) is encoded for the first attention layer, e.g., DomainAN2Sent first encodes words to sentence encodings and then encodes the sentence encodings to one domain encoding.

Table 3: Proposed neural network architectures for text structures with three hierarchical layers.

Abbreviation	Word*	Sentence*	Page*	Input Dimensions
SentANConv1D	✓			[50, 40]
SentANConv2D	✓			[10, 5, 40]
PageANConv1D	✓			[10, 200]
PageAN2Conv1D	✓	✓		[10, 5, 40]
DomainAN2Sent	✓	✓		[50, 40]
DomainAN2Page	✓		✓	[10, 200]
DomainAN3	✓	✓	✓	[10, 5, 40]

*: Attention layer.

4.3. Considered Methods

In this paper, the aforementioned methods are applied to classify companies based on the content on their websites. All the models that we used are listed in Table 4. The Word2Vec embeddings are obtained from [33]. They are 160-dimensional and were trained on Dutch Wikipedia pages. Different combinations of word embeddings and neural networks are tried and compared to two benchmark models: Logistic regression on TF-IDF representations and Word2Vec embeddings, and SVMs on TF-IDF representations and Word2Vec embeddings. The neural networks are implemented by means of the Python library *Keras* [34] that uses the *TensorFlow* [35] backend for computation. In addition to the proposed architectures, we also experiment with 1-D convolutional networks, such as those described by [36]. CNN2 and CNN3 refer to neural networks with respectively two and three convolutional networks in a sequence. Each of these networks includes a convolutional layer and a max pooling layer.

In the end, the best model is used to identify sustainability in the Netherlands. The model is run over a list of about 6,000 companies in the 'industry' branch to obtain a list of sustainable companies per region. This can be used to colour regions according to the number of sustainable companies they have, scaled by their number of inhabitants. Preferably one would scrape the entire list of companies in the Netherlands but the

Table 4: Benchmarks and (proposed) neural network models with combinations of word representations (*italicized*) and classification methods (regular font) used to classify companies as either sustainable or non-sustainable.

Benchmark	Neural network
TF-IDF BoW Logit	CNN1 (*Word2Vec*)
TF-IDF Unigrams + Bigrams Logit	CNN2 (*Word2Vec*)
TF-IDF BoW + Word2Vec Logit	CNN3 (*Word2Vec*)
TF-IDF Unigrams + Bigrams + Word2Vec Logit	SentANConv1D (*Word2Vec*)
	SentANConv2D (*Word2Vec*)
TF-IDF BoW SVM	PageANConv1D (*Word2Vec*)
TF-IDF Unigrams + Bigrams SVM	PageAN2Conv1D (*Word2Vec*)
TF-IDF BoW + Word2Vec SVM	DomainAN2Sent (*Word2Vec*)
TF-IDF Unigrams + Bigrams + Word2Vec SVM	DomainAN2Page (*Word2Vec*)
	DomainAN3 (*Word2Vec*)

computation time associated with this is too long to be practical. The branch 'industry' was chosen as it includes all forms of production and therefore contributes greatly to the amount of pollution in the Netherlands.

4.4. Evaluation

To evaluate the classification, metrics of accuracy, precision, recall, and F_1 are used. Note that accuracy on its own is usually not enough. In the case that the classes are imbalanced, a model can have fairly high accuracy by always predicting the most common class, but not provide any additional information. For example, when trying to predict whether companies are sustainable, always predicting not sustainable will not provide any additional valuable information.

Precision is the amount of true positives divided by the total amount of predicted positives. It measures how many of the selected positives are actually positive or, in our case, how many of the companies that are predicted to be sustainable are actually sustainable. Recall is the amount of true positives divided by the true positives and false negatives. It measures how many of the actual positives were captured, or, in our case, what portion of the companies that are sustainable the model was able to find. To balance these two objectives the F_1 measure is often used, calculated as the harmonic mean of recall and precision. Note that precision and recall are defined for just one class. To get to a final evaluation measure, several approaches can be taken. In our case, we weight by the proportion of actual cases in the test

Table 5: Hyperparameter spaces and optimal values for different CNNs with Word2Vec embeddings.

Parameters	Options	CNN1	CNN2	CNN3
# Filters 1	{32, 64, 128}	64	32	32
Kernel size 1	{3, 5, 7}	5	3	7
Pooling size 1	{3, 5, 7}	5	7	5
Dropout 1	U(0.0, 0.6)	0.599	0.171	0.410
# Filters 2	{32, 64, 128}		64	32
Kernel size 2	{3, 5, 7}		7	3
Pooling size 2	{3, 5, 7}		7	5
Dropout 2	U(0.1, 0.6)		0.540	0.144
# Filters 3	{32, 64, 128}			32
Kernel size 3	{3, 5, 7}			5
Pooling size 3	{3, 5, 7}			3
Dropout 3	U(0.1, 0.6)			0.041
Learning rate	LogU{0.001, 0.0001}	0.00087	0.00056	0.00097

set.

5. Results

The results of the research are given in the sections below. Section 5.1 gives an overview of the process used to find the optimal hyperparameters for the neural network models, by means of the Tree-structured Parzen Estimator (TPE) such as given by [37]. Section 5.2 contains the F_1 measures corresponding to the optimal models. Here the optimal models are compared

Table 6: Hyperparameter spaces and optimal values for different attention models with Word2Vec.

Parameters	Options	SentANConv1D	SentANConv2D	PageANConv1D	PageAN2Conv1D	DomainAN2Sent	DomainAN2Page	DomainAN3
# Filters 1	{32, 64, 128}	64	32	128	64	-	-	-
Kernel size 1	{3, 5, 7}	3	3	5	7	-	-	-
Pooling size 1	{3, 5, 7}	5	7	3	3	-	-	-
Dropout 1	U(0.0, 0.6)	0.242	0.285	0.199	0.140	-	-	-
# Filters 2	{32, 64, 128}					-	-	-
Kernel size 2	{3, 5, 7}					-	-	-
Pooling size 2	{3, 5, 7}					-	-	-
Dropout 2	U(0.1, 0.6)					-	-	-
GRU Units 1	{32, 64, 128}	128	32	32	64	32	64	128
GRU Dropout 1	U(0.0, 0.6)	0.462	0.111	0.380	0.409	0.215	0.005	0.578
GRU Units 2	{32, 64, 128}				32	128	32	128
GRU Dropout 2	U(0.0, 0.6)				0.321	0.016	0.543	0.171
GRU Units 3	{32, 64, 128}							64
GRU Dropout 3	U(0.0, 0.6)							0.586
Learning rate	LogU{0.001, 0.0001}	0.00015	0.00035	0.00036	0.00024	0.00070	0.00014	0.00020

Parameters not included in the search are denoted with a hyphen (−).

and assessed. In the last section, Section 5.3, the optimal model is applied to a particular list of URLs to get insight into the distribution of sustainable companies in the 'industry' branch. We look at the influence of activity and location on the probability of being sustainable.

5.1. Optimal Model Hyperparameters

All of the neural network models require one to tune hyperparameters. In this paper, this is done by means of the Tree-structured Parzen Estimator (TPE) by [37]. The data is split into 64% train set, 16% validation set, and 20% test set. For every model, we run 30 trials of the algorithm, after which the best model is selected. The first 10 trials use random hyperparameters from a configuration space. The configuration spaces are shown in Tables 5 and 6. After these 10 random trials, the TPE chooses the set of hyperparameters to train with. TPE does not go beyond the configuration space boundaries.

Every CNN layer always has (i) a number of filters, (ii) a kernel size, (iii) a pooling size and a regular dropout probability. Dropout is applied after pooling. Every RNN layer has (i) a number of GRU units and (ii) a recurrent dropout probability. Note that GRU always refers to Bidirectional GRU in our case, which means the actual number of units is double the number in Table 6. For the convolutional layers, the activation function is always ReLU and the type of pooling is Max Pooling. A maximum of three GRU layers and two CNN layers is tried for the attention models while a maximum of three CNN layers is tried for the CNN models.

The configuration space is based on manual assessment of several options and on computational constraints, e.g., we do not try GRUs with 256 units as they severely slow down the training. Note that in case a model has multiple of the same layer, the same parameters are tried in every layer, e.g., in CNN, kernel sizes of 3, 5, and 7 are tried in both layer 1 and layer 2. The architectures are less complex compared to commonly used networks such as Inception v3 by [38], as the amount of data we have is limited.

Within every trial, the model is chosen by means of early stopping based on the validation loss. The patience is set at 0, meaning that whenever the validation loss goes up during an epoch, the training is stopped and that model is selected. The maximum number of epochs is set at 60, so if the validation loss does not go up within 60 epochs, the 60th model is selected. In practice, this number is never reached.

5.2. Performance Comparison

Table 7 lists the weighted F_1 measure for all the traditional methods on the test set, made up of 449 sustainable companies and 501 non-sustainable ones.

The weights are given by the proportion of cases of the class in the test set. The majority refers to always predicting the majority class, which is the non-sustainable class in our case. As for SVMs, both Linear SVMs and RBF SVMs were tried. For both Logit and SVM only the top 1,000 features were used based on the chi-squared statistic between the feature and the class. Testing showed this gives better performance than using all features. Chi-squared is used as [39] show it to be better than other approaches such as using mutual information or term strength.

In general, precision is slightly higher than the other metrics for all the methods, but, as the classes were fairly balanced, accuracy, precision, and recall do not differ much for all the methods, apart from choosing the majority. Among the traditional methods, we can spot three main results: (i) using both unigrams and bigrams rather than just unigrams increases the F_1 score; (ii) using a linear SVM rather than Logit or an RBF SVM

Table 7: Accuracy, (weighted) precision, (weighted) recall, and (weighted) F_1 in percentages, on the test set, for traditional methods.

Model*	Acc.	Prec.	Rec.	F_1
Majority (non-sustainable)	52.74	42.21	27.81	36.42
BoW Logit	85.26	85.39	85.26	85.22
Uni- + Bigrams Logit	85.58	85.72	85.58	85.53
BoW + Word2Vec Logit	87.16	87.32	87.16	87.11
Uni- + Bigrams + Word2Vec Logit	86.42	86.59	86.42	86.37
BoW Linear SVM	86.84	87.03	86.84	86.79
Uni- + Bigrams Linear SVM	86.95	87.13	86.95	86.90
BoW + Word2Vec Linear SVM	87.16	87.21	87.16	87.13
Uni- + Bigrams + Word2Vec Linear SVM	86.95	86.97	86.95	86.93
BoW RBF SVM	86.11	86.38	86.11	86.04
Uni- + Bigrams RBF SVM	86.53	86.86	86.53	86.45
BoW + Word2Vec RBF SVM	**87.37**	**87.66**	**87.37**	**87.30**
Uni- + Bigrams + Word2Vec RBF SVM	86.95	87.26	86.95	86.88

*: For all models, prefix 'TF-IDF' is omitted for notational convenience.

increases the F_1 score in most of the cases; (iii) methods using Word2Vec embedding perform significantly better when compared to the rest.

Table 8 shows the 10 features with the highest coefficients. They are very similar for Logit and the SVM, both containing features such as 'sustainable' and 'biological'. The SVM puts slightly more emphasis on features related to 'environment', while Logit includes 'garbage'.

Table 8: Top 10 features for both Logit and SVM (Unigrams and Bigrams).

Logit	SVM
sustainable[1a]	environment
environment	sustainable[1a]
biological	biological
sustainable[1b]	sustainable[1b]
nature	nature
natural	natural
biological	sustainability
sustainability	biological
energy	energy
garbage	environmentally friendly

[1]: From Dutch (a) 'duurzame' and (b) 'duurzaam'.

Once again, Tables 5 and 6 list the optimal hyperparameters for all discussed neural networks. The batch size was always kept at 8 instances. As [40] shows, larger batch sizes can lead to better performance, but the difference is small. In any case, batch sizes larger than 8 were not tried due to the limited VRAM available. Table 9 lists the accuracy and weighted precision, recall, and F_1 for the neural network models on the test data set.

Most of the neural network models outperform the traditional methods by a considerable margin when looking at weighted F_1 scores. Among the CNN models, CNN2 performs best, having a weighted F_1 score of 89.96%. The only attention model that outperforms CNN2 is PageANConv1D, scoring 90.20%. This shows that attention models can outperform CNNs, even on a small data set, but with a small margin. We hypothesize the difference is small due to the fact that the more complex models have both more hyperparameters to optimize and more parameters within the model. Both of these aspects would benefit from a larger data set. CNNs do have relatively fast computation time with only 20 seconds per epoch while PageANConv1D

takes 400 seconds per epoch. This can be an argument to choose a CNN when working in a small scale environment.

Among the attention models, three groups can be distinguished with different levels of performance. One is made up of DomainAN2Page and DomainAN3 having weighted F_1 scores around 87%, one is made up of SentANConv1D, SentANConv2D and DomainAN2Sent, having F_1 scores around 89-89.5%, and the last, with F_1 scores of 89.8-90.20%, is made up of PageANConv1D and PageAN2Conv1D. DomainAN2Page and DomainAN3 both encode page encodings to one domain encoding by means of an AN layer. As noted before, RNNs should be used when a temporal component is present in the data. It can be doubted whether such a component is present between Web pages as they do not have some natural ordering, unlike words and sentences. The second group of models does not use Web page encodings, while the last group does. Encoding up to this level by means of AN layers seems to give the best performance. Note that the domain encodings are not used here.

Table 10 gives us the accuracy, weighted precision, recall, and F_1 for the neural network models on the validation data set. Comparing them to the results from Table 9, the results are slightly better for the majority of the models. This gives us the indication that there is slight overfitting. This is rather common when training models with many parameters like neural networks.

Table 9: Accuracy, (weighted) precision, (weighted) recall, and (weighted) F_1 in percentages, number of epochs, and time per epoch (in seconds) for the test set, for neural network models with Word2Vec as word embeddings.

Model	Acc.	Prec.	Rec.	F_1	Epochs	Time
CNN1	89.47	89.49	89.47	89.48	2	18 - 21
CNN2	90.00	90.23	90.00	89.96	2	16 - 18
CNN3	89.58	89.58	89.58	89.58	2	17 - 20
SentANConv1D	88.95	88.96	88.95	88.94	3	199 - 201
SentANConv2D	89.58	90.10	89.58	89.51	2	83 - 87
PageANConv1D	**90.21**	**90.24**	**90.21**	**90.20**	2	419 - 462
PageAN2Conv1D	89.79	89.84	89.79	89.77	1	151 - 165
DomainAN2Sent	89.37	89.57	89.37	89.33	3	369 - 377
DomainAN2Page	87.37	88.06	87.37	87.36	2	437 - 449
DomainAN3	87.79	87.80	87.79	87.78	1	259 - 262

We can also look at the evaluation measures per class in Table 11. We see that PageANConv1D still wins in terms of F_1 scores for both the sustainable class and the non-sustainable class, with scores of respectively 89.4% and 90.9%, but is edged out by other methods when looking at precision and recall separately. SentANConv2D has higher precision for the sustainable class, of 94.6% and higher recall for the non-sustainable class, of 95.8%, while DomainAN2Page has higher recall for the sustainable class, of 93.3% and higher precision for the non-sustainable class. Policy assessors are mostly interested in following a subset of companies that are sustainable and seeing where they are located. For such purposes, we need not necessarily identify all the sustainable companies and precision may be more valued. Under such circumstances, SentANConv2D can be a valuable alternative to PageANConv1D. If we do want to capture all companies and we have sufficient manpower to check the companies that are labelled sustainable, recall is more valued and DomainAN2Page is a good alternative.

Finally, for the majority of the models, the results for the validation set are again slightly better than for the test set.

5.3. Applying the Model

After obtaining the model, it can be used to classify any set of companies. In our case, we classify all companies in the branch 'industry' using PageANConv1D. Industry was chosen as it contains the most polluting activities. This includes activities such as 'production of clothes', 'production

Table 10: Accuracy, (weighted) precision, (weighted) recall, and (weighted) F_1 in percentages for the validation set, for neural network models with Word2Vec as word embeddings.

Model	Acc.	Prec.	Rec.	F_1
CNN1	91.45	91.46	91.45	91.45
CNN2	90.26	90.92	90.26	90.23
CNN3	90.66	90.74	90.66	90.65
SentANConv1D	91.18	91.43	91.18	91.17
SentANConv2D	89.34	90.13	89.34	89.30
PageANConv1D	91.18	91.25	91.18	91.18
PageAN2Conv1D	**91.71**	**91.90**	**91.71**	**91.70**
DomainAN2Sent	90.13	90.61	90.13	90.11
DomainAN2Page	90.39	90.59	90.39	90.38
DomainAN3	87.50	87.67	87.50	87.49

of furniture', etc. The complete set of activities is shown in Table 12. After scraping the Web pages of companies for which we could obtain a zip code, we were left with 5,919 companies. 1,199 of these companies are deemed to be sustainable by our model. The distribution of companies over different activities is shown in Fig. 7.

It has to be noted that both activity 3 and activity 10 only consisted of one company. If we exclude these we see that the activities with the most sustainable companies are respectively (i) production of chemical products,

Table 11: Precision, recall, and F_1 in percentages for classes separately, for the validation and test sets, for neural network models with Word2Vec as word embeddings.

	Prec.	Rec.	F_1	Prec.	Rec.	F_1
	Sustainable			Non-sustainable		
Model	Validation					
CNN1	92.29	90.61	91.44	90.63	92.31	91.46
CNN2	**96.12**	84.07	89.69	85.65	**96.55**	90.77
CNN3	92.62	88.51	90.52	88.83	92.84	90.79
SentANConv1D	94.63	87.47	90.91	88.18	94.96	91.44
SentANConv2D	95.76	82.51	88.64	84.42	96.29	89.96
PageANConv1D	92.93	89.30	91.08	89.54	93.10	91.29
PageAN2Conv1D	94.69	88.51	**91.50**	89.05	94.96	**91.91**
DomainAN2Sent	95.03	84.86	89.66	86.12	95.49	90.57
DomainAN2Page	87.80	**93.99**	90.79	**93.43**	86.74	89.96
DomainAN3	90.22	84.33	87.18	85.07	90.72	87.80
	Test					
CNN1	88.18	89.76	88.96	90.67	89.22	89.94
CNN2	93.17	85.08	88.94	87.59	94.41	90.87
CNN3	89.24	88.64	88.94	89.88	90.42	90.15
SentANConv1D	89.45	86.86	88.14	88.52	90.82	89.66
SentANConv2D	**94.64**	82.63	88.23	86.02	**95.81**	90.65
PageANConv1D	91.20	87.75	**89.44**	89.38	92.42	**90.87**
PageAN2Conv1D	91.12	86.86	88.94	88.70	92.42	90.52
DomainAN2Sent	92.23	84.63	88.27	87.17	93.16	90.28
DomainAN2Page	82.32	**93.32**	87.47	**93.20**	82.04	87.26
DomainAN3	88.10	85.75	86.91	87.52	89.62	88.56

(ii) production of leather, leather products and shoes, and (iii) production of food. The activities with the least number of sustainable companies are (i) production of computers and electronic and optical equipment, (ii) printing, and (iii) production of cars, trailers, and semitrailers.

Table 12: Number of companies and sustainability ratios per activity.

ID	Activity	# Comp.	% Sust.
1	Prod. of food	360	36.1
2	Prod. of drinks	101	29.7
3	Prod. of tobacco products	1	100.0
4	Prod. of textile	289	14.9
5	Prod. of clothes	222	15.3
6	Prod. of leather, leather products and shoes	76	22.4
7	Primary wood processing and production of products made out of wood (no furniture)	270	31.1
8	Prod. of paper, cardboard and paper- and cardboard products	36	27.8
9	Printing	508	15.0
10	Prod. of coke oven products and oil processing	1	0.0
11	Prod. of chemical products	58	48.3
12	Prod. of pharmaceutical materials and products	12	33.3
13	Prod. of rubber and plastic products	138	26.1
14	Prod. of other non metal containing mineral products	298	18.1
15	Prod. of metals in primary form	37	16.2
16	Prod. of metal products (no machines and equipment)	704	9.5
17	Prod. of computers and electronic and optical equipment	126	11.1
18	Prod. of electronic equipment	110	32.7
19	Prod. of other machines and equipment	241	20.3
20	Prod. of cars, trailers and semitrailers	89	11.2
21	Prod. of other ways of transport	139	8.6
22	Prod. of furniture	936	23.2
23	Prod. of other goods	658	16.4
24	Repair and installation of machines and equipment	600	12.7

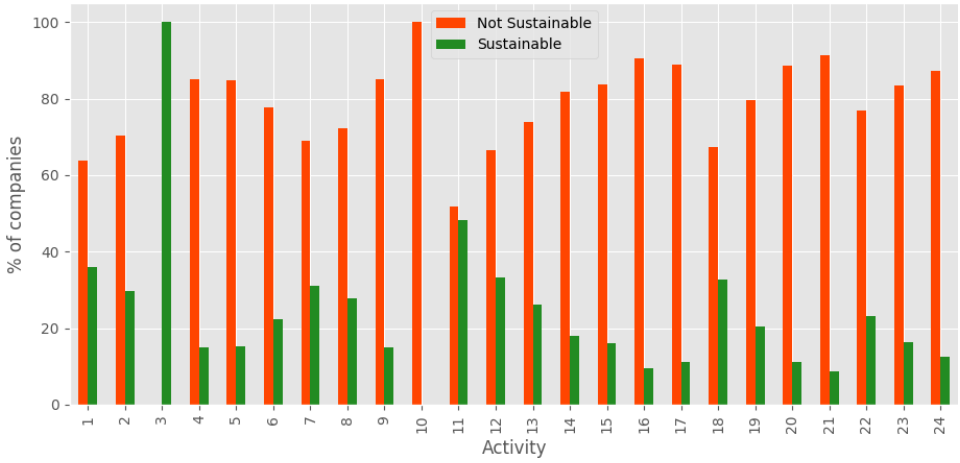

Figure 7: The percentage of sustainable companies per activity, in the branch 'industry', found using PageANConv1D.

Using the zip codes, we can also plot the distribution of sustainable companies over the Netherlands. This is shown in Fig. 8, both per capita and per company. The two maps give different impressions. Per capita, Utrecht, Friesland, Flevoland, Zuid-Holland, and Overijssel have a large number of sustainable companies, partially due to the fact they have a large number of companies in the industry branch. If we look at the actual share of sustainable companies compared to the total number of industrial companies

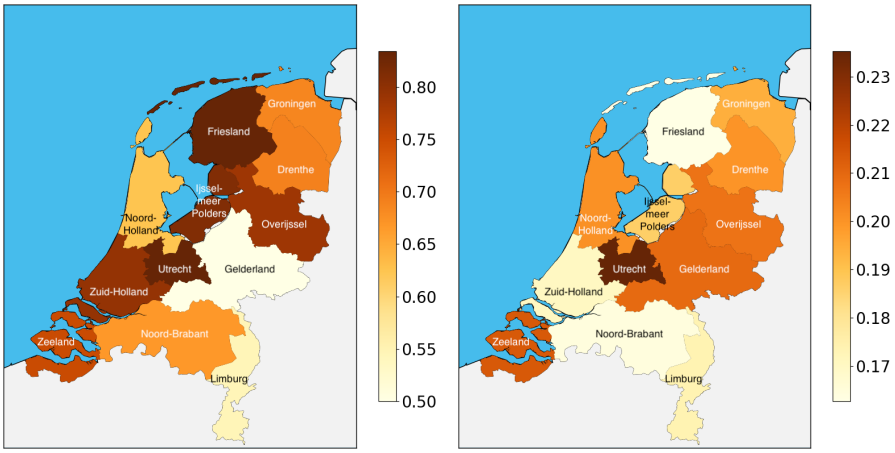

Figure 8: The number of sustainable companies per capita times 10,000 (left) and per company (right), in the branch industry, respectively.

in an area, the province Utrecht clearly stands out. Likewise, Zeeland also stands out in both the left and the right pictures. Work by ING [41] agrees with the notion that Utrecht is the cleanest economy in the Netherlands, but according to them, Zeeland is the least sustainable economy. The difference in analysis could be due to the fact that our analysis only focuses on the industry branch.

6. Conclusion

We provided a comprehensive overview of methods that are commonly used to classify texts. In addition, a new neural network was introduced that uses the hierarchical structure of Web pages, similar to [15], but adds an additional layer. We have also experimented with replacing some attention layers with convolutional layers. The methods were used to classify whether companies are sustainable, based on the information on their websites. The data set was constructed by using a base set of companies that were deemed to be sustainable and randomly taking companies from a general list of companies. Afterwards, the data was annotated.

Our novel architectures led to slightly better results in the case of one model, PageANConv1D, which encodes pages by means of attention and uses a convolutional layer to combine the page encodings. The difference between a simple CNN with two layers and PageANConv1D was, however, small.

After the best model was found, it was applied to a set of companies belonging to one particular branch: industry. Within this branch, sustainable companies were mostly found to be in (i) the production of chemical products, (ii) production of leather, leather products and shoes, and (iii) the production of food. As the reputation of chemical products is for a large part dependent on their environmental awareness, it should be no surprise they put effort into being as sustainable as possible. Production of food naturally lends itself well as an activity to be made sustainable, as it can be produced locally, lessening pollution caused by transportation. The least sustainable activities were found to be (i) production of computers, (ii) printing, and (iii) production of cars, trailers, and semitrailers. Activities (i) and (iii) are harder to be made sustainable as they require relatively more research & development before offering economically feasible sustainable solutions.

In terms of geographic spread, it was found that Utrecht has a considerable number of sustainable companies, both per capita and in relation to the total number of companies. The same is true for Zeeland. ING agrees with the remark that Utrecht is one of the most sustainable Dutch regions [41].

Our solution allows for obtaining a comprehensive overview of sustainability at company or region level, enabling governments to apply targeted policies to boost sustainability at different granularity levels. This approach is fully automatic and there is no need for filling out questionnaires and conducting interviews, improving the cost efficiency of governance. While we have applied our proposed model for sustainability, its genericity makes it amenable to other classifications tasks requiring analysis of website contents.

One limitation of our study is that we assume the companies to be truthful with the information they publish on their websites. This means that the levels of wrong or biased self-reporting are regarded to be low enough to neglect them. However, in some cases, this assumption might not be correct (we assume that the number of such cases is low as it is illegal for a company to provide statements on their websites which knowingly misrepresent reality). In the future, it would be interesting to detect these noisy observations and remove them from our data set. Further research should focus on applying similar models to other branches of companies and examining whether this is still the case then. Our solution shows how one can identify whether a company or region is sustainable without the need for questionnaires, in an automatic manner using Web data. Additionally, we would also like to test the proposed models on larger data sets to gain further evidence on the efficacy of the proposed models. Apart from that, experiments with wider networks, i.e., more units per RNN layer, can be worthwhile, as some of the networks had 128 units (the maximum we have experimented with) as their optimum.

References

[1] N. P. Melville, "Information systems innovation for environmental sustainability," *MIS Quarterly*, vol. 34, no. 1, pp. 1–21, 2010.

[2] G. T. M. Hult, "Market-focused sustainability: market orientation plus!," *Journal of the Academy of Marketing Science*, vol. 39, no. 1, pp. 1–6, 2011.

[3] S. Elliot, "Transdisciplinary perspectives on environmental sustainability: a resource base and framework for it-enabled business transformation," *MIS Quarterly*, vol. 35, no. 1, pp. 197–236, 2011.

[4] S. Seidel, J. C. Recker, and J. Vom Brocke, "Sensemaking and sustainable practicing: functional affordances of information systems in green transformations," *MIS Quarterly*, vol. 37, no. 4, pp. 1275–1299, 2013.

[5] R. G. Eccles, I. Ioannou, and G. Serafeim, "The impact of corporate sustainability on organizational processes and performance," *Management Science*, vol. 60, no. 11, pp. 2835–2857, 2014.

[6] KPMG, "The kpmg survey of corporate responsibility reporting 2017," tech. rep., KPMG, 2017.

[7] I. Bose and R. Pal, "Do green supply chain management initiatives impact stock prices of firms?," *Decision Support Systems*, vol. 52, no. 3, pp. 624–634, 2012.

[8] R. Durand, L. Paugam, and H. Stolowy, "Do investors actually value sustainability indices? replication, development, and new evidence on csr visibility," *Strategic Management Journal*, vol. 40, no. 9, pp. 1471–1490, 2019.

[9] UN, "Transforming our world: the 2030 agenda for sustainable development," tech. rep., United Nations, 2015.

[10] R. Gopal, J. R. Marsden, and J. Vanthienen, "Information mining—reflections on recent advancements and the road ahead in data, text, and media mining," *Decision Support Systems*, vol. 51, pp. 727–731, 2011.

[11] A. Garcia-Crespo, R. Colomo-Palacios, J. M. Gomez-Berbis, and B. Ruiz-Mezcua, "Semo: a framework for customer social networks analysis based on semantics," *Journal of Information Technology*, vol. 25, no. 2, pp. 178–188, 2010.

[12] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS 2013)*, pp. 3111–3119, Curran Associates, 2013.

[13] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[14] Y. Wen, W. Zhang, R. Luo, and J. Wang, "Learning text representation using recurrent convolutional neural network with highway layers," *arXiv preprint arXiv:1606.06905*, 2016.

[15] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NACCL HLT 2016)*, pp. 1480–1489, ACL, 2016.

[16] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1 (NIPS 2015)*, pp. 649–657, Curran Associates, 2015.

[17] Y. LeCun, Y. Bengio, *et al.*, "Convolutional networks for images, speech, and time series," *The Handbook of Brain Theory and Neural Networks*, vol. 3361, no. 10, pp. 276–278, 1995.

[18] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[19] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pp. 1412–1421, ACL, Sept. 2015.

[20] G. Brauwers and F. Frasincar, "A general survey on attention mechanisms in deep learning," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[21] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya, "ABCDM: An attention-based bidirectional CNN-RNN deep model for sentiment analysis," *Future Generation Computer Systems*, vol. 115, pp. 279–294, 2021.

[22] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*, pp. 2267–2273, AAAI Press, 2015.

[23] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[24] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2 (NIPS 2014)*, pp. 2204–2212, 2014.

[25] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pp. 1724–1734, ACL, Oct. 2014.

[26] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.

[27] K. L. Gwet, "Computing inter-rater reliability and its variance in the presence of high agreement," *British Journal of Mathematical and Statistical Psychology*, vol. 61, no. 1, pp. 29–48, 2008.

[28] J. Pomikálek, *Removing boilerplate and duplicate content from web corpora*. PhD thesis, Masarykova Univerzita, Fakulta Informatiky, 2011.

[29] G. Diaz, "Stopwords Dutch (NL)." https://github.com/stopwords-iso/stopwords-nl, 2016.

[30] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[31] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proceedings of the 30th International Conference on International Conference on Machine Learning (ICML 2013) - Volume 28*, pp. III–1310–III–1318, JMLR.org, 2013.

[32] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pp. 1422–1432, ACL, 2015.

[33] S. Tulkens, C. Emmery, and W. Daelemans, "Evaluating unsupervised dutch word embeddings as a linguistic resource," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pp. 4130–4136, ELRA, 2016.

[34] F. Chollet *et al.*, "Keras." https://keras.io, 2020.

[35] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation (OSDI 2016)*, pp. 265–283, USENIX Association, 2016.

[36] Y. Goldberg, "Neural network methods for natural language processing," *Synthesis Lectures on Human Language Technologies*, vol. 10, no. 1, pp. 1–309, 2017.

[37] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proceedings of the 24th International Conference on Neural Information Processing Systems (NIPS 2011)*, pp. 2546–2554, 2011.

[38] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pp. 2818–2826, IEEE, 2016.

[39] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the Fourteenth International Conference on Machine Learning (ICML 97)*, pp. 412–420, Morgan Kaufmann Publishers Inc., 1997.

[40] P. Goyal, P. Dollár, R. B. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch SGD: training imagenet in 1 hour," *arXiv preprint arXiv:1706.02677*, 2017.

[41] ING, "Utrecht de schoonste economie zeeland de meest vervuilende," tech. rep., ING, 2018.