Classifying News Media Coverage for Corruption Risks Management with Deep Learning and Web Intelligence

Albert Weichselbraun
Sandro Hörler
albert.weichselbraun@fhgr.ch
sandro.hoerler@fhgr.ch
University of Applied Sciences of the Grisons
Chur, Switzerland

Christian Hauser
Anina Havelka
christian.hauser@fhgr.ch
anina.havelka@fhgr.ch
University of Applied Sciences of the Grisons
Chur, Switzerland

ABSTRACT

A substantial number of international corporations have been affected by corruption. The research presented in this paper introduces the *Integrity Risks Monitor*, an analytics dashboard that applies Web Intelligence and Deep Learning to english and german-speaking documents for the task of (i) tracking and visualizing past corruption management gaps and their respective impacts, (ii) understanding present and past integrity issues, (iii) supporting companies in analyzing news media for identifying and mitigating integrity risks.

Afterwards, we discuss the design, implementation, training and evaluation of classification components capable of identifying English documents covering the integrity topic of corruption. Domain experts created a gold standard dataset compiled from Anglo-American media coverage on corruption cases that has been used for training and evaluating the classifier. The experiments performed to evaluate the classifiers draw upon popular algorithms used for text classification such as Naïve Bayes, Support Vector Machines (SVM) and Deep Learning architectures (LSTM, BiLSTM, CNN) that draw upon different word embeddings and document representations. They also demonstrate that although classical machine learning approaches such as Naïve Bayes struggle with the diversity of the media coverage on corruption, state-of-the art Deep Learning models perform sufficiently well in the project's context.

CCS CONCEPTS

• **Information systems → Data analytics**; • **Computing methodologies → Neural networks**; • **Applied computing → Economics**; **Annotation**.

KEYWORDS

Web Intelligence, Corruption Risk Management, Text Classification, Text Analytics, Deep Neural Networks, Word Embeddings

WIMS 2020, June 30-July 3, 2020, Biarritz, France

ACM ISBN 978-1-4503-7542-9/20/06…$15.00
https://doi.org/10.1145/3405962.3405988

ACM Reference Format:
Albert Weichselbraun, Sandro Hörler, Christian Hauser, and Anina Havelka. 2020. Classifying News Media Coverage for Corruption Risks Management with Deep Learning and Web Intelligence. In *The 10th International Conference on Web Intelligence, Mining and Semantics (WIMS 2020), June 30-July 3, 2020, Biarritz, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3405962.3405988

1 INTRODUCTION

Risks jeopardizing the integrity of an organization are widespread. According to a 2018 study by PricewaterhouseCoopers which covered over 7,200 companies across 123 territories, approximately 50% of the corporation haven been affected by illegal and unethical behavior, such as embezzlement, cybercrime, corruption, fraud, money laundering and anti-competitive agreements. The financial and social impact of these incidents is breathtaking and calls for preventive actions that address these issues.

The research presented in this paper applies Web Intelligence and Deep Learning to the task of supporting companies in identifying and mitigating integrity risks. Historical data is used for training different classifiers to recognize national and international media coverage on corruption. Afterwards, we plan to apply transfer learning techniques to the task of adapting the classifier to a wide range of integrity topics such as human rights, labor conditions and sustainability. The adapted classifier assigns scores to News articles that indicate their relevance to the topic of integrity. Sophisticated visual tools then use the annotated documents for (i) tracking and visualizing past integrity management gaps and their respective impacts, (ii) identifying whether organizations have been mentioned positively or negatively in these events, and (iii) leveraging media coverage on upcoming integrity stories for predicting and discovering existing blind spots within a company's governance.

1.1 Integrity Risks Monitor

Figure 1 illustrates the multi-lingual *Integrity Risks Monitor* dashboard that supports domain experts in analyzing, browsing, visualizing and understanding media coverage on integrity risks. The dashboard draws upon methods developed by Scharl et al. [15] to support searching and browsing past and current documents, automatic computation of concepts that are associated with search terms within a given time spawn, metadata on the documents' sources and languages, and a list of search results, matching the query. Visualizations further aid experts in analyzing the result set. The frequency graph at the top of the page, summarizes trends

Figure 1: A screenshot of the advanced *Integrity Risks Monitor* dashboard for the search query "corruption", covering media coverage between 10 January and 10 March 2020.

in media coverage over the selected time period. The geographic visualization on the right shows the distribution of corruption coverage across countries, using color coding to indicate the amount of positive versus negative coverage. The word cloud and keyword graphs below provide an overview of the most important concepts present in the search results and their connections to each other.

Currently, the system supports English and German documents and translates all aggregated metrics such as the concepts shown in the word cloud and keyword graphs to the analyst's language, regardless of the language used in the source documents.

In addition, the analytics dashboard provides powerful tools for performing drill-down analyses: clicking on any point or keyword in the visualizations, provides a context menu that describes the main concepts associated with this point (see Figure 2) and allows the specification of additional search filters. These tools enable domain experts to (i) quickly narrow down even comprehensive document collections to manageable document sets, and to (ii) perform drill-down analyses for determining the reasons behind relations and trends shown in the aggregated visualizations.

1.2 Identifying Media Coverage on Integrity Risks

The *Integrity Risks Monitor* draws upon documents obtained from popular Austrian, German, Swiss, U.K. and U.S. media outlets that are pre-processed, analyzed and semantically enriched using methods such as part-of-speech tagging, dependency parsing [21], sentiment analysis [18], keyword analysis [20] and named entity linking [19]. In addition, we assign a score to each document that indicates its likelihood to contain coverage on integrity risks. The *Integrity Risks Monitor* allows filtering queries based on this "integrity score" which in turn (i) reduces ambiguities in search queries (e.g., the query concept "CO_2" refers to carbon dioxide that is a frequently used indicator for media coverage on environment and sustainability issues. Nevertheless, "CO_2 pistol" would also match the query but yield a considerably lower integrity score, indicating that documents that contain this concept are probably not relevant to the concept of integrity risks), and (ii) allows performing advanced analyses such as analytics on terms that are associated with a high integrity score which is useful for identifying new trends and existing blind spots.

Reliable systems for computing the integrity scores, particularly methods capable of minimizing the number of false positives are

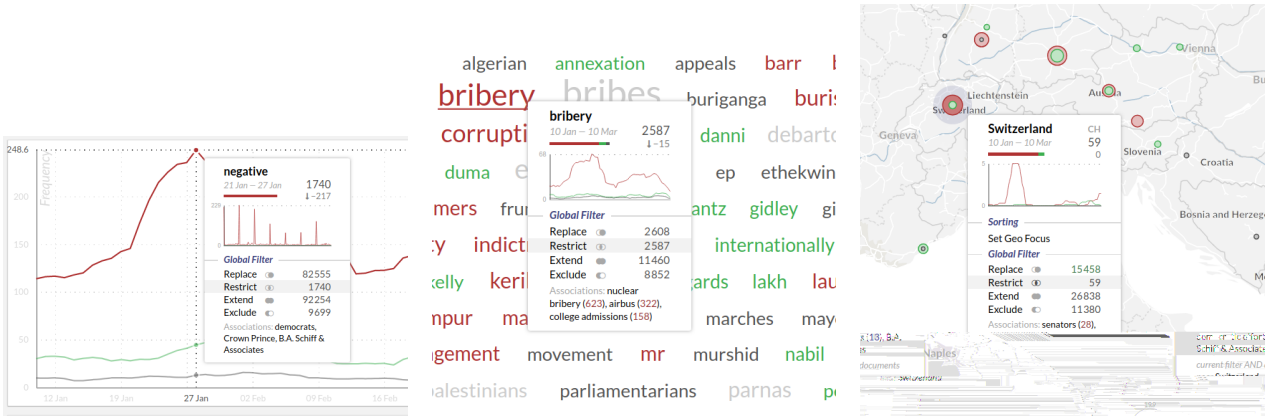

Figure 2: Context information and drill-down analyses provided by the *Integrity Risks Monitor* dashboard for trend analysis (left), keywords (middle) and locations (right).

essential for performing these sophisticated analyses, since irrelevant documents yield concepts and relations that distort aggregated analyses computed by the Integrity Risk Monitor.

The research presented in this paper focuses on the computation of the integrity score for (i) English documents and (ii) the topic of corruption which has been implemented within the presented framework to serve as a proof of concept for the feasibility of the approach.

The rest of the paper is structured as follows: Section 2 outlines related work in the domains of Web Intelligence and on the impact of Deep Learning and language models on text classification, Section 3 then introduces the methods applied to detecting media coverage on corruption, and Section 4 summarizes the experiments performed for evaluating these methods and discusses the results. The paper concludes with a summary and an outlook in Section 5.

2 RELATED WORK

2.1 Web Intelligence

Web Intelligence combines Artificial Intelligence with advanced information technology to create new web-based products and services. Web content mining and web monitoring apply these technologies to the task of analyzing web and social media streams, covering many domains, data sources and applications.

Scharl et al. [15] develop semantic systems and visual tools for analyzing and supporting stakeholder communication. They investigate how Web Intelligence platforms that analyze the media and social media coverage on climate change have been used to support stakeholders such as scientists, communication professionals and journalists in disseminating and discovering insights on climate change. Diakopoulos [3] discusses the use of computational news discovery tools which focus on identifying potentially newsworthy events or information prior to their publication in News media. His work grants insights into the interaction of journalists with such platforms and provides guidelines for their effective design. Ranganath et al. [13] research strategies for identifying advocates for political campaigns on social media. They use social movement theories to design a quantitative framework for studying nuanced messaging and propagation strategies as well as the community structure adopted by advocates for their campaigns.

Web Intelligence systems have also been successfully deployed to domains such as finance [23], sports [7], video verification based on stories from social media streams [9] and even the analysis of works of fiction [16].

All these systems face at least two major challenges: (i) harvesting, analyzing and identifying relevant content from Web sources that are heterogeneous in terms of authorship, formatting, style (e.g., news article versus tweets) and update frequency (weekly, daily or real-time); and (ii) providing interactive interfaces to select relevant subsets of the information space, and to analyze and manipulate the extracted data [16]. The research presented in this paper focuses on the first of these challenges, i.e. the identification of relevant content in the context of corruption risks management.

2.2 Deep Learning

Deep Learning has gained considerably in traction in recent years, since it enables multi-level automatic feature representation learning [22] and does not require domain experts to manually hand-craft features. In the context of natural language processing convolutional neural networks (CNN), recurrent neural networks (RNN) such as long short-term memory (LSTM), gated recurrent units (GRU) and residual networks (ResNets), and recursive neural networks have been successful in addressing problems such as part-of-speech-tagging, named-entity recognition, sentiment analysis, and semantic role labeling. In addition, advanced new neural architectures such as Graph Neural Networks [5], Transformers [17] or Capsule Networks [14, 4] yield further performance improvements for complex natural language processing tasks such as multi-label text classification and question answering [24], whereas Transformer-based models like BERT [2] or its lite version Albert [6] are known for including some syntactic and semantic information out-of-the-box and for obtaining leading scores in a variety of other tasks like natural language inference, semantic role labelling, semantic parsing, pronoun resolution or relation extraction.

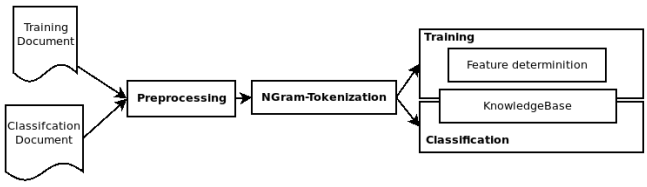

Figure 3: A simplified model of *Naive Bayes classifier* processing steps and components.

In terms of feature representation the shift from the vector space model and bag-of-words approaches towards word embeddings has been another significant development. Word embeddings draw upon the distributional hypothesis which states that words with similar meaning occur in similar context. They transfer vector space representations into a low-dimensional space which is considerably less sparse and also captures the similarity between words. Deep Learning and natural language processing toolkits (e.g. Keras[1], Torch[2] and AllanNLP[3]) support Word embeddings such as Word2Vec [8], Glove [10] and FastText [1], and researchers have published pre-trained embeddings that have been created based on comprehensive document collections. Recently, more sophisticated language models like BERT [2] and ELMo [11] have been developed that in addition to subword information also consider word context and provide built-in support for domain adaptation and fine-tuning based on task specific data. In some cases (e.g., when developing models for multi-task learning) it is better to extract word or sentence embeddings directly from these models instead of using classic word embeddings.

3 METHOD

This section elaborates on the machine learning models used for the corruption classification task and provides detailed information on the chosen pre-processing, document representation and model parameters.

We have selected the Naïve Bayes algorithm and Support Vector Machines (SVM) as baselines, since they have been extensively used for text classification and have shown to provide a decent performance for many classification tasks. The selection of the deep learning classifiers has been guided by recent reviews of the state of the art in deep learning for natural language processing [22, 12] which show that Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) are particularly well-suited for NLP tasks. Therefore, two Recurrent Neural Networks architectures (LSTM and BiLSTM) and a Convolutional Neural Network have been chosen for the design of the integrity classifier.

All deep learning models use an input layer of size of 300 which corresponds to the vector size of many available pre-trained word embeddings. The models' weight initialization strategy and the used optimizers have been selected based on the recommendations in the literature for the particular network type. In addition, extensive experiments helped in determining the model's other hyperparameters such as the size of the deep learning layers, the strategy for preventing overfitting (dropout and L2 regularization), and the number of epochs used for training.

3.1 Preprocessing

The pre-processing component removes stopwords, transforms all strings to lowercase and removes punctuation symbols. Afterwards, sentence splitting and tokenization segment the input text into sentences and tokens required for computing various input representations of the document such as the vector space model and different word embeddings.

3.2 Word Embeddings

We have conducted experiments with pre-computed embeddings as well as with custom, domain-specific embeddings. In these evaluations, the domain-specific embeddings clearly outperform pre-trained embeddings, although pre-trained models have been created based on considerably larger corpora.

Therefore, the presented evaluations only consider custom embeddings that have been trained on a corpus of English news media articles covering over 143,000 documents published in *The Guardian*, *The Times*, *The New York Time* and *The Wall Street Journal* between 1995 and 2019.

3.2.1 Word2Vec Embedding. Similar to the setup chosen by Mikolov et al. [8] the Word2Vec embedding translates words into a vector representation of size 300 and uses a context window of size five. In addition, the training performs five iterations and uses only tokens that appeared at least five times.

3.2.2 FastText. This embedding treats words as compositions of characters and, therefore, also considers subwords. These subword units frequently allow FastText to construct vectors for out of vocabulary words. Similar to the Word2Vec embedding, the custom FastText embedding has been created with 300 dimensions, considers word n-grams of up to five words and has been trained for five epochs.

3.3 Naïve Bayes

The Naïve Bayes algorithm (Figure 3) is a widely used probabilistic classifier and serves as a baseline in our experiments. The chosen pre-processing removes stopwords, transforms all strings to lowercase and removes punctuation symbols. The model uses a bag of word representation of the document and extracts n-gram features with a maximum size of up to three tokens. Once trained, the algorithm uses the 50 most significant features, extracted from a document, for its classification.

[1] https://keras.io/
[2] http://torch.ch/
[3] https://allennlp.org/

3.4 Support Vector Machine

Support Vector Machines (SVM) have been particularly successful for classifying text documents. In the presented experiments we use the Java version of LIBSVM[4] in conjunction with the following two feature representations: (i) a bag of word feature representation (Section 3.4.1) and (ii) word embeddings (Section 3.4.2). All models used a Radial Basis Function (RBF) kernel with Laplace pre-processing.

3.4.1 Bag of words. The bag of words approach uses the pre-processing introduced for the Naïve Bayes algorithm (stopword removal, strings are translated to lowercase, removal of punctuation). The dynamic feature extraction component then selects all words that appear at least five times as possible features which are then translated into a vector space presentation and used for training the SVM (Figure 4).

3.4.2 Word Embeddings. The second approach uses word embeddings as features that have been created with Word2Vec [8] and FastText [1] (Section 3.2). The feature extraction component translates all words into the 300 dimensional embedding space and then summarizes them to the document vector. Normalizing the document vector yields the input features for the SVM algorithm.

3.5 Long Short Term Memory Classifier (LSTM)

As outlined in Table 1, the Long Short Term Memory classifier (LSTM) uses an LSTM layer in conjunction with an RNN output layer and draws - depending on the setting - either upon Word2Vec or FastText word embeddings. The classifier uses the following hyperparameters:

(1) Optimizer: Adam optimizer (learning rate $\alpha = 5 \cdot 10^{-3}$)
(2) Weight initialization strategy: Xavier
(3) L2 regularization (0.0001) and a dropout value of 0.5 to limit overfitting.
(4) The RNN output layer uses the XENT binary cross entropy loss function and a sigmoid activation function

Table 1 provides an overview of the classifier's layers

Table 1: LSTM classifier layers. The function *dim* yields the size of the corresponding vector.

layer	*dim*(in)	*dim*(out)
LSTM	300	32
RNN output layer	32	2

3.6 Bidirectional Long Short Term Memory Classifier (BiLSTM)

The Bidirectional LSTM (BiLSTM) classifier uses a similar setting to the LSTM described in the previous section, but is capable of learning patterns in and against the writing direction. The BiLSTM concatenates the weights of the forward and backward LSTM yielding an output vector of size 64 (Table 2). As for the LSTM an RNN output layer with a sigmoid activation function is used for summarizing the result.

[4]https://www.csie.ntu.edu.tw/~ cjlin/libsvm/

Table 2: BiSTM classifier layers. The function *dim* yields the size of the corresponding vector.

layer	dim(in)	dim(out)
BiLSTM	300	2×32
RNN output layer	64	2

3.7 Convolutional Neural Network

Convolutional Neural Networks (CNN) have also been shown to perform well for text classification tasks [22]. The experiments conducted within this research use a CNN classifier with the following hyperparameters:

- Optimizer: Adam optimizer (learning rate $\alpha = 1 \cdot 10^{-4}$)
- Weight initialization: ReLU
- Activation: LeakyReLU
- Convolution Mode: same[5], i.e. padding values are calculated automatically based on input size, kernel size and strides; the output size is determined as follows

$$outputHeight = ceil(inputHeight/strideHight)$$
$$outputWidht = ceil(inputWidth/strideWidth)$$

- L2 regularization (0.0001) and a dropout value of 0.5 to limit overfitting.
- The output layer uses the XENT binary cross entropy loss function and a sigmoid activation function.

Table 3 provides a summary of the layers used for the CNN classifier.

Table 3: Convolutional Neural Network layers. The function *dim* yields the size of the corresponding vector.

Layer	*dim*(in)	*dim*(out)	Kernel	Parameters
CNN1	300	150	1	stride(1,300)
CNN2	300	150	2	stride(1,300)
CNN3	300	150	3	stride(1,300)
Vertex				merge(CNN1, CNN2, CNN3)
GlobalPooling				max pooling
OutputLayer		2		XENT, sigmoid

4 EVALUATION

4.1 Evaluation Dataset

The evaluation dataset consists of manually curated articles covering corruption cases between 1995 and 2019 in Anglo-American news media. The dataset consists of two major parts:

(1) 800 articles on the topic of corruption that has been compiled by domain experts from the Swiss Federal Department of Foreign Affairs (FDFA).

[5]deeplearning4j.org/api/latest/org/deeplearning4j/nn/conf/ConvolutionMode.html

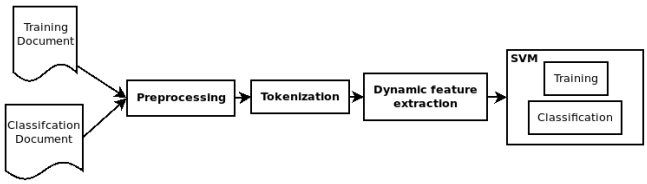

Figure 4: A simplified model of the Support Vector Machine (SVM) with Bag of Words features processing steps and components. The *Dynamic feature extraction* component removes words that appear less than five times from the vector space representation used for training and classification.

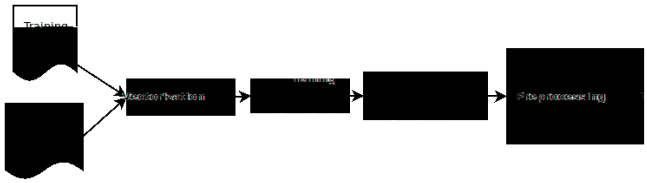

Figure 5: A simplified model of *Document vector SVM* processing steps and components. The *Vectorization* components translates documents into the corresponding embeding vectors used as features in the model.

(2) A second collection of articles that have been retrieved from U.S. and UK newspapers such as "The Guardian", "The Times", "The New York Times" and "The Wall Street Jorunal" based on query terms which domain experts deemed to be relevant to the topic of corruption. In a second step, these experts manually classified the retrieved articles in either "relevant" or "irrelevant" to the topic of corruption, creating a balanced gold standard dataset of 800 corruption-relevant and 800 corruption-irrelevant articles.

The selection of the dataset and particularly of the negative examples (i.e. corruption-irrelevant articles that still contained query terms that have been associated by domain-experts with the topic of corruption), make distinguishing between corruption and non-corruption articles a very challenging task. Nevertheless the chosen setting addresses the need for a high precision corruption classifier as outlined in Section 1.2.

4.2 Experiments

We have conducted a comprehensive number of experiments to (a) determine the optimal configurations and hyperparamters for each classifiers (see Section 3), and (b) to compare the classification performance between the developed models. These experiments draw upon the evaluation dataset introduced in the previous section.

Table 4 summarizes the evaluation results of the following classifiers:

(1) Naïve Bayes (Section 3.3) with Bag of Words features (NAIVE_BAYES).
(2) Support Vector Machines (Section 3.4) using Bag of Words (SVM_BOW), Word2Vec Embeddings (SVM_W2V) and FastText Embeddings (SVM_FST) features.
(3) Long Short Term Classifier (Section 3.5) with Word2Vec Embeddings (LSTM_W2V) and FastText Embeddings (LSTM_FST) features.
(4) Bidirectional Long Short Term Classifier (Section 3.6) with Word2Vec Embeddings (BiLSTM_W2V) and FastText Embeddings (BiLSTM_FST) features.
(5) Convolutional Neural Network (Section 3.7) with Word2Vec Embeddings (CNN_W2V) and FastText Embedding (CNN_FST) features.

Each result has been determined by averaging precision, recall and F1 of five subsequent evaluation runs that used ten-fold cross evaluations for training and testing the classifier.

Table 4: Summary of the classifiers' evaluation results.

Model	precision	recall	F1	Embedding
NAIVE_BAYES	0.6806	0.8804	0.7677	-
SVM_BoW	0.86	0.4623	0.6014	-
SVM_W2V	0.8554	0.7889	0.8208	Word2Vec
SVM_FST	0.8554	0.7634	0.8068	FastText
LSTM_W2V	0.7634	**0.9221**	0.8353	WordVvec
LSTM_FST	0.9324	0.7419	0.8263	FastText
BILSTM_W2V	**0.9615**	0.8065	0.8772	Word2Vec
BILSTM_FST	0.8065	0.8721	0.8381	FastText
CNN_W2V	0.8495	0.8977	0.8729	Word2Vec
CNN_FST	0.8495	0.9080	**0.8778**	FastText

The classifiers yield confidence values (i.e. corruption scores) that are proportional to their assessment of the article's likelihood of being relevant to the topic of corruption. For the evaluations, a cutoff value that optimizes the classifier's F1-score has been selected, to determine whether the article is considered to cover the topic of corruption or not. Only articles with a corruption score above the cutoff value are considered relevant to the topic of corruption. Table 5 provides an overview of these classifier-specific cutoff values.

Table 5: Classifier-specific cutoff values used to determine whether a document covers the topic of corruption.

Model	Cutoff
NAIVE_BAYES	0.90
SVM_BoW	0.90
SVM_W2V	0.60
SVM_FST	0.70
BILSTM_W2V	0.60
BILSTM_FST	0.60
LSTM_W2V	0.75
LSTM_FST	0.80
CNN_W2V	0.50
CNN_FST	0.50

For neural models, we also determined the optimal number of epochs (Table 6) for each classifier setting prior to the evaluation runs.

Table 6: Optimal epoch count for each neural model

Model	Epochs
LSTM_W2V	29
LSTM_FST	22
BILSTM_W2V	33
BILSTM_FST	11
CNN_W2V	26
CNN_FST	21

4.3 Discussion

As outlined in Section 4.1 the dataset used in the evaluations has been a particularly challenging:

- Corruption relevant media coverage has been overrepresented (1600 documents on corruption versus 800 documents that do not cover corruption) and
- even documents that do not focus on media coverage on corruption *do contain terms* relevant to this topic.

The experiments performed in Section 4.2 reflect these setting. Traditional bag of word approaches (NAIVE_BAYES and SVM_BoW) have serious difficulties in distinguishing articles on corruption from irrelevant content and clearly underperform when compared to models that draw upon domain-specific word embeddings. Introducing word embeddings considerably improves the model performance with F1 measures above 0.80% for all models. Although FastText enhanced upon Word2Vec embeddings by considering subword units (i.e. character n-grams) the performance differences between these two embedding types are not very pronounced for English media coverage. In the experiments, the SVM, LSTM and BiLSTM classifiers yield better results with Word2Vec embeddings. The CNN classifier, in contrast, performs better with FastText embeddings. Since FastText handles out-of-vocabulary words, we expect that models drawing upon FastText will perform better for languages that frequently use compound words such as German.

In terms of overall performance (F1), the CNN classifier with FastText embeddings produced the best performance, closely followed by the BiLSTM classifier with Word2Vec embeddings which also yields the highest precision. In terms of recall, the LSTM model with Word2Vec embeddings yielded the best results.

Since the classification algorithm within the Integrity Risk Monitor dashboard primarily serves the purpose

(1) of pre-selecting relevant document sets for identifying trends and aggregated metrics, and
(2) domain experts may include additional document at any time by lowering the required integrity score

a high precision is of particular importance. Therefore, the BILSTM_W2V classification algorithm has been used for computing the domain relevance of English documents.

5 OUTLOOK AND CONCLUSIONS

This paper described the data sets and evaluations performed for selecting the classifiers used for identifying English media coverage on corruption within the Integrity Risk Monitor dashboard. Traditionally, social scientists select corpora based on domain-specific query terms that identify relevant documents. The created gold standard dataset, therefore, not only contains (i) media articles on corruption that have been selected by domain experts, but also (ii) a corpus of articles matching typical query terms for corruption that have been classified into corruption and non-corruption coverage by domain experts. Identifying irrelevant articles within an input stream of documents that contain terms indicating corruption coverage is considerably more difficult than filtering completely unrelated content.

The conducted experiments provide the following insights: (i) self-trained word embeddings clearly outperformed pre-trained embeddings (Section 3.2), since they seemed to be better adapted to the application domain, and (ii) more sophisticated Deep Learning models such as BiLSTM and CNN yielded the highest F-measures obtaining values as high as 0.87 (Table 4). The evaluation demonstrate that even in this challenging setting, these model provided good results with a precision as high as 0.96 (BiLSTM model with Word2Vec embeddings) and 0.85 (CNN with FastText embeddings).

Future research will primarily focus on the following two areas:

(1) We will build upon the presented insights to further improve the corruption classifier and to perform more comprehensive evaluations that will also consider German documents.
(2) Creating gold standard data sets has been proven to be a very labor-intensive endeavour. We, therefore, plan to deploy a hybrid approach that (i) applies transfer learning techniques to the task of adapting the classifier to a wide range of integrity topics such as human rights, labor conditions and sustainability, and (ii) uses these classifiers for creating silver standard data sets that are then verified by domain experts.

Once these components are in place, domain experts will be able to draw upon domain relevance scores for integrity topics such as corruption, sustainability and labor conditions, for selecting and analyzing discussions, upcoming trends, and the public perception of events that are related to an organization's integrity risks.

Acknowledgement

The research presented in this paper has been conducted within the *Integrity Risk Monitor (IRM)* project funded by the KBA Notasys Integrity funds.

REFERENCES

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5, 135–146. DOI: 10.1162/tacl_a_00051. https://www.aclweb.org/anthology/Q17-1010.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, (October 2018). arXiv: 1810.04805. Retrieved 06/18/2019 from http://arxiv.org/abs/1810.04805.

[3] Nicholas Diakopoulos. 2020. Computational News Discovery: Towards Design Considerations for Editorial Orientation Algorithms in Journalism. en. *Digital Journalism*, (March 2020), 1–23. ISSN: 2167-0811, 2167-082X. DOI: 10.1080/21670811.2020.1736946. Retrieved 04/21/2020 from https://www.tandfonline.com/doi/full/10.1080/21670811.2020.1736946.

[4] Geoffrey E. Hinton, Sara Sabour, and Nicholas Frosst. 2018. Matrix capsules with EM routing. In *International Conference on Learning Representations*. https://openreview.net/forum?id=HJWLfGWRb.

[5] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *5th International Conference on Learning Representations (ICLR-17)*, abs/1609.02907. _eprint: 1609.02907. http://arxiv.org/abs/1609.02907.

[6] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *arXiv:1909.11942 [cs]*. arXiv: 1909.11942. Addis Abeba, Ethiopia, (April 2020). Retrieved 03/09/2020 from http://arxiv.org/abs/1909.11942.

[7] Adam Marcus, Michael S. Bernstein, Osama Badar, David R. Karger, Samuel Madden, and Robert C. Miller. 2011. Twitinfo: aggregating and visualizing microblogs for event exploration. en. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. ACM Press, Vancouver, BC, Canada, 227. ISBN: 978-1-4503-0228-9. DOI: 10.1145/1978942.1978975. Retrieved 04/21/2020 from http://dl.acm.org/citation.cfm?doid=1978942.1978975.

[8] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.

[9] Lyndon Nixon, Daniel Fischl, and Arno Scharl. 2019. Real-Time Story Detection and Video Retrieval from Social Media Streams. en. In *Video Verification in the Fake News Era*. Vasileios Mezaris, Lyndon Nixon, Symeon Papadopoulos, and Denis Teyssou, editors. Springer International Publishing, Cham, 17–52. ISBN: 978-3-030-26752-0. DOI: 10.1007/978-3-030-26752-0_2. Retrieved 04/21/2020 from https://doi.org/10.1007/978-3-030-26752-0_2.

[10] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, (October 2014), 1532–1543. DOI: 10.3115/v1/D14-1162. Retrieved 12/12/2019 from https://www.aclweb.org/anthology/D14-1162.

[11] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. en-us. In (June 2018), 2227–2237. DOI: 10.18653/v1/N18-1202. Retrieved 06/18/2019 from https://aclweb.org/anthology/papers/N/N18/N18-1202/.

[12] Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and S. S. Iyengar. 2018. A Survey on Deep Learning: Algorithms, Techniques, and Applications. *ACM Comput. Surv.*, 51, 5, (September 2018), 92:1–92:36. ISSN: 0360-0300. DOI: 10.1145/3234150. Retrieved 02/25/2019 from http://doi.acm.org/10.1145/3234150.

[13] Suhas Ranganath, Xia Hu, Jiliang Tang, and Huan Liu. 2016. Understanding and Identifying Advocates for Political Campaigns on Social Media. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining* (WSDM '16). ACM, New York, NY, USA, 43–52. ISBN: 978-1-4503-3716-8. DOI: 10.1145/2835776.2835807. Retrieved 02/25/2016 from http://doi.acm.org/10.1145/2835776.2835807.

[14] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic Routing Between Capsules. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, 3856–3866. http://papers.nips.cc/paper/6975-dynamic-routing-between-capsules.

[15] Arno Scharl, David Herring, Walter Rafelsberger, Alexander Hubmann-Haidvogel, Ruslan Kamolov, Daniel Fischl, Michael Föls, and Albert Weichselbraun. 2017. Semantic Systems and Visual Tools to Support Environmental Communication. *IEEE Systems Journal*, 11, 2, 762–771. DOI: 10.1109/JSYST.2015.2466439.

[16] Arno Scharl, Alexander Hubmann-Haidvogel, Alistair Jones, Daniel Fischl, Ruslan Kamolov, Albert Weichselbraun, and Walter Rafelsberger. 2016. Analyzing the Public Discourse on Works of Fiction - Automatic Emotion Detection in Online Media Coverage about HBO's Game of Thrones. *Information Processing & Management*, 52, 1, 129–138. ISSN: 0306-4573.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances*

in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA. Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, 5998–6008. http://papers.nips.cc/paper/7181-attention-is-all-you-need.

[18] Albert Weichselbraun, Stefan Gindl, Fabian Fischer, Svetlana Vakulenko, and Arno Scharl. 2017. Aspect-Based Extraction and Analysis of Affective Knowledge from Social Media Streams. *IEEE Intelligent Systems*, 32, 3, (May 2017), 80–88. ISSN: 1541-1672. DOI: 10.1109/MIS.2017.57.

[19] Albert Weichselbraun and Philipp Kuntschik. 2017. Mitigating linked data quality issues in knowledge-intense information extraction methods. In *7th ACM International Conference on Web Intelligence, Mining and Semantics (WIMS 2017)*. Amantea, Italy, (June 2017).

[20] Albert Weichselbraun, Arno Scharl, and Stefan Gindl. 2016. Extracting Opinion Targets from Environmental Web Coverage and Social Media Streams. In *Proceedings of the 49th Hawaii International Conference on System Sciences (HICSS-49)*. Accepted 17 August 2015. IEEE Computer Society Press, Kauai, Hawaii, (January 2016).

[21] Albert Weichselbraun and Norman Süsstrunk. 2015. Optimizing Dependency Parsing Throughput. In *Proceedings of the 7th International Conference on Knowledge Discovery and Information Retrieval (KDIR 2015)*. Accepted 21 September 2015. Lisbon, Portugal.

[22] T. Young, D. Hazarika, S. Poria, and E. Cambria. 2018. Recent Trends in Deep Learning Based Natural Language Processing [Review Article]. *IEEE Computational Intelligence Magazine*, 13, 3, (August 2018), 55–75. ISSN: 1556-603X. DOI: 10.1109/MCI.2018.2840738.

[23] X. P. S. Zhang and D. Kedmey. 2018. A Budding Romance: Finance and AI. *IEEE MultiMedia*, 25, 4, (October 2018), 79–83. DOI: 10.1109/MMUL.2018.2875858.

[24] Wei Zhao, Haiyun Peng, Steffen Eger, Erik Cambria, and Min Yang. 2019. Towards Scalable and Reliable Capsule Networks for Challenging NLP Applications. en, (June 2019). Retrieved 06/18/2019 from http://128.84.21.203/abs/1906.02829.