



UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D.C. 20555-0001

December 10, 2024

MEMORANDUM TO: Kimberly Webber, Director
Division of Systems Analysis
Office of Nuclear Regulatory Research

FROM: Jennie Rankin, Acting Branch Chief **/RA/**
Division of Systems Analysis
Office of Nuclear Regulatory Research

SUBJECT: APPLICATION OF NATURAL LANGUAGE PROCESSING TO
NRC REGULATORY DOCUMENTS FUTURE FOCUSED
RESEARCH (FFR) PROJECT CLOSEOUT

This memo provides notice of the closeout of the Future Focused Research (FFR) project titled, "APPLICATION OF NATURAL LANGUAGE PROCESSING TO NRC REGULATORY DOCUMENTS FUTURE FOCUSED RESEARCH". The main objectives of this project were to explore Natural Language Processing (NLP) techniques and explore how those tools could be applied to regulatory documents and how they could be leveraged to support increased staff efficiency when performing regulatory reviews.

This work involved three tasks: (1) create a matching based tool to perform Named Entity Recognition (NER) for regulatory references; (2) create a vector space model of regulatory guides to perform similarity searches; and (3) train the Bidirectional Encoder Representations from Transformers (BERT) language model, developed by Google, on 10 CFR Parts 1-199. The results of these efforts are documented in the project report attached to this memo.

This project provided the project lead the opportunity to learn about multiple NLP techniques and how those tools can be leveraged to support agency activities. Multiple entities within the agency have expressed interest in leveraging some of the techniques explored under this effort, especially leveraging NER to extract references to the licensing framework.

CONTACT: Trey Hathaway, RES/DE/REB
301-415-2461
Alfred.Hathaway@nrc.gov

Application of Natural Language Processing to NRC Regulatory Documents
Future Focused Research Project

Cc: Kimberly Webber, RES/DSA
Victor Hall, RES/DSA
Luis Betancourt, RES/DSA/AAB

Prepared by: Trey Hathaway
U.S. Nuclear Regulatory Commission
Office of Nuclear Regulatory Research
Division of Engineering

ABSTRACT

The U.S. Nuclear Regulatory Commission (NRC) is striving to increase efficiency when performing licensing reviews. Natural Language Processing (NLP) is a set of techniques which can be leveraged to extract insights from unstructured data within text documents. These techniques could be leveraged to support staff when performing their daily duties. The techniques explored in this effort include Named Entity Recognition (NER), creating a vector space model to perform similarity calculations for information retrieval (IR), and attempting Masked Language Modelling (MLM) on the Bidirectional Encoder Representations from Transformers (BERT) language model, developed by Google, on regulatory language taken from Title 10 of the Code of Federal Regulations (CFR) Parts 1-199.

NER attempts to label text using predefined categories. Standard NER tools tend to be trained on general language; therefore, they may have difficulty finding insights that require understanding of domain specific jargon. This work attempted to define patterns which would enable these pre-trained models to label and extract text from regulatory documents. The tool developed in this work demonstrated the strength of using NER to find connections between documents within the regulatory framework and created a dataset which could be used to train a more robust NER tool.

Creating a vector space model and performing similarity calculations is an efficient method of sorting documents based on similarity of word usage between documents. In this work, term frequency-inverse document frequency (tf-idf) weighting factors were created for the words used in a document. These weighting factors represented the word usage in a document. Once the text is represented as a vector, similarity calculations were performed using linear algebra. A vector space representation of the regulatory guides was created to return regulatory guides which have language most similar to a licensing action.

Finally, NLP employing neural networks were examined. MLM was performed with the BERT language model using text taken from 10 CFR Parts 1-199. MLM involved masking the portion of a text and having the model train on predicting the masked language. Training a neural network-based NLP tool can enable a vector space-based representation of text to retain semantic meaning, which is lost when employing tf-idf weighting factors.

This work can be considered a knowledge management document to inform staff on an assortment of NLP techniques and how those techniques could be applied to the technical jargon used by the NRC. Multiple branches within the agency have expressed interest in employing these techniques to support their activities.

Table of Contents

ABSTRACT	2
Table of Figures	4
List of Tables	4
Acronyms and Abbreviations	5
1 Introduction	6
1.1 Future Regulatory Need(s) Addressed	7
1.2 Technical Issues(s) Addressed	7
2 Project Summary and results	7
2.1 Task 1: Named Entity Recognition	7
2.2 Task 2: Information Retrieval	10
2.3 Task 3: BERT LLM Training	12
3 Conclusions	17
4 References	19

List of Figures

Figure 2-1. Network diagram demonstrating interconnectedness of regulatory documents. 9
Figure 2-2. Distribution of text passage token length constructed passages taken from 10 CFR Parts 1-199 used to pre-train the BERT model. 13
Figure 2-3. Histogram of text passage token length constructed from passages taken from 10 CFR Parts 1-199 to pre-train the BERT model. 14
Figure 2-4. Training and validation loss curves as a function of hyperparameter optimization. ... 15
Figure 2-5. Training and validation loss curves out to 100 epochs during hyperparameter testing. 16

List of Tables

Table 2-1. List of entities defined for this work with assigned label and example text. 8
Table 2-3. Example MLM steps and predictions. The word *material* is masked from the text and the model is requested to make a prediction for the [mask] token. 17

Acronyms and Abbreviations

BERT	Bidirectional Encoder Representations from Transformers
CFR	Code of Federal Regulations
IR	Information Retrieval
MLM	Masked Language Modeling
NER	Named Entity Recognition
NRC	Nuclear Regulatory Commission
NLP	Natural Language Processing
TF-IDF	Term Frequency-Inverse Document Frequency

1 Introduction

One focus of U.S. Nuclear Regulatory Commission (NRC) is to ensure adequate assurance of adequate protection to public health and safety. It accomplishes its mission through performing regulatory reviews of licensing actions submitted by its regulated entities. The NRC is striving to increase its efficiency in performing licensing reviews. One method that could be employed to support this goal is the application of Natural Language Processing (NLP) to regulatory documents. This effort will apply NLP techniques to regulatory documents to demonstrate capabilities that these techniques could provide. Numerous NLP techniques are available which could be applied to regulatory activities to increase staff efficiency. The techniques explored in this effort include Named Entity Recognition (NER) and Information Retrieval (IR) techniques.

NER is a technique which can be leveraged to extract features from unstructured text. In this case, the tool finds words in text which can be grouped into pre-defined categories, for example, labeling a proper name as a person or a country as a nation, etc. In the examples provided, the entities would be *person* and *nation*. NER tools are typically trained on general language; therefore, the entities will predominantly reflect persons, organizations, dates, etc. Because the NRC uses domain specific terminology, creating user defined entities would be beneficial to staff to enable the extraction of NRC specific terminology. One such example is the extraction of references to regulations, or other portions of the regulatory framework, from documents. This would enable the staff to quickly process long documents and extract regulations and pertinent references. This would enable project managers to be able to tailor their initial reviews to see what information may be available, or missing, from a licensing action. This work will explore rule-based entity matching because of the lack of data available to train some NER techniques. The rule-based tools could then be used to create data suitable to train a more robust NER too.

The IR technique examined in this work is the creation of a vector space representation of a collection of documents. A vector space representation enables the ability to perform similarity calculations between a new document against a collection of documents, therefore, allowing the collection of documents to be sorted based on how similar it is to a new document. In the work, a vector space representation of the regulatory guides will be developed. A licensing action can be compared to the regulatory guides to try to return what regulatory guide seems most applicable to the licensing action based on how similar the licensing action's word usage is to the regulatory guides. The vector space representation employed in this work will leverage term-frequency inverse document frequency (tf-idf) weighting factors for the words used in a document. A tool such as this would quickly offer suggestions of the most applicable regulatory guides which could support the review of a licensing action to a project manager

While the tf-idf method is a straight-forward approach to document retrieval, the semantic meaning of the document is lost. Neural network models are available which can be trained to retain the semantic meaning of a document. Google developed a model known as BERT (Bidirectional Encoder Representations from Transformers), which is pretrained on a large set of documents. The BERT model will be fine-tuned on NRC specific language. The end result would be a language model that "understands" NRC language. The goal of this particular portion of the effort is to provide knowledge management so staff may understand how more complex neural network-based language models are trained.

1.1 Future Regulatory Need(s) Addressed

Licensing actions must be reviewed upon intake to determine completeness. Some applications may include hundreds of pages of documentation. NLP techniques could be employed to assist staff in efficiently performing their initial reviews. For example, if references to the regulatory framework can be quickly extracted from a licensing action, a project manager can determine if applicable regulations are missing from the document. Additionally, if information can be extracted from completed licensing actions, similarity calculations can be performed between an incoming licensing action and the historical licensing actions. This would provide a project manager or reviewer quick access to information that may be missing from an incoming licensing action relative to completed licensing actions. Application of these technologies to NRC activities can increase staff efficiency.

1.2 Technical Issues(s) Addressed

This proposal will explore NLP techniques and apply those techniques to regulatory documents. The goal is to explore how NLP tools can assist reviewers in examining licensing actions. These tools can analyze incoming documents as a whole and return summary information on the licensing action to assist in reviews.

This work proposes to use general purpose open-source NLP tools to extract data, and perform analytics, on regulatory documents. It is proposed to perform the work in stages, where each stage will yield insights that could potentially be helpful to staff performing reviews.

2 Project Summary and results

2.1 Task 1: Named Entity Recognition

NER is a natural language processing technique that attempts to label text into predefined categories. This enables the extraction of data from unstructured text into structured data that characterizes the underlying text. A number of open-source tools are available to perform this task, but these tools tend to be trained on general language and may not perform well on domain-specific language, such as used in regulatory text in NRC Regulatory Guides [1], or other licensing documents. One goal of this research was to attempt to apply NER techniques to regulatory documentations to identify and extract domain-specific references from regulatory documents.

This work used the open-source package spaCy [2]. spaCy enables multiple ways to apply NER techniques to domain-specific language. In this work, the staff used the spaCy *Entity Ruler* routine. Entity Ruler allows a user to pre-define language patterns that define a specific entity. For example, a regulation can have the format:

10 CFR 50.204

A pattern can be defined as a string that looks like a number, followed by a string that matches *cfr*, followed by another string that looks like a number. If this combination is observed in the text, those strings can be labeled as a pre-defined entity, such as `REG`, for regulation, in this case.

The “rules” that are defined to label entities can then be saved to a separate file. This file can then be loaded into other user’s spaCy Entity Ruler pipeline and provide any user the ability to extract

similar entities. By placing the user defined Entity Ruler prior to the spaCy NER tool within the NLP pipeline, the user defined entities can be extracted followed by extracting any entities that can be found using the built-in routine. The strength of using the spaCy Entity Ruler is that it not only enables the ability to label and extract predefined entities from text, but it also extracts the location of the beginning and ending character of the entity within the sentence. This enables the ability to create labeled data that could be used to train future tools, which could be more robust in extracting entity patterns.

For this work, regulatory guides and other licensing actions were examined to try to determine what entities may be of interest to extract from a regulatory guide. These entities include:

Table 2-1. List of entities defined for this work with assigned label and example text.

Entity	Label	Example
Regulations (REG)	REG	10 CFR 50.54(m)(2)(i)
General Design Criteria (GDC)	GDC	General Design Criterion 17
Regulatory Guides (RG)	RG	Regulatory Guide 1.76
NUREG (NUREG)	NUREG	NUREG/CR-6713
Accession Numbers (ACC)	ACC	ML022210067
Federal Registrar Notices (FR)	FR	39 FR 31334
Management Directives (MD)	MD	Management Directive 8.4
SECY Letters (SECY)	SECY	SECY-03-0069
NEI Position Papers (NEI)	NEI	Nuclear Energy Institute (NEI) 00-04

These entities may appear in regulatory guides in a variety of ways. Rules were defined to attempt to define the variety of ways entities can appear in a document. For example, a reference to a regulation can appear in a regulatory document as: 10 CFR 50, Title 10 of the Code of Federal Regulations Part 50, 10 CFR § 50, etc. A variety of rules were defined for the set of entities defined above. The entities were defined as a dictionary in json format. The top level defines the label of the defined entity. This is followed by a list which defines the pattern of successive strings which represent the entity. Finally, an ID is defined for the entity. While much effort went into identifying and defining rules for these entities, the rules are potentially not exhaustive. This is where training a tool with data extracted from the spaCy Entity Ruler may improve performance as the tool would try to learn rules which would define the desired entities without having to predefine them.

The 10 sections of regulatory guides were downloaded, and the text was extracted with the open-source package pdfminer [3]. The text was extracted from the pdf and saved as in a non-relational database, a json file in this case. The text was saved for each page individually. This enables the ability to return the individual pages from a pdf file where the entity appears, therefore allowing a user to quickly find the location of information of interest.

After the data was extracted, each page of the regulatory guides was passed to an algorithm to search for entities. First, a sentence tokenizer was applied to the text on a page. The sentence tokenizer separates the text into individual sentences. The sentences were analyzed individually and if a sentence contained more than 7 words, the Entity Ruler was applied to the sentence to identify if a defined entity was present. If a sentence contained less than 7 words, it was ignored.

2.2 Task 2: Information Retrieval

Information retrieval is an NLP technique which could also be leveraged to support the efficient review of licensing actions. In this instance, regulatory guides are characterized as term frequency-inverse document frequency (tf-idf) vectors, creating a vector space characterization of the set of regulatory guides. The tf-idf parameter is a weighting factor that is calculated for each word in a document. It is the product of two factors, the term frequency and the inverse document frequency. The term frequency is the number of times a term occurs in a document divided by the number of words in a document. It reflects how important a word is to the meaning of a document based on the assumption that if a word occurs frequently in a document, it is important to its meaning. The inverse-document frequency is log of the number of documents in the corpus, or the set of documents of interest, divided by the number of documents that contain a particular term. This parameter reflects the frequency of documents that contain a particular word. If a word occurs infrequently in a corpus, such as technical jargon, it will have a larger idf value compared to common words. The tf-idf, the product of the tf and the idf as stated previously, reflects how important a word is to a document within a set of documents, where larger values indicate the words that are important to both the document and occur less frequently in the corpus of documents.

To create the tf-idf model of a corpus of documents, a vocabulary is created of all the unique terms that occur in the corpus. Words that occur too frequently and infrequently are screened from consideration. If a word occurs too frequently, it is not predictive of any document and will affect performance. If a term occurs too infrequently it may be an outlier, or typo, which may not occur in other documents and therefore not be predictive. Screening out these terms will improve performance by reducing noise in the calculation and reducing the memory requirements of the calculation.

Once the vocabulary is determined, the tf-idf calculation is performed to create a tf-idf weighting factor for each term in the vocabulary for each document. This reduces the corpus to a collection of vectors which reflect the tf-idf weighting factors for the terms in the vocabulary. These vectors therefore represent the words usage within a document and yield a vector space representation of the corpus.

A tf-idf representation of a document not within the corpus can be constructed using the common vocabulary for the corpus. The new document can be compared to the other documents by calculating the dot product between the new document and the vector space representation of the corpus. The dot product reflects the “angle” between the new document and the documents within the corpus. If this angle is close to 0, the new document and document within the corpus have similar word usage, and, therefore, are assumed to be describing a similar topic. As the angle increases, that is the dot product approaches 0, the documents are further apart in word usage, and, therefore, meaning. The corpus of documents can then be sorted based on this angle to return documents most similar to a new document.

The python package GENSIM [4] was used to create the vector space model of the NRC Regulatory Guides. The text of the regulatory guides was first cleaned by removing URLs, email addresses, and telephone numbers. Common English words are also removed by using NLTK’s (Natural Language Toolkit) [5] English stopwords list. This list was supplemented by common

words found through an analysis of the regulatory guides, although this step was probably not necessary as common words are filtered from consideration, as discussed later.

NLKT's `WordNetLemmatizer` was also used to lemmatize the data. Lemmatization is a regularization technique where the shortest representation (lemma) of a wordnet corpus is used to replace the word in a text, for example running may be reduced to run. The goal is to reduce the number of ways text may be used to reflect an idea.

Once the text is cleaned, ngrams are created for the text using the GENSIM Phrases (`models.Phrases`) and Phraser (`models.phrases.Phraser`) routines. The Phrases module detects common combinations of words in texts and the Phraser module locks the Phrases model by discarding portions of the model no longer needed to improve performance. These modules were used to construct bigrams (common pairs of words occurring in the text) and trigrams (common three-word combinations occurring in the text). Finally, GENSIM was used to create a dictionary (`corpora.Dictionary`) based on the text from the regulatory guides. This dictionary was filtered with the GENSIM (`dictionary.filter_extremes`) so words that occur in over 85% of the regulatory guides are screened from the dictionary. This is done to limit the size of the tf-idf model to improve performance. Finally, GENSIM was used to create the tf-idf model (`models.TfidfModel`) for the regulatory guides and similarity calculations were performed with GENSIM's similarity module (`similarities.Similarity`).

It was found to be challenging to find a licensing action that would rely solely on a regulatory guide which made testing of the technique challenging. But when testing, a licensing action which discussed steam and power conversion systems returned Regulatory Guide 1.68.1, *Initial Test Program of Condensate and Feedwater Systems for Light-Water Reactors*. This particular test document was shorter, approximately 30 pages, and a cursory review did indicate the document discussed steam and feedwater systems, as well as initial test programs. A different test document on nuclear design of a design control document of a pressurized water reactor returned Regulatory Guide 1.236, *Pressurized-Water Reactor Control Rod Ejection and Boiling-Water Reactor Control Rod Drop Accidents*. This test document covered many different aspects; therefore, selection of a particular most applicable regulatory guide may prove challenging. Because of this, chunking of the licensing action into smaller portions may help the tool return the most pertinent information. Because semantic context is lost when using tf-idf methods, breaking the licensing action into chunks would help the tool focus on portions of the text, rather than averaging out the language usage over the entire text. This would enable the tool to return the most pertinent supporting information in the specific portion of the underlying licensing action.

Additionally, the technique would probably be more beneficial if the vector space model included more documentation from the licensing framework. Regulatory guides can range from short documents which may endorse a standard, or reference a NUREG which provides a technical justification, to long documents that cover a range of information. This may depend on the age of the regulatory guide, etc. Expanding the tool to include additional documentation could help the tool return the most pertinent information. For example, using the insights from the NER tool discussed in Section 2.1 to determine what additional documents should be incorporated into the vector space model could be beneficial.

2.3 Task 3: BERT Large Language Model (LLM) Training

BERT (Bidirectional Encoder Representations from Transformers) [6] was released by Google in 2018 and represented some of the first advancements in neural network based natural language processing, providing state-of-the-art performance for a wide range of tasks. Neural network-based language models represent words, or word pieces, as tokens. These tokens are simply a number that represents the word, or word piece. The model then performs predictions on the sequence of numbers, which can then be changed back to text. Unlike models that perform next token prediction, BERT is pre-trained to look at sentences from both the left and the right, therefore, context of the whole text is used to inform predictions, rather than the preceding text. Once pre-trained, BERT can be fine-tuned on a wide range of tasks, from question answering to summarization.

One method to fine-tune BERT is using Masked Language Modeling (MLM) rather than predicting the next word in a sentence. In MLM, random words are replaced with a mask token (`[mask]`). The text with masked tokens is then provided to the BERT model to make a prediction, where the prediction is compared to the original sentence. The model is then trained to minimize the loss between the model prediction on the text with a mask token and the original sentence. The objective of MLM is to enable the model to learn features from both the left and right of the text to predict language within the sentence. In addition to MLM, BERT was also trained on next sentence prediction, where the model attempts to predict if two sentences should follow each other or not. While MLM modeled was explored in this effort, next sentence prediction was not.

The packages used for this effort were from the Huggingface library [7], namely `BertTokenizer.from_pretrained` and `BertForMaskedLM.from_pretrained` [8]. These models were from the `transformers` package. For this work, the BERT base model was used, `bert-base-uncased`. This model has 110 million parameters. The corresponding word piece tokenizer was also used. These models are originally trained on general language; therefore, they may lack understanding of technical language, or jargon, that would be specific to the NRC. The goal of this work was to attempt to perform MLM pretraining of the BERT base model using NRC specific language. Data used to pre-train BERT in this effort is the text of Title 10 of the CFR Parts 1-199 [9], or regulations specific to the NRC. This data was chosen because it could be easily obtained from the NRC website in electronic form.

The tokenizer used for the base model has a vocabulary of approximately 30,000 words, or pieces of words. For example, the word *operational* may be broken into two word-pieces, *operation* and *##al*. This enables the model to represent a wide range of text into a limited number of words. Because the tokenizer is trained on general language, technical jargon may require breaking words into numerous tokens. It could be possible to create a tokenizer that could work directly on NRC language and understand the domain specific jargon, by processing the text of a wide variety of documents and determining a set of word pieces that describe that text. This would enable the tool to capture jargon, for example acronyms, that are specific to the NRC, e.g. NRR, LOCA, etc. Creating a domain specific tokenizer is beyond the scope of this effort.

The BERT model can operate on sequences of 512 tokens. Each part of 10 CFR Parts 1-199, not the individual sentences, were tokenized to determine the number of tokens present in the part.

Text outside of the 512 token sequence was trimmed, and therefore ignored by the model. To maximize the data available for training, the text was broken into 375-word word-chunks. This meant nearly 90% of the sequences were less than the 512 token limit, as seen in Figure 2-2, resulting in 4,545 sequences. A train-test split was created with those sequences, where 90% of the text was used for training and 10% was used for testing, yielding 4,090 pieces of text for training and 455 pieces of text for validation. If the text was less than 512 tokens, a padding token was applied to the sequences so they were 512 tokens long. For the 375-word word-chunks, the average token length was 347.6 ± 190.3 . Figure 2-3 presents a histogram of the length of text used for pre-training the BERT model. The peak around 500 tokens can be attributed to chunking the text into 375-word word-chunks.

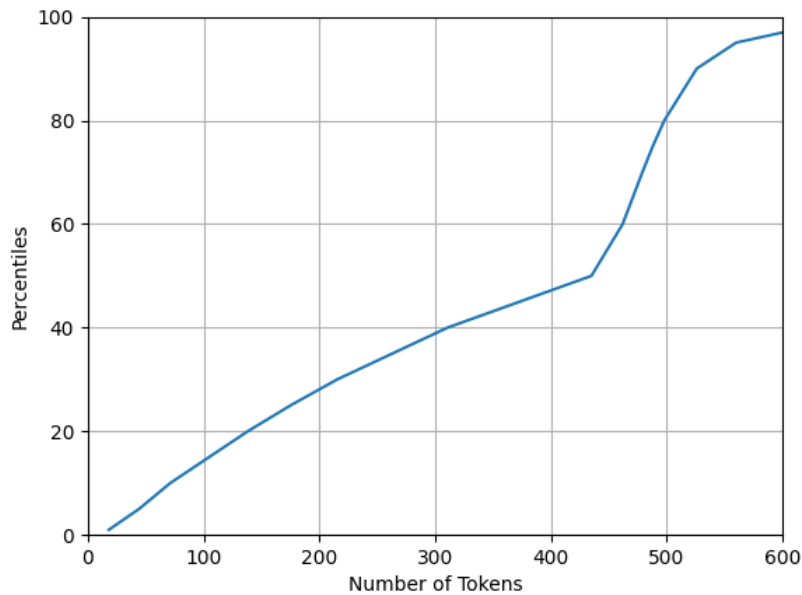


Figure 2-2. Distribution of text passage token length constructed passages taken from 10 CFR Parts 1-199 used to pre-train the BERT model.

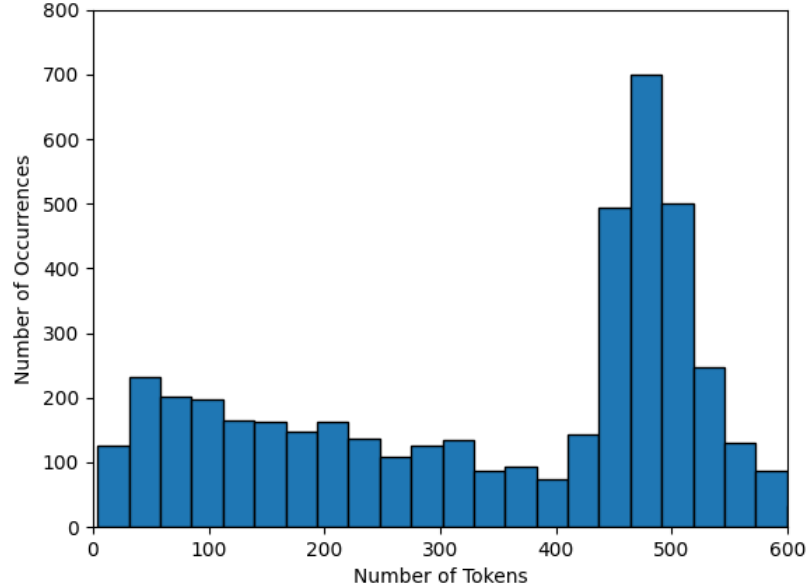


Figure 2-3. Histogram of text passage token length constructed from passages taken from 10 CFR Parts 1-199 to pre-train the BERT model.

Once the text of 10 CFR Parts 1-199 was chunked and tokenized, the data was prepared to perform the MLM. To accomplish this, 15% of the tokens were randomly selected from the sequences. The following was applied to those selected tokens:

- 80% of the selected tokens were replaced with a mask token, [MASK]
- 10% of the selected tokens were replaced with a random token from the tokenizers dictionary
- 10% of the selected tokens were left unchanged

Because 20% of the selected tokens are not changed to the masked token, the encoder does not know what words it is trying to predict, therefore, it is forced to try to maintain the contextual distribution of the text. That is to say, the model would not be simply looking to replace a [MASK] token with a prediction token. Replacing the selected token with a random token forces the model to consider the word itself and leaving 10% of the tokens unchanged biases the model to look for the observed word. BERT requires the addition of a [CLS] token to the beginning of the sequences and a [SEP] token to the end of a sequence to prepare the data for MLM.

Hyperparameters were tested to see what model gave the best performance, that is, the lowest loss. The optimal hyperparameters taken from Appendix A.3 of the BERT paper [6] were tested, but that does not imply these are the optimum parameters. The tested parameters included:

- Batch size: 16, 32
- Learning rate (Adam): 5e-5, 3e-5, 2e-5
- Number of epochs: 4

The training was performed in a loop and the learning loss curves were examined to observe the performance as seen below. The batch size was defined in the data generator, the learning rate was defined in the optimizer, and the model was trained for 4 epochs for each iteration. The results of this optimization can be seen in Figure 2-4.

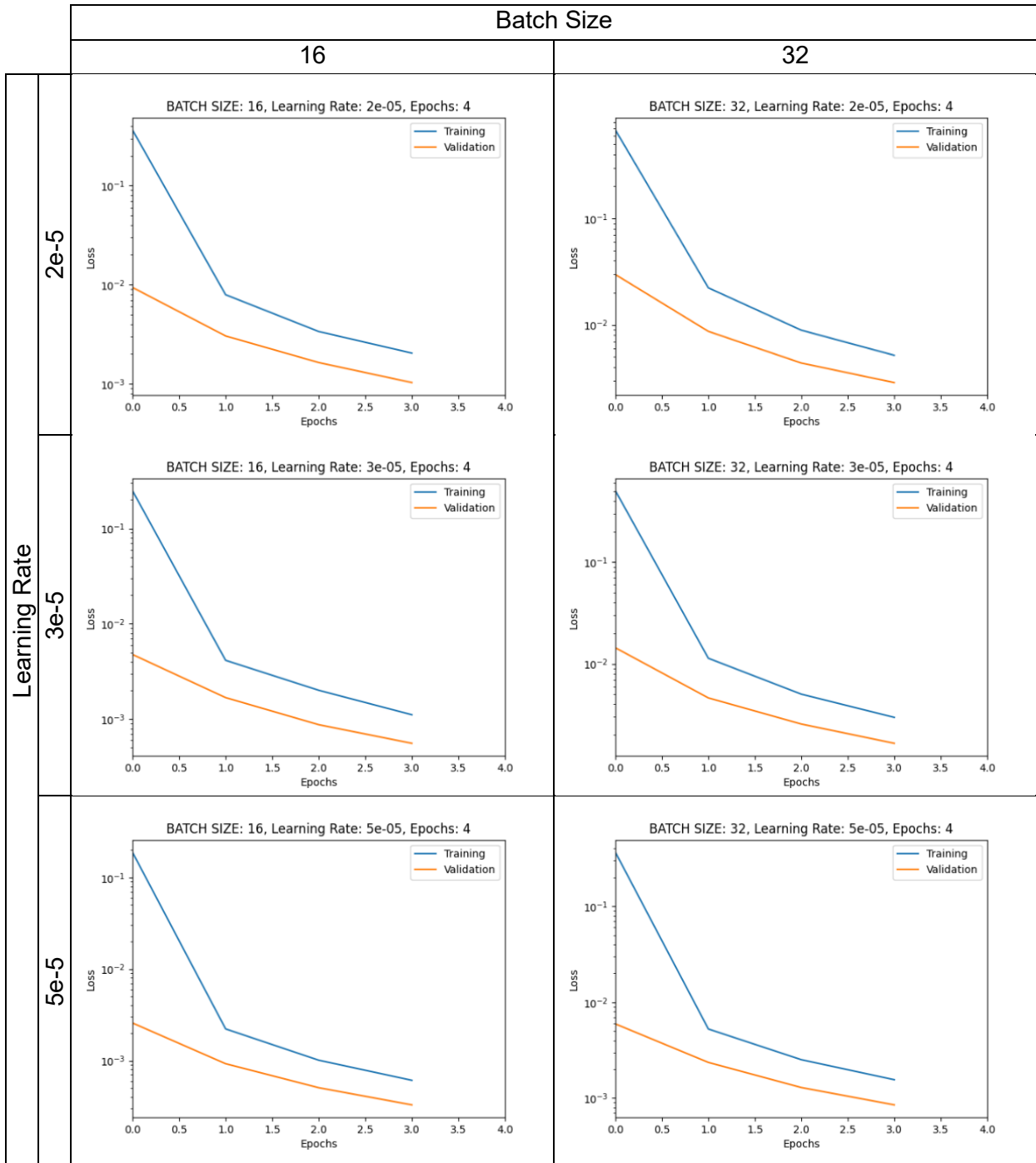


Figure 2-4. Training and validation loss curves as a function of hyperparameter optimization.

The selected hyper parameters resulted in the lowest loss for the training and validation set:

- Batch size: 16
- Learning rate (Adam): 5e-5
- Number of epochs: 4

It is not claimed that these are the optimum hyperparameters used to finetune the BERT model with data taken from 10 CFR Parts 1-199. The selected batch size and learning rate were used to train the model for 25 epochs, which further reduced the loss, as seen in Figure 2-5.

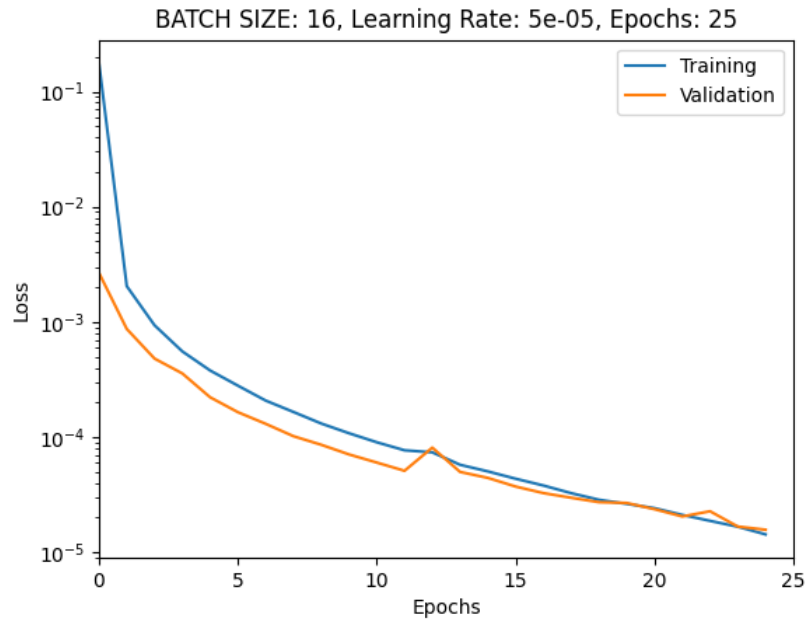


Figure 2-5. Training and validation loss curves out to 100 epochs during hyperparameter testing.

Below is an example of MLM using the model discussed above. The introduction section of 10 CFR Part 72.23 is taken as an example text and a single word is randomly masked. In this case, the word *material* is masked from the sentence. The train model is added to the Huggingface transformers pipeline package and a `fill_mask` is requested to return the top 5 predictions for the masked token. In this instance, the top prediction does return the masked word. This statement also demonstrates the challenges with training a model with NRC specific technical language. NRC technical jargon includes a lot of numbers. In more traditional NLP techniques, removing numbers is a typical cleaning technique. It is challenging to remove numbers and maintain meaning for NRC technical jargon; therefore, a lot of data and training may be needed to create a suitable model to predict aspects of the language used.

Table 2-2. Example MLM steps and predictions. The word *material* is masked from the text and the model is requested to make a prediction for the [mask] token.

Text	this section contains specific requirements for the protection of safeguards information in the hands of any person subject to the requirements of § 73.21(a)(1)(ii) and research and test reactors that possess special nuclear material of moderate strategic significance or special nuclear material of low strategic significance.
Masked text	this section contains specific requirements for the protection of safeguards information in the hands of any person subject to the requirements of § 73.21(a)(1)(ii) and research and test reactors that possess special nuclear [MASK] of moderate strategic significance or special nuclear material of low strategic significance.
Tokenization	['this', 'section', 'contains', 'specific', 'requirements', 'for', 'the', 'protection', 'of', 'safeguard', '##s', 'information', 'in', 'the', 'hands', 'of', 'any', 'person', 'subject', 'to', 'the', 'requirements', 'of', '§', '73', '.', '21', '(', 'a', ')', '(', '1', ')', '(', 'ii', ')', 'and', 'research', 'and', 'test', 'reactors', 'that', 'possess', 'special', 'nuclear', '[MASK]', 'of', 'moderate', 'strategic', 'significance', 'or', 'special', 'nuclear', 'material', 'of', 'low', 'strategic', 'significance', '.']
Encoding	[101, 2023, 2930, 3397, 3563, 5918, 2005, 1996, 3860, 1997, 28805, 2015, 2592, 1999, 1996, 2398, 1997, 2151, 2711, 3395, 2000, 1996, 5918, 1997, 1073, 6421, 1012, 2538, 1006, 1037, 1007, 1006, 1015, 1007, 1006, 2462, 1007, 1998, 2470, 1998, 3231, 22223, 2008, 10295, 2569, 4517, 103, 1997, 8777, 6143, 7784, 2030, 2569, 4517, 3430, 1997, 2659, 6143, 7784, 1012, 102]
Predictions	Score
material	0.7430110573768616
materials	0.14589929580688477
weapons	0.011325751431286335
technology	0.005938297603279352
equipment	0.004544890020042658

The strength of the BERT model, and other open source LLMs, is the flexibility of capabilities which can be leveraged. For example, further layers can be added to the base BERT model to create an autocomplete functionality or layers can be added to classify text. Also, embeddings of text can be taken from the models to create vector space representations for text. This can be combined with other techniques to perform semantic searches.

3 Conclusions

The goal of this work was to demonstrate how natural language processing could be applied to regulatory documents to support increased efficiency of agency activities. NER has promise to support increased efficiency in many agency use cases. For example, the NER tool developed in this work was used to support a regulatory readiness assessment by quickly linking numerous regulatory guides to the underlying regulations they were meant to support. Therefore, if the novel

technologies were found to challenge the guidance provided in a regulatory guide, the underlying regulations could be identified to examine if they were robust enough to enable the novel technologies.

Comparisons of licensing actions to regulatory guides to identify applicable regulatory guides showed promise, but it was challenging to find a licensing action that focused on a particular regulatory guide for testing. Future testing could include extending the library of documents included in the model to include other documents from the licensing framework, for example, Technical Specification Task Force travelers, etc. This could extend the usefulness of the tool. Additionally, because the documents used in the model and potential licensing actions may significantly vary in length, breaking the documents into chunks may help the models perform better in regard to identifying applicable portions from the regulatory framework.

Combinations of these tools could be leveraged to increase efficiency when making assessments of licensing actions or the underlying regulatory framework. For example, NER tools could be used to extract references to the regulatory framework from historical licensing actions. These references could be treated as features defining the underlying licensing action. Information retrieval models could be created from those historical licensing actions and similarity calculations could be performed between a new licensing action and the historical licensing actions to find the most similar licensing requests. The references from the incoming action and the similar historical licensing actions could be compared to quickly determine to if an incoming action considers similar references as the historical licensing actions. This could give a reviewer an idea of how complete a licensing action is by determining if similar references are used, or if similar regulations are addressed.

The straightforward information retrieval techniques explored in this effort lose the semantic meaning of the underlying documents. Neural network-based models can learn the underlying semantic meaning of text and represent text as a vector of numbers. For this work, the BERT LLM model was trained on NRC specific regulatory language, 10 CFR Parts 1-199, to demonstrate how this model could be fine-tuned on NRC domain specific language. This was used as a knowledge management exercise to demonstrate how these models work. Additional effort would be required to use these models to create the vector representation of the underlying documents. For example, the regulatory documents may be significantly longer than the context window of the LLM model, therefore, an appropriate way to represent the text would need to be explored.

4 References

- 1 U.S. Nuclear Regulatory Commission. *NRC Regulatory Guides: Divisions 1 through 10*. Washington, DC. Link: <https://www.nrc.gov/reading-rm/doc-collections/reg-guides/index.html>.
- 2 Honnibal, M. and Montani, I. *spaCy: Industrial-strength NLP*. 2017. Link: <https://spacy.io/>.
- 3 Shinyama, Y. *pdfminer*. 2019. Link: <https://pypi.org/project/pdfminer/>.
- 4 Rehurek, R., Sojka, P. *Gensim – python framework for vector space modeling*. NLP Center, Faculty of Informatics, Masaryk University, Brno, Czech Republic, 3(2). *GENSIM: topic modelling for humans*. 2011. Link: <https://radimrehurek.com/gensim/index.html>.
- 5 Bird, S., Klein, E., Loper, E., et al. *Natural Language processing with Python: analyzing text with the natural language toolkit*. 2009. Link: <https://www.nltk.org/> .
- 6 Devlin, J., Chang, M., Lee, K., Toutanova, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Google AI Language. 24 May 2019. arXiv: 1810.04805v2. Link: <https://arxiv.org/abs/1810.04805>.
- 7 Wolf, T. et al. *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. 14 Jul 2020. arXiv: 1910.03771. Link: <https://arxiv.org/abs/1910.03771>.
- 8 Huggingface. *BERT*. Link: https://huggingface.co/transformers/v3.0.2/model_doc/bert.html.
- 9 U.S. Nuclear Regulatory Commission. *NRC Regulations, Title 10 of the Code of Federal Regulations (CFR) Parts 1 through 199*. Washington, DC. Link: <https://www.nrc.gov/reading-rm/doc-collections/cfr/index.html>.