Athens Journal of Sciences- Volume 11, Issue 2, June 2024 – Pages 89-108

Identifying Sustainability Efforts in Company's Reports Using Text Mining and Machine Learning

By Evangelos Xevelonakis[*] & Tanbir Mann[±]

This study delves into the utilization of text mining to scrutinize social and environmental reports of companies, showcasing its effectiveness in evaluation. It explores various text mining techniques and practically applies decision tree, k-nearest neighbors, and naïve Bayes methods. The paper offers guidance on extracting pertinent terms related to four CSR dimensions: Environment, Employee, Social responsibility, and Human rights. Results demonstrate the successful differentiation of text based on these dimensions, leveraging a CSR-relevant dictionary by Pencel and Malascue. Employing document classification techniques, the study constructs four models using distinct text mining approaches for comparative analysis. Through this research, the valuable role of text mining in assessing social and environmental disclosures is underscored, providing insights into optimizing these techniques for evaluations and emphasizing their potential to enhance understanding and decision-making in corporate social responsibility assessments.

Keywords: *sustainability, text mining, machine learning, Corporate Social Responsibility - CSR, environmental reports*

Introduction

Text mining, also known as text data mining or knowledge detection from textual sources, generally refers to the process of extracting compulsive and significant patterns or information from unstructured text documents (Tan 1999). This handling of unstructured data distinguishes text mining as a discriminative technology from newer models whose application is based more on the use of generative AI.

With text being the most common form of storing information, it is safe to say that text mining has a high potential in commercial contexts. As most of the information within companies is stored in textual form, text mining is one of the most useful means of gathering company related information. Text mining, however, is also a complex task as it includes dealing with unstructured and fuzzy text data. Text mining is a multidisciplinary field involving information retrieval, text analysis, information extraction, clustering, categorization, visualization, database technology, machine learning, and data mining (Hearst 1997).

One interesting field where text mining technology could produce important insights is social responsibility, which plays a major role in today's society. Social responsibility is not only restricted to individuals, but also to companies and most

[*]Head, Center for Data Science & Technology, HWZ Zurich University of Applied Sciences, Switzerland.
[±]Research Assistant, HWZ Zurich University of Applied Sciences, Switzerland.

https://doi.org/10.30958/ajs.11-2-2 doi=10.30958/ajs.11-2-2

companies are expected to undertake social aspects in their business engagements. This field is known as Corporate Social Responsibility (CSR). CSR is as a supervised notion whereby companies apply management notions in social and environmental concerns in their business operations and interconnections with their stakeholders. It is immensely important to have full disclosure to stakeholders, especially in issues of corporate social responsibility. Therefore, a large number of companies are getting involved in preparing sustainability reports and generally disclosing more of their impact on sustainability, specifically providing information on economic, social and environmental dimensions, which are the major measures of corporate sustainability quality – the so-called 'triple bottom line'. These reports are either voluntary or, depending on the country examined, could be a result of legislation or part of a code of practice. Such CSR reports are often based on frameworks and standards like UN Global Compact Principles, OECD Guidelines for Multinational Enterprises, GRI guidelines, ISO 26000, AA1000 and SA88000 (Aureli 2016).

Objective

The objective of this project is to use text mining methods and techniques to perform screening and selection of qualitative information from text. This project is focused on the process of document classification that helps in automating the assessment of company related data to screen for text that is specifically related to the CSR management concept with respect to its four different dimensions – Environmental responsibility, Employee responsibility, Social responsibility and Human rights.

Related Work

In recent years, an extensive amount of research has been done in this field. Joao and Paulo used Text Mining methodology (a Bayesian contextual analysis algorithm known as Correlated Topic Model, CTM) to conduct a comprehensive analysis of 246 articles published in 40 different journals between 1988 and 2013 on the subject of cause-related marketing (Guerreiro et al. 2016). Lin and Hsu (2018) conducted a study to examine the impact of corporate social responsibility (CSR) news reports on corporate operating performance forecasting using a large database of publicly listed electronics firms in Taiwan. Applying text mining techniques and latent topic modelling, they construct and measure the intensity of the CSR-corpus index (ICSRI), which can compress tremendous amounts of CSR textual information content into synthesized meaningful dimensions (Lin and Hsu 2018). In another study by Te Liew et al. (2014), text mining was used to identify sustainability trends and practices in the process industries. Four main sectors of the industry were studied: oil/petrochemicals, bulk/specialty chemicals, pharmaceuticals, and consumer products. The study reveals that the top sustainability focuses of the four sectors are very similar: health and safety, human

rights, reducing GHG, conserving energy/energy efficiency, and community investment (Te Liew et al. 2014).

In an empirical study, results show that the financial report sentiment based on the PESTEL model, Porter's Five Forces model, and Value Chain (Primary and Support Activities) significantly correlates with the CSR score. The study characterizes the interaction between the sentiment analysis of financial reports and CSR scores (Song et al. 2018). Pencle and Mălăescu (2016) developed a content analytic dictionary using computer-aided text analysis (CATA). The dictionary, validated in the context of U.S. IPOs between 2011 and 2013, revealed four dimensions of corporate social responsibility. Each of these dimensions is used to predict IPO (Initial Public Offering) size in terms of offering price and total shares offered, as well as underpricing on the first day of trade. Their research provides a new measure of CSR that can be generalized to other text documents issued by a corporation (Pencle and Mălăescu 2016).

Text Mining Algorithms

Text mining algorithms are specific data mining algorithms in the domain of natural language processing (NLP). The text can be any type of content – postings on social media, email, business word documents, web content, articles, news, blog posts, and other types of unstructured data. These algorithms help to provide some understanding of how text is processed. Some text mining techniques, such as document classification and information retrieval, are concentrated on ranking or locating specific types of documents in a large document base. The main assumption behind these algorithms is that syntactic similarity (similar words) implies semantic similarity (similar meaning). However, these techniques work very well depending on the syntactic information, just because the used documents are often on the same subject and therefore share many keywords or terms (Ghosh et al. 2012).

Algorithms for text analytics incorporate a variety of techniques such as text classification, categorization, and clustering. All of them aim to uncover hidden relationships, trends, and patterns which are a solid base for business decision-making.

Document Classification

Document Classification methods are the best-known methods of Text Mining. They use training data with known target variables to classify new documents into a set of categories (Miner et al. 2012). Statistical models but also rule-based models can be used. There are two approaches, the instance-based approach (lazy learning), where new data sets are classified directly using the training data, and the model-based approach (eager learning), where a model is calculated and applied to the new data (Cleve and Lämmel 2016). These methods are used, for example, to identify spam emails or to classify customers into risk groups according to their creditworthiness.

For classification, a term-by-document matrix is used as input. The terms represent the features that should be carefully chosen, and the documents should be well described. However, other properties can also be used as features - for example, the amount of text, number of key words or the URL. Especially documents from the internet have special features that can be used for classification. For example, the URL contains information about the protocol used, the domain and the directory depth. At the end, all text-based features must be transformed into numerical features (Miner et al. 2012).

K-Nearest Neighbor

The k-nearest neighbor method is known as an instance-based method, in which new data sets are classified with the help of the training data. This approach requires a term-by-document matrix, including the target variables. Then, depending on these variables, the method tries to find the "nearest neighbor", i.e. a record in the training data that is the closest to the new record to be classified. The category of the found training data record is finally adopted. To find the nearest neighbor, similarity measures like the Euclidean distance are used. The number of "neighbors" to be used for the selection of the category must be selected in advance. At k=1, the category of the next neighbor is taken over. If k>1, the category that occurs most frequently among the neighbors used is selected (Cleve and Lämmel 2016).

Figure 1. *Addition of New Data Point*

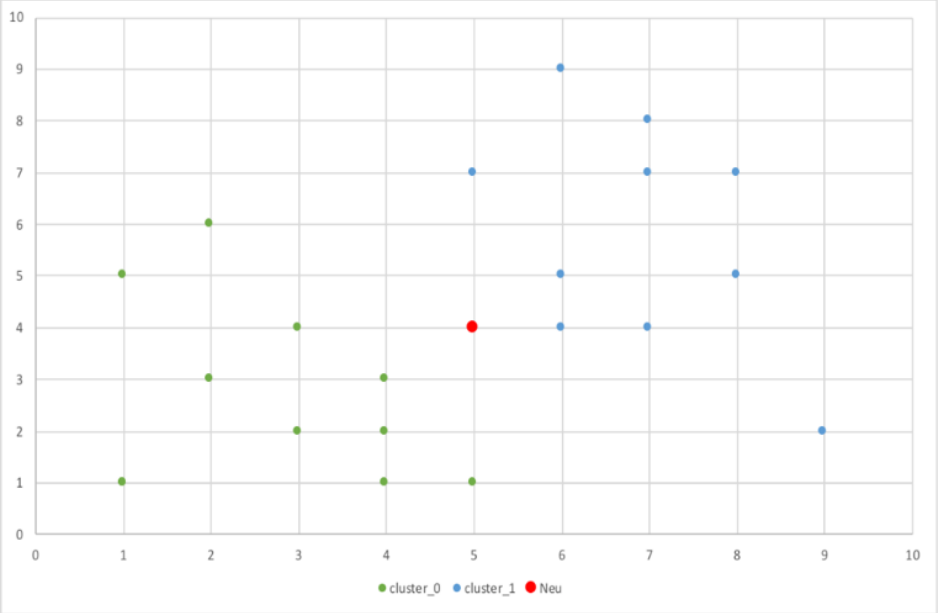


Figure 2. *Cluster Assignment*

k	x	y	Cluster
1	5	4	cluster_1
2	5	4	cluster_1
3	5	4	cluster_1
4	5	4	cluster_1
5	5	4	cluster_1
6	5	4	cluster_1
7	5	4	cluster_1
8	5	4	cluster_0
9	5	4	cluster_1
10	5	4	cluster_1
11	5	4	cluster_0
12	5	4	cluster_0

In the above example a new data point (red) is added with the coordinates (5:4) in Figure 1. The algorithm should now decide to which cluster the new point belongs - to "cluster_0" or to "cluster_1". If the algorithm is used with different values for k, the assignment to a cluster can change (see Figure 2).

Naive Bayes Classifier

The Naïve Bayes method is a probability-based method, with an instance-based approach. The goal is to predict the most probable class. The algorithm "is based on the Bayesian formula, which in calculations with conditional probabilities allows the interchanging of dependent events" (Cleve and Lämmel 2016). The algorithm assumes the independence of the variables. X is the target variable, so that $(X|Y)$ is calculated for both values and finally the value with the higher value is applied.

$$P(X|Y) = \frac{P(Y|X) * P(X)}{\mathrm{P(Y)}}$$

Decision Trees

A decision tree is an instrument to display knowledge or results of a condition in a tree-like structure. The topmost node is the root, an edge represents a rule and a node which has no further branches represents a leaf. From the root to a leaf, certain rules apply, which are also applied to the data to classify it according to the leaf.

A rule imposes certain conditions (for example, greater, less or equal) on the variables of the data set, which lead to the data set being separated in such a way that one or more variables are as complete as possible in a subset. The variables that best describe the target variable are highest in the decision tree (Cleve and Lämmel 2016). The selection of the variables can be done manually, randomly or by calculation.

Figure 3. *Decision Tree Example*

In the first step, a variable is selected, and a rule is established to separate the data sets. The goal is to maximize the information content at each step. If a split is no longer possible, the branch ends in a leaf. The class of this sheet is determined by the most frequently occurring class in the sheet. The information content can be measured, for example, by the Gini index, an unequal distribution coefficient. It measures the diversity in a set and varies between 0 and 1. The lower the index, the more even the distribution.

In Figure 3, the decision tree formation is shown with a document example (Sayad 2024).

Data Collection and Preparation

In the initial phase of the project, only a single document was analyzed to check the feasibility of the text mining approach. Later, the same approach was used on around 60 documents. The data science tool KNIME was used for the implementation.

The process that has been followed in order to successfully implement the objective of this project starts with data preprocessing, where all the collected documents are preprocessed first, resulting in a clean dataset. This preprocessed data is then tagged using a CSR dictionary from Pencle and Malaescu (2016). After document tagging a bag-of-words model is used to generate the word cloud. A word cloud helps to analyze the terms that are frequently reoccurring in a document. Later, based on this bag of words creator, a document vector was generated and the terms which were present in the bag of words creator were then analyzed using 3 different text mining algorithms as described in section 3.

In the following, a step-by-step description of the analysis and the workflows that are used in the project is presented.

Data Preprocessing

The collected documents, which are in a text format also known as unstructured data, are processed and converted into a format that can be used for analysis using text mining methods. The pdf documents are cleaned in a first step to ensure that only relevant data is loaded. Subsequently, the data set is tagged by a CSR

Dictionary. First, a CSR dictionary is procured, a KNIME Extension is programmed and finally a KNIME Workflow is developed to tag the CSR relevant terms in the texts. More detailed information about the CSR dictionary is provided in the next section.

In the data cleaning process, numbers, punctuation marks and stop words are removed and all letters are converted to lower case. As a last step, all words with less than 3 letters are removed. Further pre-processing steps, such as the creation of a bag-of-words, are performed later in the application of a CSR dictionary. Stemming and lemmatization are not used as they are not required when using a dictionary.

Figure 4. *Data Preprocessing Workflow*

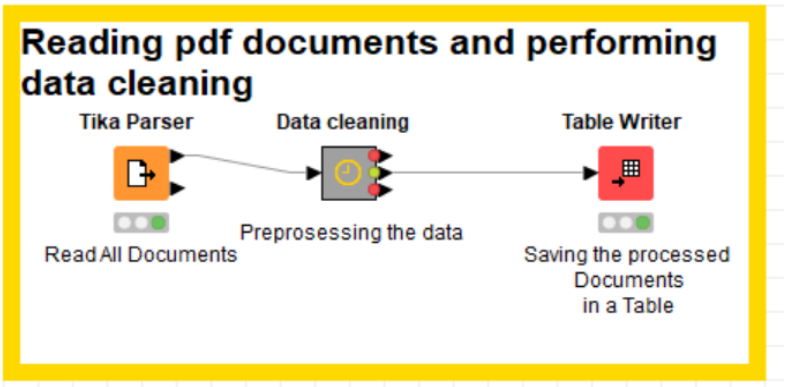
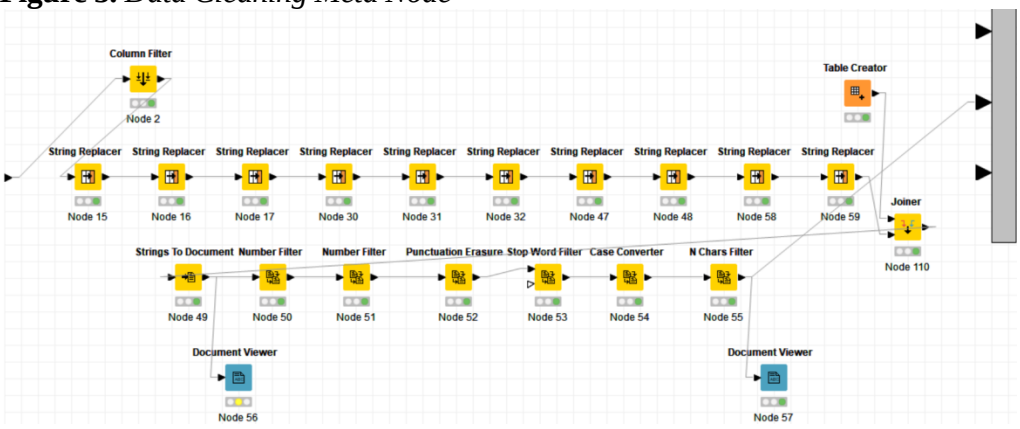


Figure 5. *Data Cleaning Meta Node*



As shown in Figure 4, a *Tika Parser* node is used to read the pdf data. In the data cleaning meta node (see Figure 5), a *String Replacer* node is used to clean the documents with patterns like: –, ®, •, etc., along with punctuations and stop words. A *Table Creator* node is utilized to categorize each document into one of four dimensions: Environment, Employee, Social, and Human Rights. All the documents are pre-classified with respect to these dimensions and hence are added in the documents pre-processing phase. This step is performed in the very beginning to get the category of each document. This step is essential as later *Document Vector* and *Document Extraction* nodes are used to create term metrics (further description is provided in the analysis section). A *Joiner* node is used to

join the two tables into one and adds a new column called 'class' in the document table. At the end of the preprocessing phase, the documents are stored in a new table which is then used to perform the text mining methods.

After preprocessing, the next steps focus on tagging the terms in the clean documents with a CSR dictionary developed by Pencle and Malaescu (2016).

CSR Dictionary

Text Mining often uses lexical resources. A dictionary is a list of words that is relevant to a certain topic. It is important to use a specific dictionary for each topic. These can be simple word lists, dictionaries or even a thesaurus. The creation of such lists is very time-consuming and requires experts in the respective field. Similar to the POS-Tagger, where words are assigned to different word types, there can also be different domains/categories assigned to the words. It can happen that a word is assigned to several categories (Ignatow and Mihalcea 2018).

For CSR, Pencle and Malaescu (2016) published a dictionary that they used to predict metrics for IPOs. In doing so, they made sure that the dictionary could also be used for other analyses of company texts. The dictionary consists of four dimensions and a total of over 1,400 words. To determine the dimensions, the largest CSR reporting frameworks and guidelines were analyzed and finally the dimensions "Social and Community", "Employee", "Environment" and "Human Rights" were selected. To determine the words per dimension, an initial list was drawn up based on CSR literature, which was reviewed and supplemented by experts in several stages.

For this analysis, we used the dictionary on the given documents and counted the number of CSR-relevant words per document and per dimension. The following assumptions are made for the analysis:

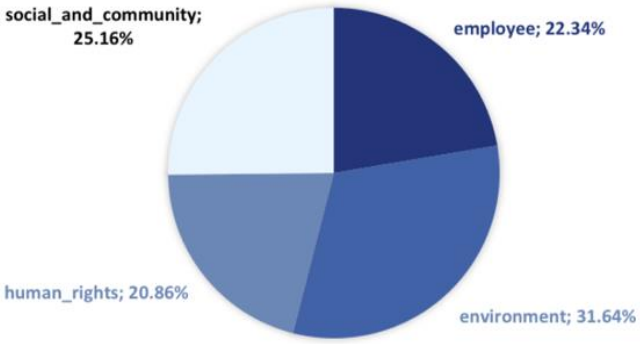
- Companies that want to go public must be attractive for investors. It can therefore be assumed that they include CSR-relevant information in their texts.
- A high CSR word count per text can be seen as an indicator of sustainable business practices and therefore leads to a higher company value (offering price).
- The companies are involved in numerous activities in various dimensions. All activities together form the CSR orientation of the company.

The dictionary is available online and can be downloaded, for example, using the R programming language. It is included in the package "lexicon" with the name "key_corporate_social_responsibility". The dictionary contains the three columns "dimension", "regex" and "token" (see Figure 6). "Token" is a placeholder for the actual word and with the help of "regex", you can easily search for the word in a text. A total of 1´421 words is included, which are distributed over the four dimensions, as shown in Figure 7. The dictionary contains tokens with only one word or tokens that are composed of several words (e.g., "water desalination"). A token can occur in several dimensions.

Figure 6. *CSR Dictionary*

dimension	regex	token
environment	\bwaste reduction\b	waste reduction
environment	\bwasteland\b	wasteland
environment	\bwater\b	water
environment	\bwater desalination\b	water desalination
environment	\bwater purification\b	water purification
environment	\bwater purifications\b	water purifications
environment	\bwave\b	wave
environment	\bweather\b	weather

Figure 7. *CSR Dimension Distribution*



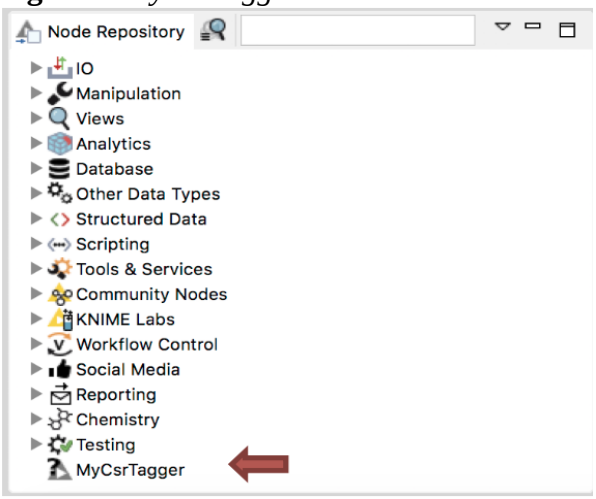
Tagging Records

In KMIME, besides the well-known POS taggers (POS Tagger, Standfort Tagger), there are also taggers for medical or chemical terms (Abner Tagger, Oscar Tagger). The *Wildcard Tagger* node can be used to tag these records and also allows you to use your own word list as input. This node has 12 tag types implemented, with the corresponding tag values to be used. These are common taggers, which are mainly used in the medical field. For CSR there is no integrated tag set available. However, KNIME offers the possibility to program your own node extension with your own tag set (KNIME 2023).

For programming the node extension, Eclipse, a development environment that is often used for Java programming, is needed. The same instructions as on the Sentimental analysis node example can be followed, only a few adjustments are required (KNIME 2023). The name of the project and the Java classes should be "MyCsrTagger", the TAG_TYPE should be "CSR" and the tagset should be the four categories from the CSR Dictionary by Pencle and Malaescu (2016): "Social_and_Community", "Employee", "Human_Rights" and "Environment".

Once all the steps in the instructions have been completed, the project must be saved as a .jar file and placed in the "dropins" folder of the KNIME installation folder. The next time KNIME is started, the new node *MyCsrTagger* will appear in the node repository (see Figure 8).

Figure 8. *MyCsrTagger Node*



The new CSR tag-set is now available in the *Wildcard Tagger* Node and in the *Tag Filter* Node, the four dimensions are provided (see Figure 9).

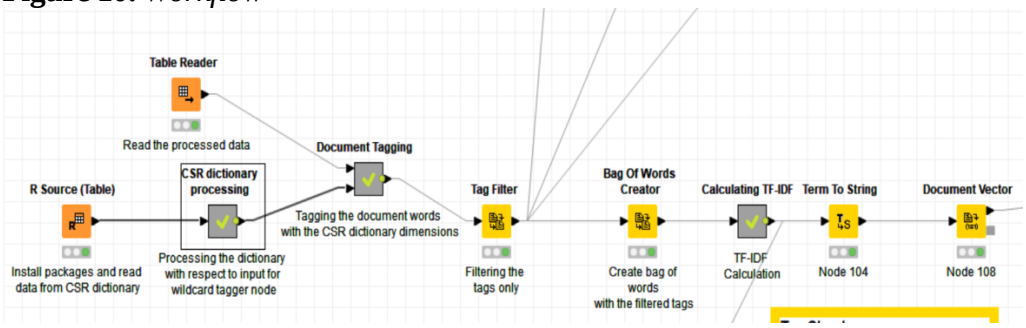
Figure 9. *CSR Tag-set*

To use the new extension with the four tag values, you have to select the category and enter the corresponding words as input into the node. But since the dictionary contains tokens with one or more words, the process must be designed accordingly. If the token with one word (e.g., "class") is found first, the token with more words (e.g., "working class"), which contains the first token, cannot be found anymore.

Therefore, the tokens in the dictionary are sorted in descending order according to the number of words per token. To implement this, the spaces per token are counted. Once these are known, the dictionary is sorted with respect to the number of spaces in descending order. This results in the tokens with several words being used to tag the documents first, while the tokens with only one word follow.

In the following workflow (see Figure 10), the dictionary is sorted into right format. The text in the preprocessed documents is tagged with the tokens in the dictionary.

Figure 10. *Workflow*



The *Document Tagging* meta-node contains the workflow where each document is tagged one by one in a recursive loop (see Figure 11). Within the Recursive Loop, the document is tagged by the *Wildcard Tagger*. Inputs are the documents and the dictionary. The *Row Splitter* passes only the top line of the dictionary to the *Wildcard Tagger*. The rest becomes input for the next iteration. The selected row is converted into variables to control the wildcard tagger and set the corresponding tag value. In each iteration only one word is tagged. The document is then returned to the start of the loop by the *Recursive Loop End* node, so that the document is tagged with all words at the end of the loop.

Figure 11. *Document Tagging Meta Node*

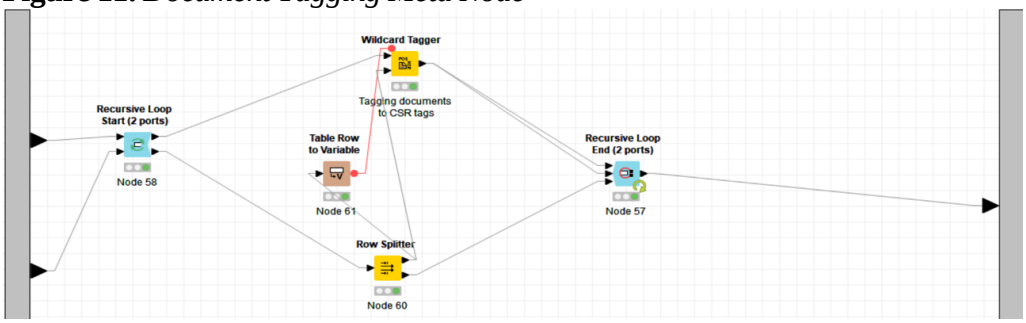


Figure 12. *Tagged Text*

UNKNOWN

responsible ***[CSR(Employee)]*** sourcing ***[CSR(Environment)]*** target ***[CSR(Environment)]*** principles ***[CSR(Employee)]*** guidelines ***[CSR (Environment)]*** design ***[CSR(Environment)]*** corporate ***[CSR(Environment)]*** design ***[CSR(Environment)]*** principles ***[CSR(Employee)]*** principles ***[CSR(Employee)]*** sourcing ***[CSR(Environment)]*** duty ***[CSR(Human_rights)]*** ownership ***[CSR(Human_rights)]*** duty ***[CSR (Human_rights)]*** duty ***[CSR(Human_rights)]*** transparency ***[CSR(Social_and_Community)]*** sourcing ***[CSR(Environment)]*** farmer ***[CSR (Environment)]*** duty ***[CSR(Human_rights)]*** duty ***[CSR(Human_rights)]*** reliability ***[CSR(Social_and_Community)]*** human ***[CSR (Social_and_Community)]*** responsible ***[CSR(Employee)]*** responsible ***[CSR(Employee)]*** employment ***[CSR(Human_rights)]*** freedom ***[CSR (Human_rights)]*** freedom ***[CSR(Human_rights)]*** employment ***[CSR(Human_rights)]*** equal ***[CSR(Human_rights)]*** respect ***[CSR (Social_and_Community)]*** safety ***[CSR(Human_rights)]*** health ***[CSR(Employee)]*** environment ***[CSR(Employee)]*** environmental ***[CSR (Environment)]*** social ***[CSR(Human_rights)]*** care ***[CSR(Human_rights)]*** nature ***[CSR(Social_and_Community)]*** stewardship ***[CSR (Environment)]*** knowledge ***[CSR(Employee)]*** transparency ***[CSR(Social_and_Community)]*** duty ***[CSR(Human_rights)]*** transparency ***[CSR (Social_and_Community)]*** duty ***[CSR(Human_rights)]*** care ***[CSR(Human_rights)]*** duty ***[CSR(Human_rights)]*** care ***[CSR(Human_rights)]*** people ***[CSR(Human_rights)]*** environment ***[CSR(Employee)]*** farm ***[CSR(Environment)]*** family ***[CSR(Human_rights)]*** farm ***[CSR (Environment)]*** responsible ***[CSR(Employee)]*** sourcing ***[CSR(Environment)]*** agricultural ***[CSR(Environment)]*** worker ***[CSR(Human_rights)]*** health ***[CSR(Employee)]*** respect ***[CSR(Social_and_Community)]*** gender ***[CSR(Human_rights)]*** empowerment ***[CSR(Human_rights)]*** principles ***[CSR(Employee)]*** seasonal ***[CSR(Employee)]*** management ***[CSR(Employee)]*** nature ***[CSR(Social_and_Community)]*** conservation ***[CSR (Environment)]*** quality ***[CSR(Employee)]*** water ***[CSR(Social_and_Community)]*** management ***[CSR(Employee)]*** practices ***[CSR(Employee)]*** farm ***[CSR(Environment)]*** water ***[CSR(Social_and_Community)]*** management ***[CSR(Employee)]*** water ***[CSR(Social_and_Community)]*** responsible ***[CSR(Employee)]*** management ***[CSR(Employee)]*** biodiversity ***[CSR(Environment)]*** management ***[CSR(Employee)]*** health ***[CSR (Employee)]*** preservation ***[CSR(Human_rights)]*** management ***[CSR(Employee)]*** farm ***[CSR(Environment)]*** cultivation ***[CSR (Environment)]*** animal ***[CSR(Environment)]*** experience ***[CSR(Employee)]*** animal ***[CSR(Environment)]*** animal ***[CSR(Environment)]*** welfare ***[CSR(Employee)]*** freedom ***[CSR(Human_rights)]*** freedom ***[CSR(Human_rights)]*** freedom ***[CSR(Human_rights)]*** freedom ***[CSR (Human_rights)]*** freedom ***[CSR(Human_rights)]*** responsible ***[CSR(Employee)]*** sourcing ***[CSR(Environment)]*** responsible ***[CSR(Employee)]*** sourcing ***[CSR(Environment)]*** responsible ***[CSR(Employee)]*** sourcing ***[CSR(Environment)]*** care ***[CSR(Human_rights)]*** respect ***[CSR (Social_and_Community)]*** enhancing ***[CSR(Human_rights)]*** communities ***[CSR(Human_rights)]*** quality ***[CSR(Employee)]*** future ***[CSR (Social_and_Community)]*** responsible ***[CSR(Employee)]*** sourcing ***[CSR(Environment)]*** requirements ***[CSR(Environment)]*** code ***[CSR (Environment)]*** responsible ***[CSR(Employee)]*** sourcing ***[CSR(Environment)]*** parties ***[CSR(Human_rights)]*** responsible ***[CSR(Employee)]*** materials ***[CSR(Environment)]*** sustainable ***[CSR(Social_and_Community)]*** agricultural ***[CSR(Environment)]*** reduce ***[CSR (Social_and_Community)]*** guidelines ***[CSR(Environment)]*** multinational ***[CSR(Employee)]*** core ***[CSR(Human_rights)]*** sourcing ***[CSR (Environment)]*** sustainable ***[CSR(Social_and_Community)]*** development ***[CSR(Human_rights)]*** goals ***[CSR(Employee)]*** services ***[CSR (Employee)]*** principles ***[CSR(Employee)]*** people ***[CSR(Human_rights)]*** communities ***[CSR(Human_rights)]*** regulations ***[CSR(Human_rights)]*** sourcing ***[CSR(Environment)]*** activities ***[CSR(Human_rights)]*** sourcing ***[CSR(Environment)]*** activities ***[CSR(Human_rights)]*** improve ***[CSR (Social_and_Community)]*** practices ***[CSR(Employee)]*** parties ***[CSR(Human_rights)]*** requirements ***[CSR(Environment)]*** help ***[CSR (Social_and_Community)]*** improve ***[CSR(Social_and_Community)]*** practices ***[CSR(Employee)]*** projects ***[CSR(Social_and_Community)]***

Figure 12 shows a text from the analysis data set that was tagged with the CSR Dictionary. All four dimensions can be seen in the text.

Once the documents are tagged, we filter out the terms from the documents that are only relevant with our CSR dictionary and create a bag-of-words with the filtered tags. The *Bag of words creator* node helps us to collect these terms in one bag and term frequency is calculated with respect to each document to generate a Tag cloud. The Tag cloud helps to visualize the terms that have more weightage i.e., that are occurring more frequently in the documents as shown in Figure 13.

Figure 13. *Tag Cloud*

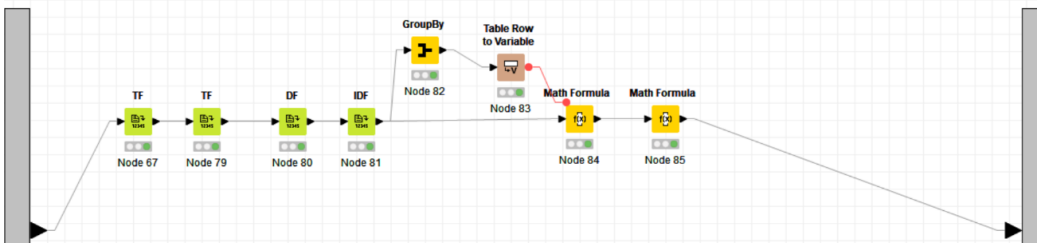
Model Development and Analysis

The models created for this analysis are based on a CSR Dictionary (lexicon-based approach). All documents have been tagged with this dictionary so that it is possible to filter according to the CSR tags. The analysis assumes that, on the one hand, only these terms are relevant for the classification and, on the other hand, that the more CSR terms occur, the more likely it is that texts are relevant.

Lexicon Based Approach with Decision Tree

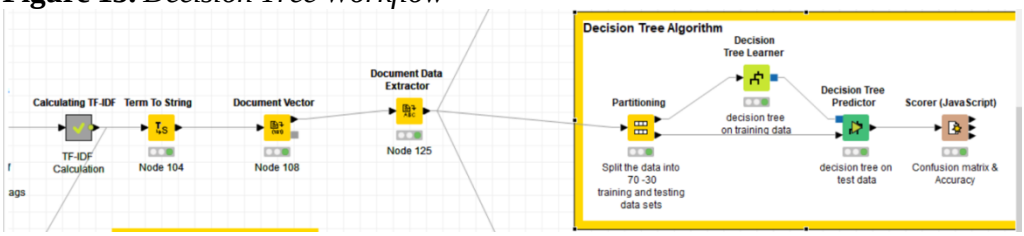
With this approach, the tagged analysis data is first used to calculate the key figure TF-IDF per term. For the calculation of the TF-IDF ratios per term, the ratios TF_abs, TF_rel, DF_abs, DF_rel and IDF are calculated in the *Calculating TF-IDF* metanode (see Figure 14). To reduce the features, i.e. the terms, two filters are used. The first filter reduces the features using the key figure TF_abs. As second filter, the DF_rel value per term is used. Only terms with a DF_rel value between 0.1 and 1.0 are retained.

Figure 24. *TF-IDF Calculation*



Finally, as shown in Figure 15, the data is converted into a document matrix using the *Document vector* node. *Document data extractor* is used to extract the class of the documents which has been added manually during the data preparation phase for classification of the documents. Finally, the analysis dataset is divided into two parts by the node *Partitioning*. In this case, 70% of the data is used for the learner and 30% for the predictor. For this purpose, stratified sampling, i.e. that the distribution of the target variable, is the same in both parts, and a random seed (1578008215742) is chosen so that the results are repeatable.

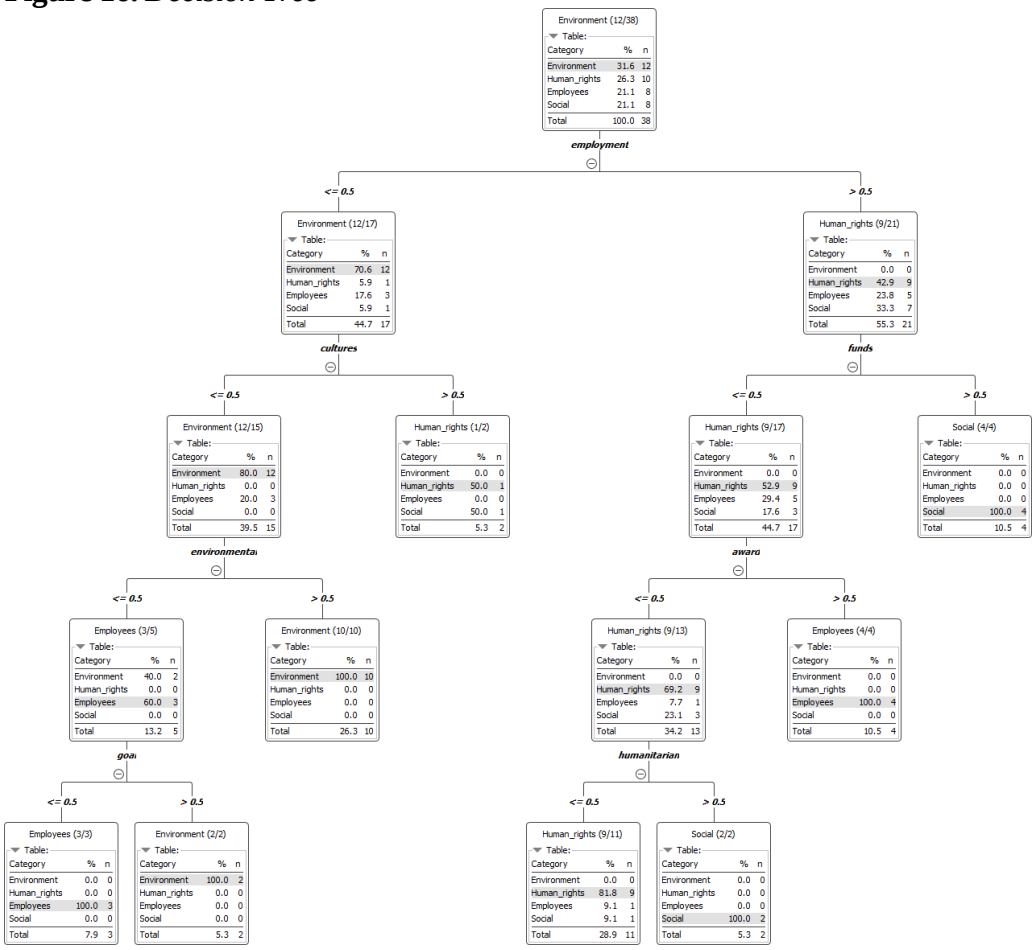
Figure 15. *Decision Tree Workflow*



The Decision Tree Learner creates a model which is used for the training data to measure the quality and for the test data to make a prediction. In the node configuration, ‘Gain ratio’ is provided as a quality measure instead of ‘Gini index’ for better performance. The model has an accuracy of 0.529 and a Cohen's Kappa value of 0.352. These values are acceptable and therefore, the model can be used for making predictions. Figure 16 shows the decision tree for the model. In total, there are only seven leaves, which are created by dividing the documents by the words "employment", "cultures", "funds", "environmental", "award", "goal" and “humanitarian”.

For the above analysis, it was described how a predictive model was built to predict the labels of documents with respect to CSR tags (environment, employee, social and human rights). In this approach, single words were used as features. It is shown that the most discriminative terms w.r.t separating the four classes are "environment", "waste", and "rights". If the term "waste" occurs in a document, it is likely to belong to the environment class. If "goal" occurs, it is equally likely to belong to the environment or employee class, etc.

Figure 16. *Decision Tree*



N-gram Approach

The single word approach may have several drawbacks. It is very likely that a single term analysis could lead to misclassification of data (Thiel 2016). Therefore, using frequent n-grams as features in addition to single words can overcome this problem. In order to analyze the terms that provide better understanding of the documents, the n-gram approach is used.

In this section, 1- and 2-grams as features for prediction are used in order to check if a 2-gram approach helps to increase the accuracy of the model in comparison with only single word features (see Figure 17). The KNIME Text Processing extension is used in combination with the traditional KNIME learner and predictor nodes to process the textual data as well as build and score the predictive model. For the analysis, the same approach is used as in the previous model for data preprocessing and after tagging the data, a *N-Gram creator* node is used to generate the terms that are more likely to occur together than separately (see Figure 18). A slight change is made to the configuration of the decision tree predictor node, where the 'Gini index' is used instead of the 'Gain ratio' in 2-gram and 1- and 2-gram approaches in order to achieve better accuracy of the models.

Figure 17. *N-grams*

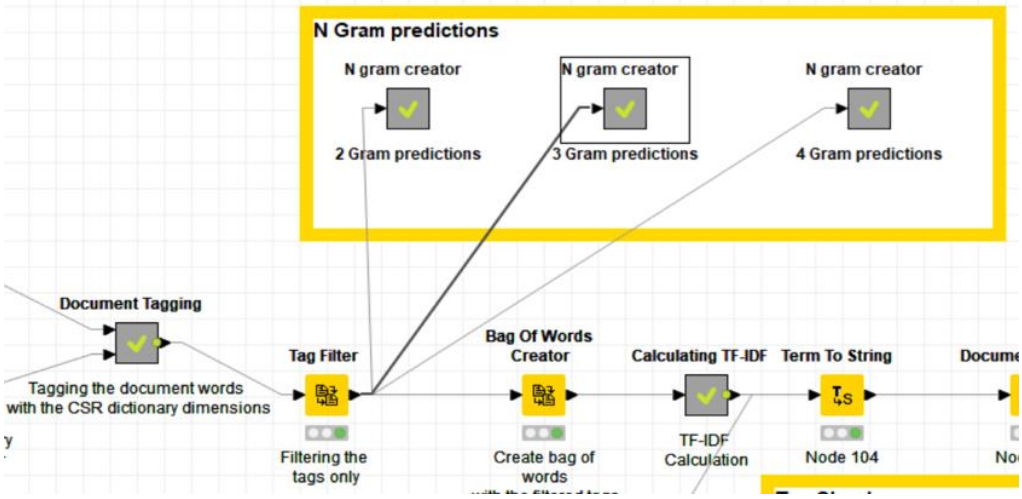
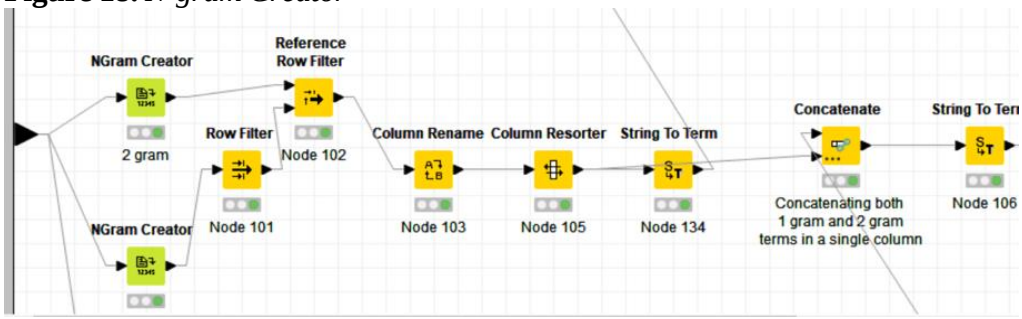
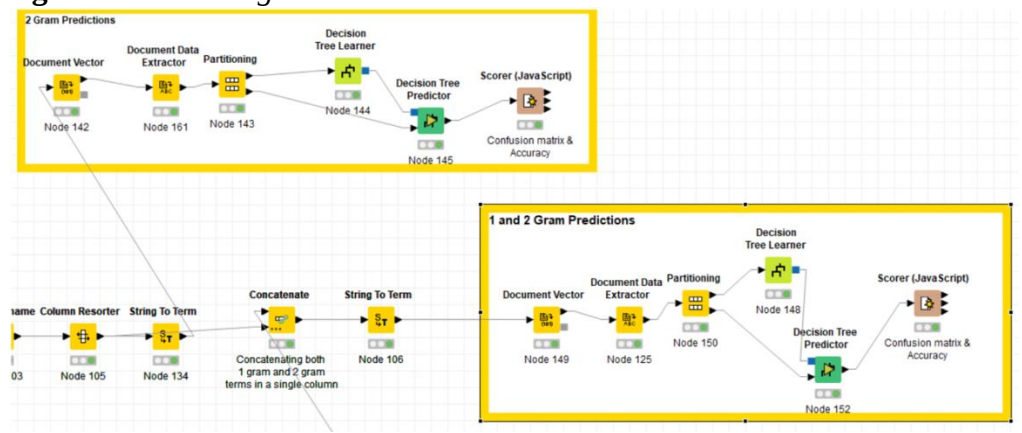


Figure 18. *N-gram Creator*



In the 2-gram predictions, the meta node has two input ports and the *String to Term* node has two output ports. In the top output port, document vectors are provided with two-word features only. In the bottom output port, document vectors with 1- and 2-gram features are provided (see Figure 19). These vectors are used for classification in the next steps. The vectors containing 2-grams only, consist of 265 features. The vectors with 1- and 2-grams consist of 551 features. Taking the 2-grams into account doubles the feature space in this case.

Figure 19. *1- and 2-gram Predictions*



For both branches (top with 2-grams as features only, bottom with 1- and 2-gram features), predictive models are built, tested and compared. For classification, any of the traditional mining algorithms available in KNIME can be used (e.g. decision trees in this case). When the documents have been created, the target variable is stored as a category in the documents. To extract the target variable as a column, the *document extraction* node is used. The *Partitioning* node splits the dataset into training (70%) and test (30%) sets in both branches.

At the end, two decision trees are built on the training sets and applied on the test sets (see Figures 20-21). The decision tree in the top branch, trained on the data set with 2-gram features, only achieved an accuracy of 0.529. The decision tree of the bottom branch, trained on the data set with 1- and 2-gram features, achieved an accuracy of 0.588. Cohen's kappa delivers a value of 0.346 and 0.449 respectively.

Figure 20. *1- and 2-gram Decision Tree*

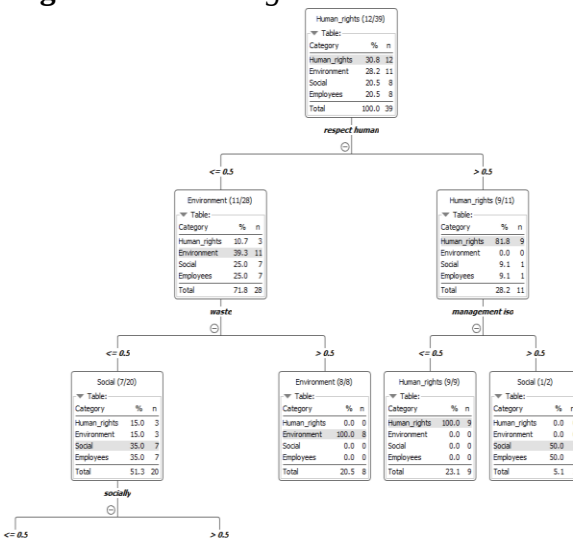
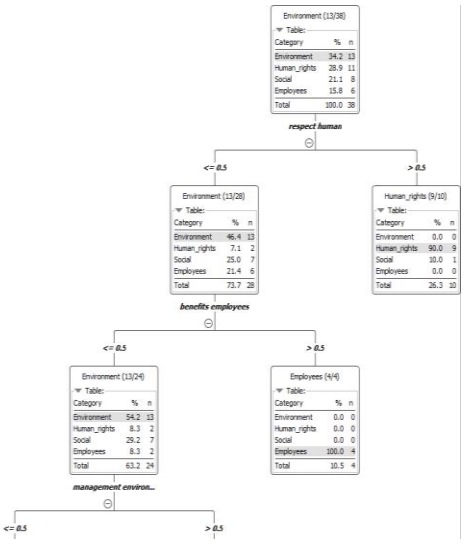


Figure 21. *2-gram Decision Tree*

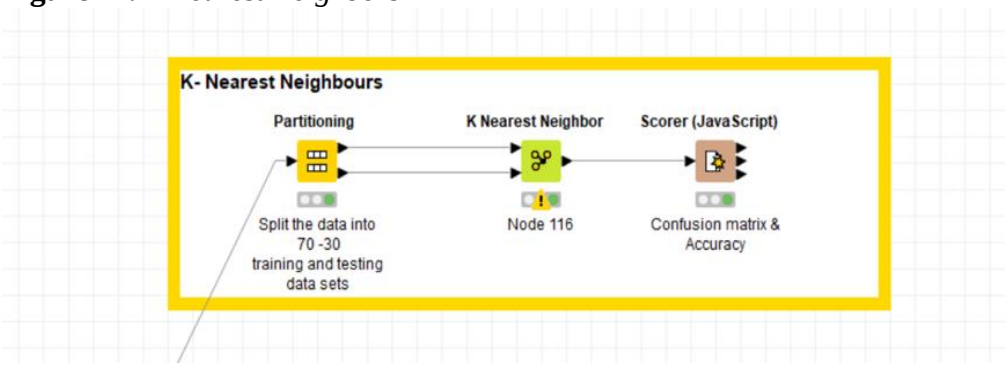


The same approach is applied on 3- and 4- gram models which resulted in less accuracy and statistically insignificant results.

Lexicon Based Approach with K-nearest Neighbor

As in the previous approach, the data set for this approach is converted into a document matrix, with the TF-IDF values as elements of the matrix. Again, the columns can be harmonized. To test the model, the analysis data is separated by the node *Partitioning* (see Figure 22). For this approach, 70% of the data is used for training the model and 30% for testing. For the k-nearest-neighbor approach the 4 nearest neighbors (k=4) are used for prediction. Although this number is not odd, as generally odd numbers are preferred for this approach, it provides the best results nonetheless (a test with different values for k was performed). With these settings, an accuracy of 0.471 and a Cohen's Kappa of 0.292 could be achieved. These results indicate that this approach is acceptable for the classification of training agendas but provides less statistically significant results in terms of Cohen's kappa and accuracy.

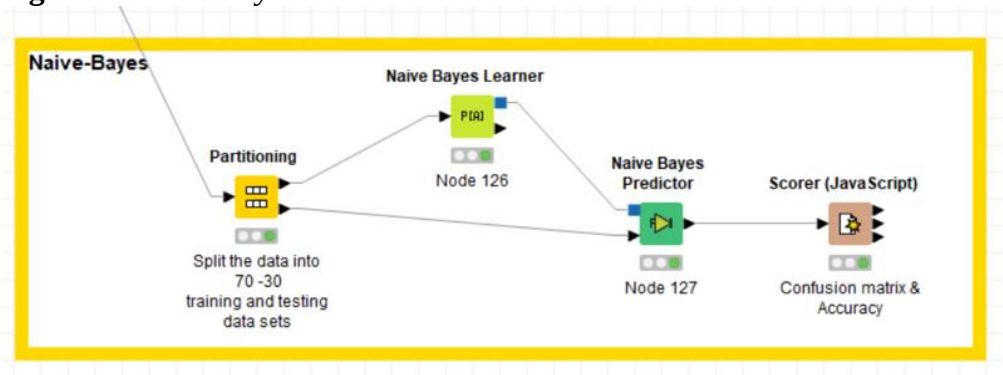
Figure 22. *K-nearest Neighbors*



Lexicon Based Approach with Naïve Bayes

As in the two previous approaches, the data sets are brought into a document matrix and the columns are harmonized. For this approach, the data is again divided into 70% for training and 30% for testing by the *Partitioning* node. Again, the stratified sampling approach is used, with a random seed (1610365614571). The Naive Bayes Learner creates a model that is used in the Naive Bayes Predictor (see Figure 23). For the training data an accuracy of 0.527 and a Cohen's Kappa of 0.355 can be achieved. These results show that the model does achieve good classifications in comparison to the two previous models.

Figure 23. *Naive Bayes*



Comparison of the Models

A comparison of the results of these different models shows that except for the K-nearest neighbor model, the values for Accuracy and Cohen's Kappa are comparable (see Table 1). The decision tree model with a 1-and 2- gram approach delivers the best results.

Table 1. *Model Comparison*

Model	Accuracy	Cohen's Kappa
Decision Tree	0.53	0.36
2-gram (decision tree)	0.53	0.35
1 & 2 gram (decision tree)	0.59	0.45
K-nearest neighbors	0.47	0.30
Naïve Bayes	0.53	0.36

Conclusion and Future Work

The importance of Corporate Social Responsibility has increased over the last years. Nowadays, large, established companies often include CSR in their strategy due to the ever-increasing pressure from the public. The models developed in this paper help us to identify the CSR relevant terms in a company's report.

The basis of the analysis is a CSR dictionary published in 2016 by Pencle and Malaescu (2016). We used this dictionary to analyze documents for an IPO. In a first step, the CSR terms in the dictionary were divided into four dimensions, which are later used as target categories for the analysis. The use of the mentioned dictionary resulted in a reduction of number of words per text, improved workflow performance and general improvement of models based solely on these words.

Three of the four used models (N-gram models, Decision tree and Naive Bayes) showed good results. These models had an accuracy of around 53% – 59% and Cohen's Kappa values of over 0.45 when applied to the analyzed data. Only the model with k-nearest-neighbor was less accurate. Data classification by this model should therefore be interpreted with caution. The decision tree with 1-gram models showed more descriptive results with more branches and leaves with equal

weightage on both the left- and the right-hand side of the tree. Whereas, in 2-grams only and 1- and 2- grams models, the decision tree is less descriptive and has more weightage on left hand side of the trees.

In order to validate the aforementioned results and to know why the trees are so different from when using the 1 term or 1-gram approach, a more in-depth analysis is needed. Furthermore, it is important to bear in mind that the success of a model depends heavily on the quality and quantity of the pre-classified analysis data. To obtain more reliable results, it is recommended that to use a larger dataset. This is also shown by the fact that the choice of a different random seed in the partitioning of the analysis data leads to very different results (for example when using the k-nearest-neighbor model).

For future work, large data sets are recommended for the validation of the above models. It is also possible to enhance the CSR dictionary with more terms which can map to different dimensions as per the business case. For enhancing the terms in the dictionary, a table must be created with the exact columns as in the CSR dictionary (see Figure 24).

Figure 24. *Creation of CSR Dictionary with New Terms*

File

Table Creator Settings | Flow Variables | Job Manager Selection | Memory Policy

Input line:

	regex	tokens	dimensions	tagtype
Row0	\bseed\b	seed	Employee	CSR
Row1	\bselection\b	selection	Human_rights	CSR
Row2	\bsensitivity\b	sensitivity	Environment	CSR
Row3	\bserved\b	served	Environment	CSR
Row4	\bserves\b	serves	Environment	CSR
Row5	\bservice\b	service	Human_rights	CSR
Row6	\bservices\b	services	Human_rights	CSR
Row7	\bsexually\b	sexually	Employee	CSR
Row8	\bshipyards\b	shipyards	Human_rights	CSR
Row9	\bshore\b	shore	Employee	CSR
Row10	\bshrinking\b	shrinking	Social_and_C...	CSR
Row11	\bsick\b	sick	Environment	CSR
Row12	\bsite\b	site	Environment	CSR
Row13	\bsize\b	size	Environment	CSR
Row14	\bsocial\b	social	Environment	CSR
Row15	\bsocially\b	socially	Social_and_C...	CSR
Row16	\bsocietal\b	societal	Social_and_C...	CSR
Row17	\bsolar\b	solar	Human_rights	CSR
Row18	\bsolubility\b	solubility	Employee	CSR
Row19	\bsolvents\b	solvents	Environment	CSR
Row20	\bsourcing\b	sourcing	Human_rights	CSR
Row21	\bsponsors'\b	sponsors'	Employee	CSR
Row22	\bsponsorship\b	sponsorship	Employee	CSR

The output table has 93 rows and 4 columns.

In order to customize the tagging dimensions, one can e.g. build any other dimensions instead of ‘environment’, ‘employee’, ‘human rights’ and ‘social responsibility’. This can be achieved by simply building a node extension and following the same steps as described in the section on CSR Dictionary. It is recommended to pay more attention during the data cleaning phase as occasionally, identical terms (e.g., ‘time time’ or ‘human human’) will occur in the decision tree while using the 2-gram approach. This could lead to confusion and doubts in the

descriptiveness of the models. So, for future work, it is recommended to pay attention on data when applying a 2-gram approach.

Finally, it is safe to say that this work serves as a strong base for future projects that are focused on analyzing text data for different CSR dimensions and gathering qualitative information with respect to these dimensions.

References

Aureli S (2016) A comparison of content analysis usage and text mining in CSR corporate disclosure. *The International Journal of Digital Accounting Research* 17: 1–32.

Cleve J, Lämmel U (2016) *Data mining* (2nd ed.). De Gruyter Oldenbourg.

Ghosh S, Roy S, Bandyopadhyay SK (2012) A tutorial review on Text Mining Algorithms. *International Journal of Advanced Research in Computer and Communication Engineering* 1(4): 223–233.

Guerreiro J, Rita P, Trigueiros D (2016) A text mining-based review of cause-related marketing literature. *Journal of Business Ethics* 139(1): 111–128.

Hearst MA (1997, July) *Text data mining: Issues, techniques, and the relationship to information access.* Presentation for UW/MS workshop on data mining, Berkeley, CA, United States.

Ignatow G, Mihalcea R (2018) *An Introduction to text mining: Research Design, Data Collection, and Analysis*. Los Angeles, CA: Sage.

KNIME (2023, November 22) *Create a New Java based KNIME Extension*. Available at: https://docs.knime.com/latest/analytics_platform_new_node_quickstart_guide/index.html#_introduction.

Lin S-J, Hsu M-F (2018) Decision making by extracting soft information from CSR news report. *Technological and Economic Development of Economy* 24(4): 1344–1361.

Miner G, Delen D, Elder J, Fast A, Hill T, Nisbet R (2012) *Practical text mining and statistical analysis for non-structured text data applications*. 1st Edition. Elsevier.

Pencle N, Mălăescu I (2016) What's in the words? Development and validation of a multidimensional dictionary for CSR and application using prospectuses. *Journal of Emerging Technologies in Accounting* 13(2): 109–127.

Sayad S (2024) *Decision Tree – Classification*. Available at: https://www.saedsayad.com/decision_tree.htm.

Song Y, Wang H, Zhu M (2018) Sustainable strategy for corporate governance based on the sentiment analysis of financial reports with CSR. *Financial Innovation* 4(2).

Tan AH (1999) Text mining: The state of the art and the challenges. In *Proceedings of the pakdd 1999 workshop on knowledge discovery from advanced databases*. Beijing, 8, 65–70.

Te Liew W, Adhitya A, Srinivasan R (2014) Sustainability trends in the process industries: A text mining-based analysis. *Computers in Industry* 65(3): 393–400.

Thiel K (2016, January 8) *Sentiment Analysis with N-grams*. KNIME. Available at: https://www.knime.com/blog/sentiment-analysis-with-n-grams.