SMU Data Science Review

Volume 2 | Number 3 Article 10

2019

Increasing Robustness in Long Text Classifications Using Background Corpus Knowledge for Token Selection.

Clovis R. Bass
Southern Methodist University, crbass@smu.edu

Brett Benefield
Southern Methodist University, sbenefield@smu.edu

Debbie Horn
IBM, dnh@us.ibm.com

Rebecca Morones
IBM, rlnadeau@us.ibm.com

Increasing Robustness in Long Text Classifications Using Background Corpus Knowledge for Token Selection.

Clovis R. Bass[1], Brett Benefield[1,2], Debbie Horn[2], and Rebecca Morones[2]

[1] Master of Science in Data Science, Southern Methodist University, Dallas TX 75275 USA {crbass,sbenefield}@smu.edu
[2] IBM, 3039 E Cornwallis Rd Research Triangle NC 27709 USA {bbenefield,dnh,rlnadeau}@us.ibm.com

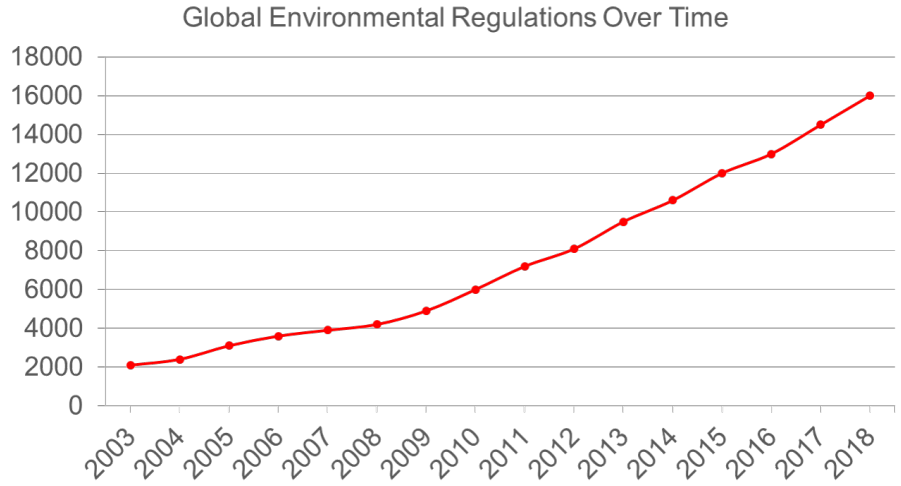
Abstract. In this paper we present a system of long text classification for environmental regulations that emphasizes background Subject Matter Expert (SME) knowledge for token selection to increase its robustness and accuracy. Here we characterize anything as long text if it exceeds more than twice the normal token limit for neural networks (512). Over one thousand new environmental regulations are released each year and our system helps SMEs prioritize post-classified regulations and highlights their important aspects via Natural Language Processing (NLP). We have utilized environmental regulation SME expertise through a business specific keyword dictionary to target tokens to input into NLP neural networks. We sought to optimize this method as it doesn't rely on fallacies in common approaches like inputting the beginning tokens of a document where formatting differences can bias a model, or segmenting the entire document that muddles a model's ability to find important features as well as increase computational time drastically. Many NLP neural networks were tested along with different token pre-processing to find the optimal combination for this unique environmental regulation corpus. We found our best results were with the BERT neural network with it's extracted tokens lemmatized but keeping stop words with a recall of 93.33% and accuracy of 81.67%. We conclude that this mostly transparent method can yield highly accurate classifications and can easily be translated to any field of expertise, given enough background knowledge to build a proper keyword dictionary.

1 Introduction

Global environmental regulations have been growing at a linear rate the last several years, increasing by over one thousand documents per year (Figure 1). Companies must read and interpret these new regulations to ensure compliance and not subject themselves to potentially large financial penalties, along with the potential irreversible harm done to the environment. Also, very few of these regulations will be relevant to a given company. A Subject Matter Expert (SME) in environmental regulations, as they are the most knowledgeable and capable

of accurately assessing each document, must read each regulation to determine whether it is relevant to a given company. If it is found to be relevant then the SME must determine what actions must be taken as a result and apply it to their company's products and procedures. In our company's field of expertise, regulations on shipping, storing, and creating electronic goods are the topics of relevance for our SME. This umbrella of products then has under it anything relevant to the chemicals that are used to make these products or their byproducts. For example with electronic goods, regulations for the disposal and shipment of lithium (a vital component of lithium batteries) are highly relevant, whereas the shipment and/or disposal of medical waste is not. Many of these regulations are easily dismissed based on their title alone, but discerning from the subset that are closely related to our relevant documents is not trivial outside of reading the majority of a regulation before determining its relevance by a SME.

Fig. 1. Growth of Regulatory Documents over Time [9]



A regulatory document is composed of many sections, but they are not ubiquitous across different regulatory agencies. For classification purposes, we were concerned with what section described what was being regulated. However, many of these sections describe how and where these regulations are to be implemented. Many times what is being regulated is described thoroughly in an introduction section, but scraping only a proper section from numerous different regulatory agencies proves complicated and inaccurate quickly. This is due to the different style and formatting used by the many regulatory agencies. So following conventional methods of inputting only the beginning of a document leaves us at the mercy that a regulation has important features needed to classify a model in

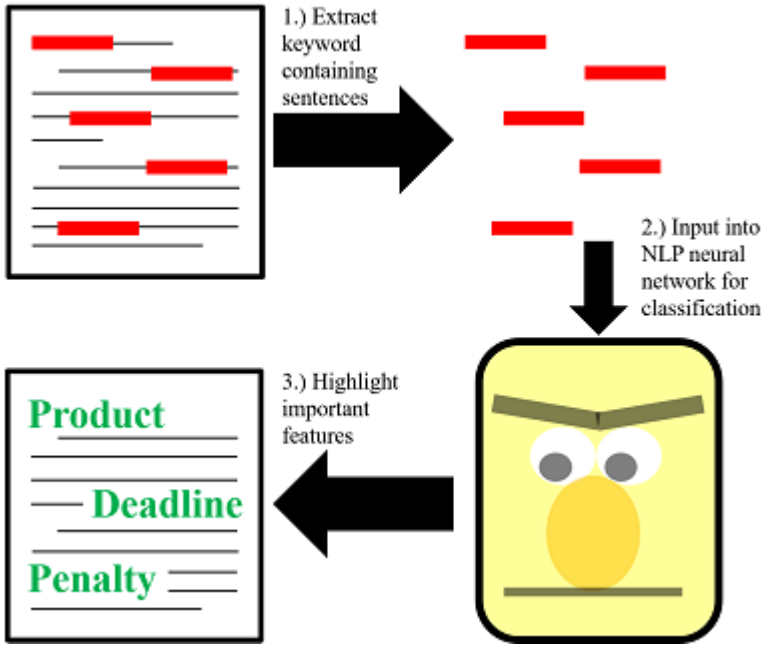
this portion of the document. Our other option is to input the entire document into a model via token or sentence segmentation, but this requires much more computational time and would likely require an enormous corpus to find important features hidden in these large text's. Instead, we found it more reasonable to use a keyword dictionary that lists a company's product types that are commonly found in the regulatory documents. This would make our model more robust and less reliant on a document's formatting, but only on the way they describe product types which should be well known by a SME. These keywords were used to find the sentences that contained the keyword, then this sentence and the flanking sentences were collected to input into a neural network for classification. The flanking sentences were collected to provide further context of where the keyword was found since some regulations describe what is not being regulated rather than only what is. Outside of providing a classification result via a probability of relevance (the percent confidence the algorithm predicted the regulation as relevant), stakeholders are always needing some important aspects that are available in every regulation. We addressed this in our pipeline (Figure 2) post-classification by highlighting effective dates of the regulation, the amount a penalty is for non-compliance, and the product keyword itself.

Neural networks in Natural Language Processing (NLP) have quickly turned into an arms race among some of the most powerful technology companies. A significant improvement in training times and several NLP metrics like General Language Understanding Evaluation (GLUE, a widely used NLP benchmark) have been observed recently with Google announcing and releasing of Bidirectional Encoder Representations from Transformers (BERT) in late 2018. BERT almost instantly became the NLP neural network of choice for most complex NLP problems. The original BERT model was pre-trained on the entire English Wikipedia corpus (2,500M words) and the BooksCorpus (800M words) [1]. With such a broad spectrum of topics it was pre-trained on and its architecture leading it to provide state-of-the-art results on NLP problems, this lent BERT to be a great option for many NLP situations. BERT now has many different iterations in attempt to improve it in all situations or to be pre-trained on more specific corpus' to improve its capabilities on a niche problem. RoBERTa is Facebook's version to optimize BERT by modifying key hyperparameters to improve on downstream task performance and was pre-trained on 10 times the data BERT was. Facebook then went on to create Cross-lingual Language Model (XLM) to build upon BERT by using a different pre-processing technique yet again and a dual-language training mechanism. An even newer Transformer model, XLNet, has recently been released by Google Brain that improves upon BERT in 20 different NLP situations [12].

As shown in Howe et al., unique corpus' like a legal documents prove BERT underwhelming and less accurate than fine tuning traditional statistical models [2]. They infer that while BERT was outperformed, it was likely that a BERT model that had been pre-trained on a more niche corpus like pure legal cases could result in a better model performance. There are not many clear scenarios where one neural network will outperform all others in a specific problem. A

common approach is to compare several models to find the best for a particular problem. However, if a corpus is large enough the best option is to pre-train a neural network on the identified corpus. This allows for jargon that will be unique to a certain corpus, such as legal terms and chemical names in our instance, to be more clearly contextualized by the neural network since it will be much more prevalent in its pre-training. This pre-training is computationally intensive and requires lots of computational time, and consequently money, to carry this out. Without access to these resources, we focused on finding the best open source pre-trained neural network to classify our documents.

Fig. 2. Project Pipeline



An inherent limitation of these neural networks is how many tokens that can be input into the model to classify each document. The token amount of each document will almost always far exceed this limit of 512 tokens. We describe "long text" in our situation as documents that more than double this token limit even though a majority of our documents far exceed this. A choice had to be made on what tokens to input into the model to help it classify more accurately. Rather than finding the most predictive sections that might different from agency to agency, token selection was done in an exhaustive iterative process with company SMEs to find keywords that are prevalent in these regulations. However, finding these alone aren't useful in certain situations when an environmental regulation defines its scope. It could describe that products

important to a company are actually not in the scope of the environmental regulation listed. So context around each keyword is necessary to help the neural network decipher between relevancy and not. Other options are available to use the entire document in a segmented fashion as many inputs for one document but we found the keyword option more efficient.

With all these combined aspects we found that even neural networks that are not pre-trained on a niche corpus can yield impressive classification results with large documents. With a recall of 93.33% and accuracy of 81.67% on a base BERT model we found our method viable and potentially translatable to any business with the knowledge to build a keyword dictionary for their product's of interest.

2 Methodology

2.1 Document Collection

The initial environmental regulation documents that were found relevant to the company were stored on a secure cloud document management platform. Irrelevant files were then collected from a regulatory compliance management platform.

2.2 Document Transformation

Environmental regulations came in three forms. Most common was Microsoft Word files where all formatting had to be stripped to be converted into plain text form. Next was Portable Document Format (PDF) files that were actual text that could be converted to plain text directly. We used Adobe Acrobat Pro's export feature to convert these documents to plain text documents as additional formatting code has no value in our model. Lastly some PDF files were only scanned images of a document. The scanned image PDFs had to be converted to text through character optical recognition (OCR) techniques already implemented in Adobe Acrobat.

2.3 Document Analysis

Environmental regulation documents complicate our corpus with both legal jargon and chemical names. There are no tokenizers that tokenize both instances well simultaneously. Chemical names, particularly ones using IUPAC nomenclature, will almost never be tokenized correctly or recognized as a named entity unless using a chemical specific tokenizer. IUPAC nomenclature uses numbers, special characters, and spaces between words in a chemical name (Table 1), and tokenizing this correctly has been largely remedied by tokenizers like ChemDataExtractor [4]. We chose to use ChemDataExtractor because recognizing chemical tokens more readily than legal jargon proved more useful in helping with classification. Also downstream applications needing chemical information outside the scope of the project would find this useful.

Table 1. IUPAC Chemical Substance Names

1,1,2-Trichloroethane
2,2'-Methylenediphenyl diisocyante
3-ethyl-3-methylpentane
1,5-Heptadien-4-one,3,3,6-trimethyl
N-ethyl-N-methylpropanamine

2.4 Token Selection

Descriptors of general product types were created as a keyword dictionary that could be used to search for each product type in each document. These were generated by a SME with a high level of background knowledge of the descriptors used in the regulations that refer to the product types of importance of their company. To allow this to scale we utilized a package called FlashText because it outperforms RegEx drastically as the amount of keywords to search for increases past several hundred [16]. FlashText finds pre-defined keywords quickly, but at the expense that the keyword must exactly match what is given in its dictionary. We used this to find sentences that contained keywords and their flanking sentences. Collecting the flanking sentence along with the main sentence that contains the keyword allows us to gain any context that might be important surrounding the keyword, such as the scope might describe that keyword is not affected by this regulation. This process was continued until the token limit for the neural network was reached. If a keyword was not found, the tokens taken to be input were split evenly between the beginning and the end of the document. This is because a common format for regulations describe the scope of the regulation in the introduction section and have a definition section at the end of the document.

2.5 Corpus Pre-Processing

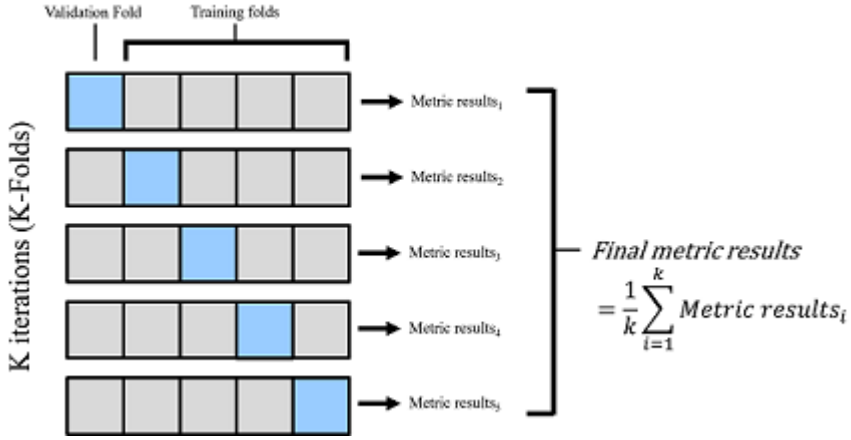
In NLP, most, if not all, corpus must be pre-processed before utilizing even basic techniques. Each document was parsed into individual words, called tokens. Appropriate pre-processing was done to eliminate noise in the documents, such as stop words. Stop words are common words in any language that do not provide context to overall topics in a given phrase. Stop words are article adjectives that include 'a', 'an', 'in', 'the', etc. Other common legal jargon were added as stop words so our system could focus on more infrequent terms to provide better results. Further corpus normalization included: converting all text to lowercase; white space and punctuation removal; expanded contractions; lemmatization of words (ex. running lemmatizes to run); and any misspellings were corrected.

2.6 Model Computation

Our model was trained and tested on 200 documents. With an initial small dataset we used 8-fold cross-validation. This was the best choice to find a robust

model without having a large corpus currently at our disposal. Cross validation "holds out" a fraction of data to be tested on while the remaining fraction of data is trained on (Figure 3).

Fig. 3. K-Fold cross validation



2.7 Highlighting document

After a classification was made, we utilized the Adobe Acrobat API in Python to highlight important features present in all PDF regulations. Microsoft Word documents were highlighted via the Python-docx library ([17]). First, we highlighted the product type that was provided via our keyword dictionary. Next, effective dates for the regulation were highlighted, and lastly the penalty amount for non-compliance. The combination of these three answer common questions stakeholders need answered for each regulation.

2.8 Programming language

Our solution was primarily coded in the Python programming language. Python is a functional programming language that is commonly used in academia and businesses to solve problems due to its ease of use and terse coding style. This allows programmers to focus on the solution and less on the technicalities of the language. In addition, it has garnered large support from the open source community in the form of many libraries released for public use. Pytorch is valuable library that was developed by Facebook's artificial intelligence research group [13] that has many of the neural networks used in our analysis. Pytorch is specifically designed around the use of tensors which can be thought of as multidimensional arrays that can take advantage of the computational speed and parallelization of tasks on a Graphical Processing Unit (GPU). The parrallelization of tasks allows a GPU to out-compute the CPU by 10 - 50x depending

on the hardware being compared (Table 2). An advantage of the use of tensors is the bookkeeping done within the code to calculate its own gradient. The computation of gradients is instrumental in the construction of neural networks as this allows the weights of neurons to be updated accordingly to minimize the loss function. The gradient represents the rate of change and identifies the step in which direction the weights should be updated as information is back propagated through the network. The contents of the tensor are updated and the contents are nothing more than weight values as seen here [[-0.16702934 0.07173464 -0.04512421] [-0.02265321 0.06509651 -0.01419079]]. This is an oversimplified example where the input is limited to 2 inputs and 3 outputs with no hidden layers.

Table 2. CPU vs. GPU Computation Time [14]

Processing Unit	Forward Propagation (ms)	Backward Propagation (ms)	Total (ms)
CPU: Dual Xeon E5-2630 v3	847.46	1348.33	2195.78
GPU: Maxwell Titan X	55.82	96.01	151.82
GPU: GTX 1080	42.94	79.17	122.10
GPU: GTX 1080 Ti	50.01	65.99	116.03
GPU: Pascal Titan X	34.76	61.64	96.40

3 Data Set

Our dataset consisted of 200 environmental regulations. 100 of which were deemed relevant to our company's products or processes and were previously stored in a cloud document management platform. The remaining 100 irrelevant regulations were collected from a regulatory compliance management platform. A typical regulatory document consists of sections that defines the entity being regulated, the governing rules for that entity, when the regulation goes into effect, and penalties associated with non-compliance. An excerpt from a regulation in California regulating e-waste is provided as an example text [7]. We also limited all regulations we processed to those released within the United States and European Union across different levels of environmental agencies and written in English.

> "Existing law, the Electronic Waste Recycling Act of 2003, requires a retailer selling a covered electronic device in this state to collect a covered electronic waste recycling fee from the consumer, as specified. Under existing law, the fees are deposited in the Electronic Waste Recovery and Recycling Account, and the California Integrated Waste Management

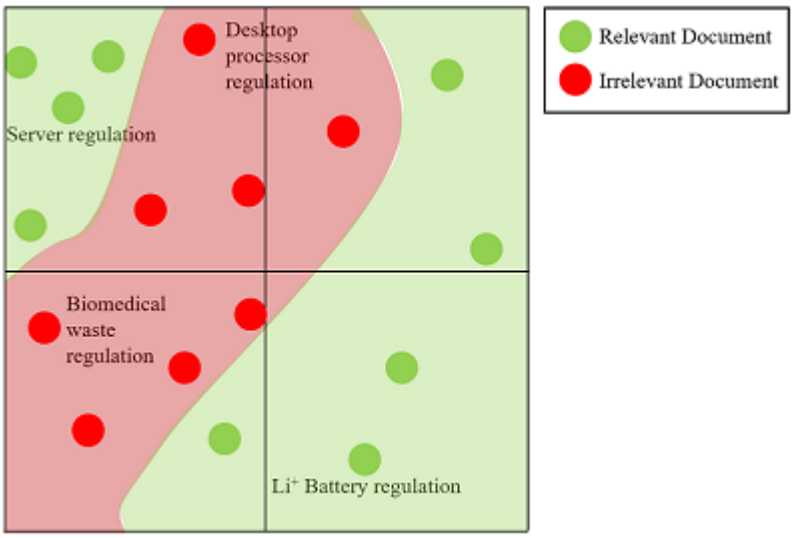
> Board and the Department of Toxic Substances Control are continuously appropriated the money in the account to make electronic waste recovery payments and recycling payments to cover the net cost of an authorized collector in operating a free and convenient system for collecting, consolidating, and transporting covered electronic wastes, and to make electronic waste recycling payments to cover an e-waste recycler's net cost of receiving, processing, and recycling covered electronic waste."

Environmental regulations are also commonly not published as full documents, amendments are often published that only call reference to the original full regulation and where it should be amended. These amendment documents are on average much smaller than their entire regulation that it is amending.

4 Methods and Experiments

In binary text classification algorithms are trained to identify a boundaries that adequately separate the documents into two separate groups. In conventional machine learning algorithms like Support Vector Machines it does this by finding a hyperplane in n-dimensional space to separate the different groups. A hyperplane is a mathematical concept that allows something that exists in hyperspace (greater than 3 dimensions) to be divided within that space. However, in neural networks they are composed of many layers of simple algorithms to combine to form a non-linear classifier. An oversimplified version on how the documents could be classified can be seen in Figure 4. The image shows how the algorithm attempts to find the optimum non-linear equation such that there is maximum separation between the relevant and irrelevant documents.

Fig. 4. Non-Linear Classification Example



We utilized several state-of-the-art NLP neural networks in a heuristic approach in attempt to find the best one for our corpus. There is no clear choice in what NLP neural network will be best suited for a particular problem so we decided to employ several of the most highly touted ones currently in popular use. BERT was a cutting edge neural network that has dominated the NLP problem solving field and only now is being surpassed by other neural networks being released in the past year. BERT was able to do this by applying non-directional (all at once) training to Transformer neural networks, even though the naming convetion indicates it is bidirectional. Google showed that training this way gave the algorithm much more context on what words could be predicted if words were masked rather than reading it in only one or both directions. Facebook built upon BERT with their version RoBERTa by pre-training it on ten times the data BERT was (161gb to 16gb). This increased the subword count from 30,000 to 50,000. Also its pattern of masking words to predict them with the context that surrounds it differs by constantly changing this pattern (dynamic vs static). It does this by training on its masked data ten times with different masking patterns, while BERT only does this once. These improvements help RoBERTa outperform its predecessor BERT in many common NLP metrics. Facebook then went on to great a new Transformer, Cross-lingual Language Model (XLM). While this Transformer was created to help mainly with cross-lingual classifications, we wanted to see its promise in an only English corpus initially before the inevitable expansion of our corpus in future work into other regulations that are not published in English. XLNet is the latest Google Brain NLP Transformer neural network released, outperforming BERT in many NLP

tasks. Rather than gauging how well it can predict words when they are masked given their context like in BERT or all the different masking combinations like in RoBERTa, XLNet trains on various permutations of the tokens in a sentence and predicts tokens in order of their sequence presented in the permutation. BERT, RoBERTa, XLM, and XLNet are some of the most powerful and popular NLP neural networks in use right now that we thought would find a near optimal classification result.

We also wanted to explore which amount of pre-processing would generate the best model. After tokenizing the document and extracting the sentence containing a keyword along with its flanking sentence, we either input these tokens as is, lemmatized these tokens, or lemmatize and remove stop words.

Determining the most appropriate error metric is vital in all supervised learning models. Predicting an environmental regulation as irrelevant when it is relevant (false negative) carries a drastic risk of non-compliance and fines because of a SME not reading it. Therefore, our priority was to reduce false negatives as much as possible. Doing so causes our model to classify more actual irrelevant documents as relevant for a SME to read, but this is the best option for lowering risk for the company. The metric that encompasses this is recall, which penalizes false negatives at the expense of not penalizing false positives at all. To provide more context on how the model is performing, we also chose to look at accuracy (ratio of correct predictions to the total amount of regulations) to ensure our model is not just predicting all regulations as relevant.

$$Recall = \frac{TruePositives}{(TruePositives + FalseNegatives)} \tag{1}$$

5 Results

Fig. 5. Neural network recall per input type

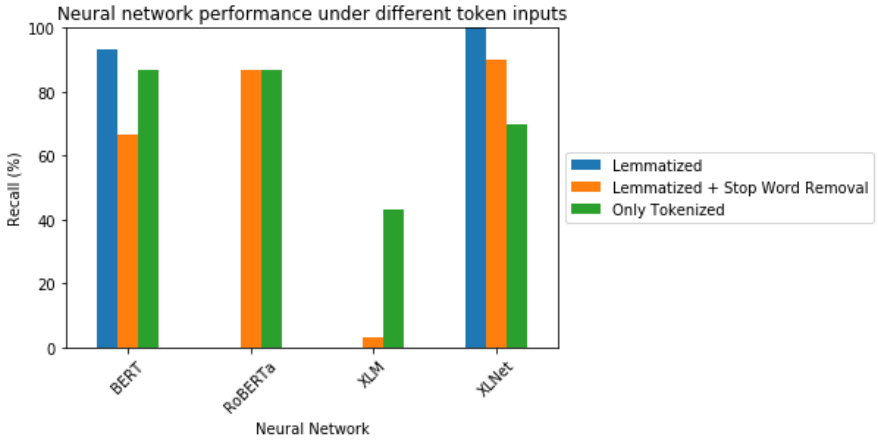
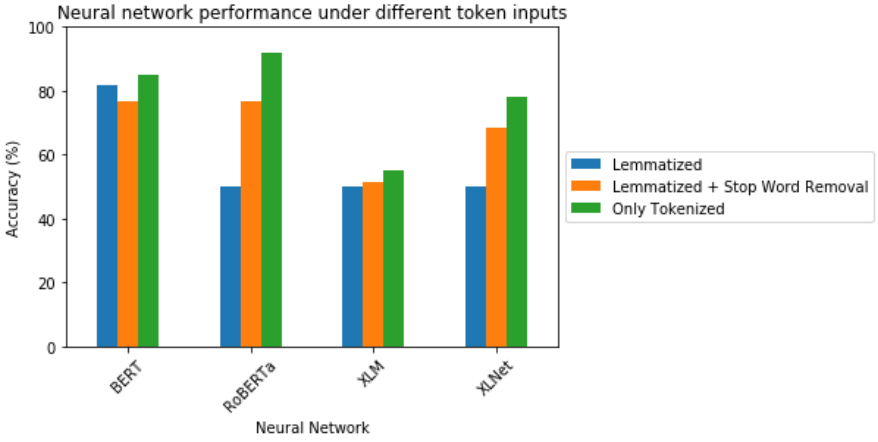


Fig. 6. Neural network accuracy per input type



With four different neural networks and using three similar but differently processed inputs we found BERT as our best performing neural network having the highest recall of 93.33% (without only predicting all regulations as relevant as XLNet with lemmatized input did) with its tokens only being lemmatized. RoBERTa had a higher accuracy (92%), but had a lower recall (87%) which we deemed as our most important metric. Our second highest recall of 90% came from XLNet when its tokens were lemmatized and stop words removed but had a relatively lower accuracy of 68.33%. XLM was our least impressive neural network tested with its best results being only slightly better than chance at 55% accuracy and a recall of 43.33% when its token inputs were not further pre-processed.

Table 3. BERT Performance

Data Input	Recall (%)	Accuracy (%)
Only Tokenized	86.70	85.00
Lemmatized	93.33	81.67
Lemmatized + Stop Word Removed	66.67	76.67

As noted, BERT was our best performing neural network while using lemmatized tokens from the extracted sentences. Even without lemmatization and/or stop word removal BERT generated the 4th highest recall (86.7%) while having the second highest accuracy (85%) of all neural networks and token input combinations tested.

RoBERTa was our second third performing neural network with a recall of 87% while the highest accuracy of all with 92% using tokens that were not lemmatized or had their stop words removed. However, it performed the worst out of

Table 4. RoBERTa Performance

Data Input	Recall (%)	Accuracy (%)
Only Tokenized	87.00	92.00
Lemmatized	0.00	50.00
Lemmatized + Stop Word Removed	86.67	76.67

all models tested when the tokens were lemmatized. It predicted all regulations as irrelevant causing it to have a recall of 0% and accuracy of 50%. Next, with its removal of stop words prior to lemmatization, its performance rebounded to the 5th highest recall of 86.67% with an accuracy of 76.67%.

Table 5. XLNet Performance

Data Input	Recall (%)	Accuracy (%)
Only Tokenized	70.00	78.00
Lemmatized	100.00	50.00
Lemmatized + Stop Word Removed	90.00	68.33

XLNet with its input lemmatized and stop words removed was our second highest performing model in recall with 90%, but a relatively lower accuracy of 68.33% when compared to the other higher performing models. It's performance without lemmatization or stop word removal gave less promising results with a recall of 70% and a accuracy of 78%. Opposite of RoBERTa's lemmatized model, XLNet's lemmatized model predicted all documents as relevant so generated a perfect yet misleading 100% recall with an accuracy of 50%.

Table 6. XLM Performance

Data Input	Recall (%)	Accuracy (%)
Only Tokenized	43.33	55.00
Lemmatized	0.00	50.00
Lemmatized + Stop Word Removed	3.33	51.67

XLM performed the worst out of our neural networks with only reaching a recall of 43.33% when its tokens were not lemmatized or had stop words removed while only having an accuracy of 55%. This was followed by worse performances when the tokens were lemmatized, generating only irrelevant predictions giving it a recall of 0% and accuracy of 50%. Lastly, when the tokens were lemmatized and stop words removed a recall of only 3.33% reached with an accuracy of 51.67%.

Our second step of this project was to highlight important features within the document. The highlighting of the document allowed the SME to focus on key

words which when highlighted stood out from the rest of the text. Another benefit of the annotation process was so these documents could be shared with other individuals who do not have expertise in environmental regulatory documents and easily identify the key aspects as well.

6 Analysis

Our results seem to show slight trends that these neural networks seem to prefer tokens that are not processed further by lemmatization or stop word removal. We believe this is the case because of how these neural networks were trained to include all context around a word in a sentence. So when we lemmatize words that their lemma's are either not in the subword dictionary of the neural network pre-training or mis-contextualize a sentence it might muddle our model further in classification. This thought process might hold true for stop word removal as well. Once stop words are removed the context around words in a sentence begin to blur. However, two of the better results came from when these further preprocessing steps were applied. This includes our top two highest recall scores of 93.33% and 90% by BERT with lemmatized tokens and XLNet with lemmatized and stop words removed from its tokens, respectively. Our clearest take away for the negative though is how XLM performed the poorest out of all neural networks tested, only achieving a recall of 43.33% on its best model.

7 Ethics

The most ethical dilemma with our solution is the potential for misclassification or pure over reliance without updating the model. Misclassification would involve the scenario where the model predicts an environmental regulation is not relevant to a company when it actually is (a false negative). This poses a large financial risk to the company itself and potential irreversible harm to customers or the environment as well. Constant updating of the model would need to ensue. This would include updating it if new products or chemicals become relevant to the company. A common occurrence are new findings in research as they unveil detrimental characteristics not known previously. If this chemical is utilized in a product, updating the model by including this chemical name to be used as input would be pivotal.

With most machine learning solutions, an ethical dilemma is replacing large amounts of jobs. However, our solution will likely not be used as a replacement for an environmental regulatory SME, rather it must always have a "human in the loop." It may however help limit the growing need for these positions. There will be inevitable new regulations that are dissimilar enough from our training set environmental regulations. It is foreseeable that new wording of current product types or entirely new regulations on previously unregulated products/chemicals would be easily misclassified by a model trained on all past regulations with an out of date keyword dictionary. This bias will have to be constantly noted by SMEs as new regulations are released to improve upon the model in attempt

to keep misclassifications to a minimum. This can be done by adding these new descriptors to the keyword dictionary and retraining the model so future instances will be classified correctly. Also an environmental regulation SME will still have to read the final paper to correctly understand and implement changes in order to be under compliance. Each of these steps show that while our solution might prevent the growing need for environmental regulation SMEs, the need for them to is vital at every step of the solution. This starts at the beginning of the solution in essentially creating the model by building the keyword dictionary, to its fine tuning by diagnosing causes for misclassifications, and of course with their final readings of the regulations as they did previously.

8 Conclusions

Our implementation of a keyword dictionary to funnel the amount of tokens as input into a neural network from potentially several thousand through segmentation or an arbitrary collection via at the beginning of a document proved promising, yielding recall percentages in the low 90's and sometimes accuracy's nearly as high. As noted before, this problem needs to be trained on what is being regulated while a majority of the document describes how something is to be regulated. So disregarding all information that doesn't pertain to describing what is being regulated narrowed the focus of what tokens our model could train on and make predictions easier. We sought to show this was a viable method for long text classification models with a corpus that might not provide key features needed for classification in a consistent placement across different document's, and we believe this remedies this situation given an appropriate background knowledge of the corpus. We believe with an increase in corpus size that our results will improve even further due to more clear examples being classified as well as the fringe topics being parsed as their amount in a training set increases.

Although we report a rather high recall for a classification problem (93.33%) with BERT using lemmatized keyword extracted sentences, our solution is in a scenario where false negatives are not acceptable. So although our results are promising and yielded very accurate classification, it is currently not viable to be relied upon fully. However, it does provide some value in it's classification by prioritizing properly many regulations that it deemed relevant. This shows that most of the value will come from the supplemental aspects provided in our solution, such as highlighting import aspects of the document for quick reading for SMEs or stakeholders. Further advancements in the solution will include transparency in the classification so we can tune the model by examining documents post-classification to find more important keywords to include in the keyword dictionary.

Also while we believed chemical names would have been productive in increasing classification accuracy, we found it slightly detrimental to our model. We believe this was the case due to it filling the token limit with erroneous sentences that did not help with classifications. Identifying chemicals in the reg-

ulations were still helpful for SME knowledge though so highlighting these was still implemented.

Formatting in environmental regulations is still a difficult issue and requires a much more intricate approach. Many regulations are published as only the amended portion of a document rather than republishing the entire regulation with the new information amended in. This was an issue with increasing our accuracy due to an amendment not having important keywords that would have been found outside of the amended section in the original document. Future workarounds to this might require managing a form of document reconstruction prior to classification.

References

1. Devlin, Jacob, and Ming-Wei Chang. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." 2018.
2. Howe, Jerrold, and Lim Khang. "Legal Area Classification: A Comparative Study of Text Classifiers on Singapore Supreme Court Judgments." 2019.
3. Lee, Jinhyuk, and Wonjin Yoon. "BioBERT: a Pre-Trained Biomedical Language Representation Model for Biomedical Text Mining." 2019.
4. Swain, M. C., & Cole, J. M. "ChemDataExtractor: A Toolkit for Automated Extraction of Chemical Information from the Scientific Literature", J. Chem. Inf. Model. 2016, 56 (10), pp 1894–1904 10.1021/acs.jcim.6b00207
5. Bird, Steven. Natural Language Processing with Python. 1st ed., O'Reilly Media, 2016.
6. Sarkar, Dipanjan. "Text Analytics with Python: a Practical Real-World Approach to Gaining Actionable Insights from Your Data." 1st ed., APRESS, 2019.
7. California AB-575 2005 07 18 Electronic Waste Recycling
8. The Institute for Ethical AI & Machine Learning. "Principles". Https://ethical.institute/principles.html
9. Compliance & Risks. "Deal With Your Regulatory Avalanche". https://www.complianceandrisks.com/solutions/conquering-the-regulatory-avalanche/
10. Trivedi, Kaushal. "Introducing FastBert - A Simple Deep Learning Library for BERT Models." Medium, HuggingFace, 17 May 2019, http://medium.com/huggingface/introducing-fastbert-a-simple-deep-learning-library-for-bert-models-89ff763ad384.
11. "RoBERTa: An Optimized Method for Pretraining Self-Supervised NLP Systems." Facebook AI Blog, Facebook, 29 July 2019, ai.facebook.com/blog/roberta-an-optimized-method-for-pretraining-self-supervised-nlp-systems/.
12. Yang, Zhilin. "XLNet: Generalized Autoregressive Pretraining for Language Understanding." 2019, arxiv.org/abs/1906.08237.
13. "PyTorch." PyTorch, https://pytorch.org/.
14. Johnson, Justin. "Benchmarks for Popular CNN Models." GitHub, 25 Sept. 2017, https://github.com/jcjohnson/cnn-benchmarks.
15. Chang, Wei-Cheng, et al. "X-BERT: eXtreme Multi-label Text Classification with BERT" https://arxiv.org/pdf/1905.02331.pdf
16. Singh, Vikash. "Regex Was Taking 5 Days to Run. So I Built a Tool That Did It in 15 Minutes." FreeCodeCamp.org, FreeCodeCamp.org, 15 Sept. 2017,

https://www.freecodecamp.org/news/regex-was-taking-5-days-flashtext-does-it-in-15-minutes-55f04411025f/.

17. Canny, Steve. “Create and Modify Word Documents with Python.” GitHub, 2013, https://github.com/python-openxml/python-docx.