Master Thesis Artificial Intelligence

The dual-use of BERT for regulatory compliance.

Bono van Dooren
Utrecht University
Student number 3494942

July 10, 2022

First supervisor: Dr. M.P. Schraagen
Second supervisor: Prof. dr. S. Brinkkemper

Contents

Contents 1

1 Introduction and Motivation 1

2 Related work 3
- 2.1 Text representations . . . 3
  - 2.1.1 One-Hot Encoding . . . 3
  - 2.1.2 Bag of Words . . . 3
  - 2.1.3 Term Frequency-Inverse Document Frequency . . . 4
  - 2.1.4 Word Embeddings . . . 4
  - 2.1.5 Contextualized Word Embeddings . . . 4
- 2.2 Part-of-Speech tagging . . . 5
  - 2.2.1 Rule-based . . . 5
  - 2.2.2 Hidden Markov Models . . . 5
  - 2.2.3 Deep learning models . . . 5
- 2.3 Text classification methods . . . 6
  - 2.3.1 Probabilistic Graphical Models . . . 6
  - 2.3.2 K-Nearest Neighbors . . . 6
  - 2.3.3 Support Vector Machines . . . 6
  - 2.3.4 Recursive Neural Network . . . 7
  - 2.3.5 Feed Forward Neural Network . . . 7
  - 2.3.6 Recurrent Neural Network . . . 7
  - 2.3.7 Convolutional Neural Network . . . 7
  - 2.3.8 Attention . . . 8
- 2.4 Knowledge representation . . . 9
  - 2.4.1 Knowledge Graphs . . . 9
  - 2.4.2 Knowledge Graph Embeddings . . . 10
  - 2.4.3 Improving LMs through KGs . . . 12
- 2.5 Evaluation metrics . . . 15

3 Data and Methodology 16
- 3.1 Datasets creation . . . 18
  - 3.1.1 Taxonomy . . . 18
  - 3.1.2 Dual-use goods licences dataset . . . 18
  - 3.1.3 Bill of lading dataset . . . 22
- 3.2 Knowledge sources . . . 23
  - 3.2.1 Regulation (EU) 2021/821 . . . 23

3.2.2 Wikidata . . . 24
3.2.3 Tools for Innovation Monitoring . . . 25
3.2.4 KeLM Corpus . . . 25
3.2.5 Sentence embeddings . . . 25
3.3 Knowledge Graph Embeddings . . . 26
3.3.1 KGE algorithms . . . 26
3.3.2 KGE triplet sets . . . 27
3.3.3 Sentence embeddings . . . 27
3.4 Language Models . . . 28
3.4.1 Baseline . . . 28
3.4.2 Logistic Regression . . . 28
3.4.3 BERT . . . 29
3.4.4 Parameter settings . . . 30
3.5 Evaluation . . . 31
3.5.1 Evaluation metrics . . . 31

4 Results **34**
4.1 Validation results . . . 34
4.1.1 Best settings . . . 34
4.1.2 Logistic Regression . . . 34
4.1.3 BERT . . . 35
4.2 Test results . . . 35
4.2.1 Logistic regression . . . 35
4.2.2 BERT . . . 36

5 Discussion **43**
5.1 Interpretations . . . 44
5.1.1 Logistic Regression . . . 44
5.1.2 BERT . . . 44
5.2 Limitations . . . 46
5.2.1 Datasets . . . 46
5.2.2 Knowledge Graph Embeddings . . . 48
5.2.3 Language Models . . . 49
5.3 Future work . . . 50

References **51**

Appendix A Filtered Harmonized System codes **59**

Appendix B Code **62**

Abstract

Dual-use goods are items that have both commercial and military or proliferation applications. This thesis investigates whether BERT, a contextual language model, can be used to identify dual-use goods based on a short description of the good and whether it can be improved by augmenting it with relational dual-use knowledge. Two methods from augmenting BERT are explored: by further pre-training BERT on relevant synthetic sentences from the KeLM corpus and by augmenting it with knowledge graph embeddings (KGEs) created from Wikidata. The use of KGEs can improve the performance of a logistic regression model on the dual-use identification task. All implementations of BERT perform well on the dual-use identification task and have a better performance compared to the logistic regression models. None of the BERT implementations augmented with relational dual-use knowledge outperformed the plain implementation of BERT.

Chapter 1

Introduction and Motivation

Dual-use goods are items that have both commercial and military or proliferation applications (Nederlandse Vereniging van Banken, 2020). The EU controls the export, transit and brokering of dual-use items so the EU can contribute to international peace and security and prevent the proliferation of Weapons of Mass Destruction (European Commission, 2021b). This means all companies engaged in international trade are expected to be able to classify dual-use goods and determine whether they require a license (Rijksoverheid, 2019). Being compliant with export control and sanctions legislation is of great importance for these companies as violation of this legislation can lead to severe penalties as well as reputational damage (Kneppelhout, 2021).

The interpretation of dual-use requires a degree of technical knowledge that many companies involved in international trade cannot be expected to possess (Wolfsberg Group, 2019). There is no universal language for describing the goods being traded and documentation used in trade, such as bills of lading or letters of credit, can be very different from the way a regulator may describe a regulated good (Dow Jones, 2019). By design, these documents contain only a short, open-text description of the traded good.

Deep learning based language models (LMs) have shown increasingly impressive results on various natural language processing (NLP) tasks. In particular, Bidirectional Encoder Representations from Transformers (BERT; Devlin et al. (2018)) pushed the state of the art. To improve its coverage of knowledge, many studies recently have focused on augmenting BERT with knowledge graphs (KGs), a structured representation of facts, consisting of entities, relationships, and semantic descriptions. By representing these KGs into low-dimensional vectors while capturing their semantic meanings, or knowledge graph embeddings (KGEs), BERT can be improved on downstream tasks such as text classification.

The aim of this research project is to retrieve useful information from different knowledge sources, such as Regulation (EC) No 2021/821 (Council of European Union, 2021), TIM (European Commission, 2021c) and Wikidata (Vrandečić & Krötzsch, 2014), to build a model that can identify dual-use goods based on a short description of the good. In order to achieve this goal, I will specifically focus on the use of the contextual language model BERT and the improvement of this model by complementing it with relational world knowledge about dual-use goods. This results in the following research questions:

1. *Can a contextual language model such as BERT be used to identify dual-use goods based on their description?*
2. *Can augmenting this model with relational dual-use good knowledge improve its performance on the dual-use good classification task?*

These questions are addressed by creating two datasets, one composed dataset of issued dual-use licenses and one empirical dataset consisting of actual bill of lading data. These datasets are used to train, validate and test BERT's performance on the dual-use identification task: classifying the correct dual-use control entry of a good as specified in Regulation (EC) No 2021/821 (Council of European Union, 2021) based on a description of the good.

Regarding the second research question, BERT is augmented with relational knowledge about dual-use goods using two different methods. The first method relies on the Knowledge-Enhanced Language Model (KELM) Corpus (Agarwal et al., 2020) and uses its synthetic natural language sentences based on KGs for further pre-training to incorporate relational knowledge. The second method creates KGEs from relevant dual-use KGs and enriches BERT directly with these KGEs to augment BERT with relational dual-use goods knowledge and to improve its performance on the dual-use identification task.

A key contribution of this research is the application of BERT in an unexplored specialised domain. The dual-use identification task as proposed in this research has distinct characteristics (e.g. short descriptions, almost no punctuation) and requires domain-specific knowledge in regulatory compliance. The proposed BERT models are able to classify dual-use goods reasonably well and these results persist when tested on empirical data. This suggests these models can contribute to a solution for companies involved in international trade that lack the technical knowledge required for obligatory dual-use identification.

Contributing to the current literature on augmenting BERT with relational knowledge through KGEs, enriching BERT with relevant KGEs does not improve its performance on the dual-use identification task. However, as the same KGEs are improving the performance of simple logistic regression models on the same task, this suggests BERT already encodes, or at least is able to substitute, the relational knowledge present in KGEs on dual use goods.

This thesis is structured as follows. First related work covering text representations, part-of-speech tagging, text classification methods and knowledge representations is presented in Chapter 2. The following Chapter 3 presents the approach and methodology. It starts with the creation of the dataset and the knowledge sources used to augment BERT. Then it covers the creation of KGEs, describes the architectures of the language models and ends with the evaluation metrics used to test the performance of the models. Results are presented in Chapter 4. Finally, a discussion in Chapter 5 concludes this thesis.

Chapter 2

Related work

Dual-use identification is treated as a text classification task in this research, as the aim is to assign a label to a short text describing the good. In general, text classification aims to assign labels to all sorts of texts, such as sentences, paragraphs and documents. It extracts features from raw text data and predicts the pre-defined categories of text data based on such features and is used in applications as sentiment analysis, topic labeling, news classification and question answering (Li et al., 2020).

This chapter starts with briefly describing different methods of text representation for text classification in Section 2.1. As important dual-use terms in this research are retrieved with the use of Part-of-Speech (POS) tagging, a short history of this approach will be presented in Section 2.2, before providing an overview of text classification algorithms in Section 2.3. Section 2.4 introduces knowledge graphs (KGs) and elaborates on knowledge present in the popular transformer-based model BERT. It also provides an overview of research that aims to integrate structured KG information into LMs to improve the text classification task.

2.1 Text representations

Text representation is key to text classification and other natural language tasks, as text data has to be represented such that models can process it effectively. Text can be represented in many ways, but not all representations capture the same information. With the rise of deep learning models text representations shifted from engineered features to automatically extracted ones. This section describes text representation techniques that have been used for text classification, from the basic traditional extraction methods to the features automatically extracted by deep learning methods.

2.1.1 One-Hot Encoding

In general, one-hot encoding is a simple representation in which a categorical variable is represented in a sparse vector. Concerning words, it is a representation in which a word is converted to a vector of N dimensions, where N is the number of words in the vocabulary. Each position in the vector represents a word. A word is represented as a single 1 with the other positions filled with 0s.

2.1.2 Bag of Words

In the Bag of Words (BOW) representation the text is also converted to a vector. Each element in the vector indicates the frequency of each word in the text. Words are selected based on specific criteria such as word frequency. The BOW representation does not capture the semantics of a word,

as words with the same meaning will not have similar vectors (Kowsari et al., 2019). Information about word order is also lost in this representation (N. Liu et al., 2005).

2.1.3 Term Frequency-Inverse Document Frequency

Jones (1972) proposed Term Frequency-Inverse Document Frequency (TF-IDF) to include a weighting factor to reflect the importance of a word in a document. The TF measures the occurrence of a term in a specific document. It's (inverse) occurrence in other document is represented in the IDF. There are several ways in which term frequency and inverse document frequency can be calculated.

TF can be represented as a binary (1 if a given term is present), as the raw counts of the term or different weighting schemes for example adjusting for document length. The IDF can also be calculated with different weighting schemes. The standard weighting scheme is calculated by taking the logarithm of the result of dividing the total number of documents by documents containing the term. A variation involves smoothing, which avoids dividing by zero if a given term is not present in the corpus.

The TF-IDF of a term is high if the term is frequent in the document (i.e. TF is high), but when it is rare in other documents (i.e. IDF is high). A word that is common in all documents does not get a high TF-IDF weight. TF-IDF cannot account for the similarity between words in the document (Kowsari et al., 2019).

2.1.4 Word Embeddings

In word embeddings, words are represented as dense vectors of real numbers that are derived by training neural-network LMs. These methods are designed to capture the semantics of a word, i.e. similar words will have similar vectors. Various word embedding methods have been proposed.

Mikolov et al. (2013) proposed two architectures that layed the foundation of word embedding development: Skip-gram and Continuous Bag-of-Words (CBOW). Both learn word representations by using feedforward neural networks. Skip-gram predicts surrounding words given the current word, while CBOW predicts the current word based on the context. Both these architectures are implemented in Word2Vec.

The approach in Global Vectors for Word Representation (GloVe) by Pennington et al. (2014) resembles that of Word2Vec in developing word embeddings. It trains on global word-word co-occurrence counts, leveraging both local and global corpus statistics and implementing these corpus into the word representation vectors. GloVe captures meaningful linear substructures prevalent in methods like Word2Vec.

FastText (Bojanowski et al., 2017) is another approach to develop word embeddings. Based on Skip-gram, the model represents each word as a the sum of their character n-gram representations. It allows to compute word representations that are not seen in the training process and recognizes the morphology of words.

2.1.5 Contextualized Word Embeddings

The word embeddings discussed in Section 2.1.4 have a difficult time incorporating the context in which a word is used: a word will have the same vector representation regardless of its context. In order to solve this problem, approaches have been proposed that develop embeddings based on the context in which they are used. These are called contextualized word embeddings.

Melamud et al. (2016) propose learning generic context embeddings using bidirectional LSTMs for their context2vec. It's architecture is based on CBOW where the modeling of the surrounding words by averaging the embeddings is replaced by a neural model. Peters et al. (2018) improves this

technique with deep contextualized word representations. They create Embeddings from Language Models (ELMo) in which each token is the concatenation of its forward and backward representations. The forward representation comes from a LM that computes the probability of the sequence by modeling the probability of the token given its history. The backward LM predicts the previous token given the future context and results in the backward representations. ELMo models both complex characteristics of word use and how these uses vary across contexts.

BERT (Devlin et al., 2018) uses a bidirectional Transformer to develop contextualized word embeddings that can be fine-tuned for downstream tasks. Its pre-trained word embeddings are regularly used in state-of-the-art architectures for text classification, as will be discussed in the Section 2.3.

Instead of treating a sentence as a sequence of words, Akbik et al. (2018) pass sentences as sequences of characters into a character-level language model to form word-level embeddings. They propose a contextualized character-level word embedding which can handle subword structures, misspelled and rare words better than previous contextual word embedding models, while keeping the benefits of earlier models.

2.2 Part-of-Speech tagging

In this research, information from the Regulation (EU) 2021/821 is retrieved with the use of part-of-speech (POS)-tagging. POS-tagging is a sequence labeling task in which each token in the sequence is assigned a part-of-speech tag, such as noun (NN) or verb (VB). This section provides a short history of POS-tagging approaches.

2.2.1 Rule-based

The earliest methods to annotate words automatically with part-of-speech tags involved rule-based systems. Klein & Simmons (1963) describe such a model for grammatical coding of English words. Error rates in rule-based POS-taggers are relatively high. Brill (1992) proposed a simple rule-based POS-tagger which automatically acquires its rules. This approach demonstrated that stochastic models (discussed in the next section) are not the only viable option for POS-tagging.

2.2.2 Hidden Markov Models

A hidden Markov model (HMM) is a statistical model which assumes the state of the model is generated by an underlying hidden state. For POS-tagging, words and their POS-tags can be represented in the states and hidden states of the HMM. HMMs have been used extensively for POS-tagging (Church, 1989; Cutting et al., 1992; Brants, 2000). These methods were effective, but the amount of domain knowledge and efforts on feature engineering make them difficult to extend to new areas (He et al., 2020). The same holds for the maximum entropy Markov model (Ratnaparkhi, 1996).

2.2.3 Deep learning models

Extracting valuable information from texts with deep learning models has also gained popularity from researchers for sequence labeling tasks, showing increasingly impressive results. As word morphological and shape information is extremely useful for POS-tagging, many studies use CNNs or RNNs that can learn the character-level representations of words and incorporate them in the embeddings. This also improves performance on out-of-vocabulary words (He et al., 2020).

Context dependency is important in POS-tagging. There are three commonly used deep learning architectures that extract contextual features and capture context dependencies of a given input sequence: RNNs, CNNs and Transformers (He et al., 2020). The Bi-LSTM is a commonly used context encoder architecture and has achieved excellent performance on the POS-tagging task (Huang et al., 2015; Ma & Hovy, 2016).

2.3 Text classification methods

Text classification is a fundamental task in natural language processing. Methods based on statistical theory have been applied to text classification since the early 1960s. More recently, machine learning methods have lead to great advances in text classification. This section covers the early, traditional methods of text classification to the more recent, deep learning-based methods. It covers transformer-based models more extensively as this research will build on the transformer-based model BERT.

2.3.1 Probabilistic Graphical Models

Probabilistic Graphical Models (PGMs) use a graph-based representation to model variables and their direct probabilistic interaction. A Bayesian network is a PGM with a directed graph (where the edges have a source and a target) (Koller & Friedman, 2009). Naive Bayes (NB) is a simple Bayesian network that has been applied to text classification since the 1960s (Maron, 1961). It assumes that all attributes of examples are independent of each other given the context of the class. Although this assumption is usually false in real-world tasks, the naive Bayes classifier performs well on the text classification task (McCallum et al., 1998), but it is very sensitive to feature selection (Chen et al., 2009).

2.3.2 K-Nearest Neighbors

The K-Nearest Neighbor (KNN) algorithm is a classification algorithm that assigns the majority label of the nearest k samples in the training set to an unclassified observation (Cover & Hart, 1967). The KNN classifier has been used effectively for the text classification task (Y. Yang, 1999). It has a high degree of calculation complexity for classification, as it uses all training samples. Difference in importance between training samples is not accounted for. Yong et al. (2009) proposed an improved KNN text classification algorithm that accounts for these issues by introducing a weight value to indicate a degree of importance. Other improvements include an adaptive parameter k sensitive to class imbalances (Baoli et al., 2004) or using feature similarity (Jo, 2017).

2.3.3 Support Vector Machines

Support Vector Machines (SVMs) (Cortes & Vapnik, 1995) find the decision boundary that maximizes the margin between classes. By representing each text as a vector, Joachims (1998) used SVMs for the first time for the text classification task. He argues this approach works well on characteristics of text data: the high dimensional input space related to text features, fewer irrelevant features and sparse document vectors. As Transductive SVMs (TSVMs) inherit most properties of SVMs and can be used to leverage co-occurrence information prevalent in text data, Joachims et al. (1999) propose TSVMs for text classification. This shows improvements over the SVM approach, especially for small datasets. A. Sun et al. (2009) find the characteristics of text data mentioned by

Joachims (1998) to contribute to the poor performance of weighting strategies for imbalanced text classification with SVMs.

2.3.4 Recursive Neural Network

Since the 2010s, text classification has gradually changed from traditional models such as PGMs, KNN and SVMs to deep learning models. As deep learning models automatically provide semantically meaningful representations, text representations such as BOW and TF-IDF could be avoided (Li et al., 2020). A deep learning model is a network that transforms or extracts features using multiple nonlinear processing units arranged in multiple layers. A recursive neural network is a deep learning network that recursively applies the same set of weights on an structured input to produce a structured prediction or a vector representation over variable-size input (Minaee et al., 2021). The Recursive Neural Network is one of the first deep learning models used for the text classification task (sentiment analysis) (Socher et al., 2011, 2012).

2.3.5 Feed Forward Neural Network

A feed forward neural network is a simple deep learning model that has achieved high accuracy on many text classification tasks compared to traditional models. It processes a text as a bag of words. For each word, they learn word embeddings such as word2vec or GloVe and combine these embeddings as the representation of a text. This is passed through one or more feed-forward layers and an output layer (Minaee et al., 2021). Paragraph Vector (Le & Mikolov, 2014) is an unsupervised algorithm similar to CBOW which includes paragraph information and uses a feed forward neural network architecture for sentiment analysis.

2.3.6 Recurrent Neural Network

A Recurrent Neural Network (RNN) (Elman, 1990) is a deep learning model where connections between nodes can not only feed forward, but also feed backward. They can process sequences of varying length and have a internal (memory) state. Texts can be processed as a sequence of words, thereby capturing word dependencies and text structures (Minaee et al., 2021). RNNs have been implemented effectively for text classification (P. Liu et al., 2016), but can be subject to the vanishing gradient problem and biased toward latter words compared to earlier words.

The Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) addresses the vanishing gradient problem and is designed to better capture long term dependencies. Adaptations of the LSTM have been proposed for text classification tasks. Tai et al. (2015) and Zhu et al. (2015) proposed a tree-structured LSTM to exploit syntactic properties present in natural language. LSTMs have also been embedded with text regions of variable sizes (Johnson & Zhang, 2016).

2.3.7 Convolutional Neural Network

A Convolutional Neural Network (CNN) is an adaptation of the feed forward neural network inspired by biological visual pattern recognition (Fukushima & Miyake, 1982). It has less connections compared to the feed forward network and takes advantage of patterns across space using convolution and pooling operations. CNNs have proven succesfull on many natural language processing tasks (Collobert & Weston, 2008), including text classification (Kalchbrenner et al., 2014; Johnson & Zhang, 2014). Y. Zhang & Wallace (2015) conduct a sensitivity analysis on one-layer CNNs to explore the effect of a model's architecture on its performance for sentence classification. They find that both word2vec and GloVe input vectors improve sentence level classification performance.

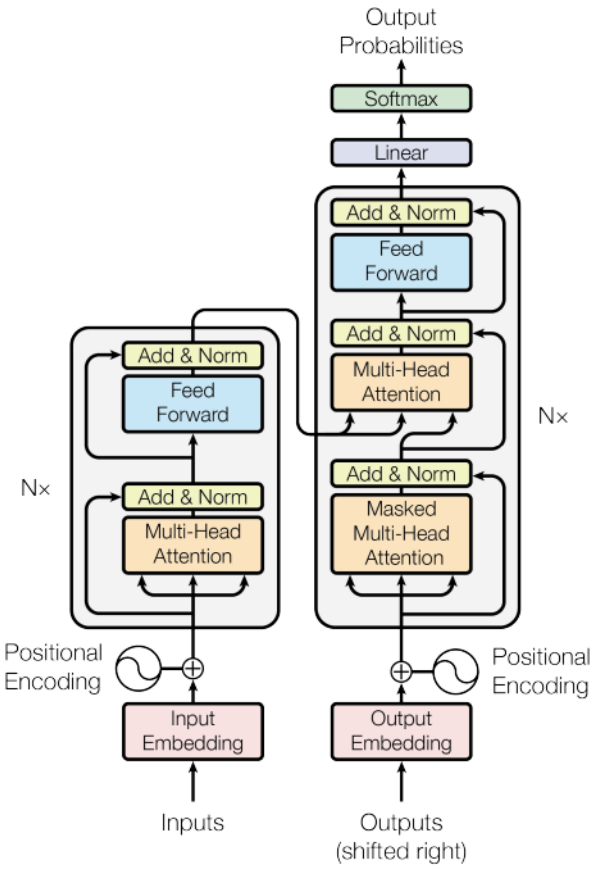


Figure 2.1: Transformer architecture as presented by Vaswani et al. (2017).

2.3.8 Attention

Attention-based methods are successful in text classification. It improves performance by weighting specific inputs differently and increases interpretability of text classification (Li et al., 2020). The Hierarchical Attention Networks (HAN) (Z. Yang et al., 2016) has a hierarchical structure of documents and is the first to apply the attention mechanism to document classification. Zhou et al. (2016) use an attention-based LSTM for cross-lingual sentiment classification.

2.3.8.1 Transformers

Vaswani et al. (2017) proposed the Transformer, a network architecture solely based on self-attention mechanisms, for sequence transduction. The model consists of a stack of encoders and a stack of decoders. Each encoder layer has two sub-layers. The first is a multi-head self-attention layer that runs through an attention mechanism several times in parallel and weights the significance of the input data differently. The second sub-layer is a fully-connected feed forward network. The decoder layers have the same layers as an encoder layer and a third sub-layer, which performs multi-head attention over the output of the encoder stack. The Transformer architecture from the original paper is presented in Figure 2.1.

Self-attention layers allow for a larger amount of computation to be parallellized as compared to recurrent or convolutional layers. It also uses smaller path lengths within the network, which makes

it easier to learn long-range dependencies. This approach achieved great improvements on two machine translation tasks and inspired multiple Transformer-based deep learning neural network architecture, such as GPT and BERT.

Generative Pre-trained Transformer (GPT) (Radford et al., 2018) model follows Vaswani et al. (2017) and trained a decoder-only transformer with masked self-attention heads. They propose a *pre-training* step to learn a generative LM using a diverse corpus of unlabeled text, followed by a *fine-tuning* step that results in a single model for all downstream tasks. By transforming the task-specific inputs for fine-tuning to a single contiguous sequence of tokens, they require minimal changes to the model's unidirectional (left-to-right) architecture.

BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018) is a pre-trained contextualized word embedding model which utilizes a transformer network. Its model architecture is a multi-layer bidirectional Transformer encoder based on Vaswani et al. (2017)'s implementation. BERT uses a masked language model (MLM) pre-training objective, where some of the input tokens are randomly masked for which the objective is to predict their original ids based on their context. Their framework consist of a *pre-training* step in which the model is trained on unlabeled data for different pre-training tasks and a *fine-tuning* step in which the pre-trained parameters are fine-tuned with labeled task-specific datasets. This approach greatly improved accuracy on tasks with smaller datasets and obtained a new state-of-the-art on eleven NLP tasks (Devlin et al., 2018).

Adhikari et al. (2019) present the first application of BERT on document classification, establishing state-of-the-art results for document classification of four popular datasets. Following Devlin et al. (2018), they introduce a fully-connected layer over the final hidden state corresponding to an input token. To reduce the computational load, they apply knowledge distillation to transfer knowledge from BERT to a BiLSTM with 30 times fewer parameters. This approach achieves a similar accuracy, demonstrating that BERT can be represented into a much simpler model (Adhikari et al., 2019).

C. Sun et al. (2019) explore several ways of fine-tuning BERT for text classification. They propose a general solution to fine-tune the pre-trained BERT model with three steps: (1) further pre-training BERT on within-task or in-domain data as this improves performance significantly, (2) optional fine-tuning BERT with multi-task learning as this also improves performance, but less than further pre-training and (3) fine-tune BERT on the target task. They also suggest to use an appropriate layer-wise decreasing learning rate to overcome the catastrophic forgetting problem in which the pre-trained knowledge is erased during learning of new knowledge (C. Sun et al., 2019).

2.4 Knowledge representation

This section starts with introducing the knowledge graph (KG) as a structured representation of knowledge in Section 2.4.1. Then it covers research on representing KGs in low-dimensions while preserving important information or knowledge graph embeddings (KGEs) in Section 2.4.2. It touches on studies that utilize KGEs to augment LMs in Section 2.4.3, as this approach can be beneficial to the model's performance on the dual-use classification task. Finally, it provides a brief overview of studies that ask whether BERT can function as a KG and what is knowledge is present in BERT.

2.4.1 Knowledge Graphs

A KG is a structured representation of facts, consisting of entities, relationships, and semantic descriptions, usually extracted from trusted sources and manually corrected by human editors. As

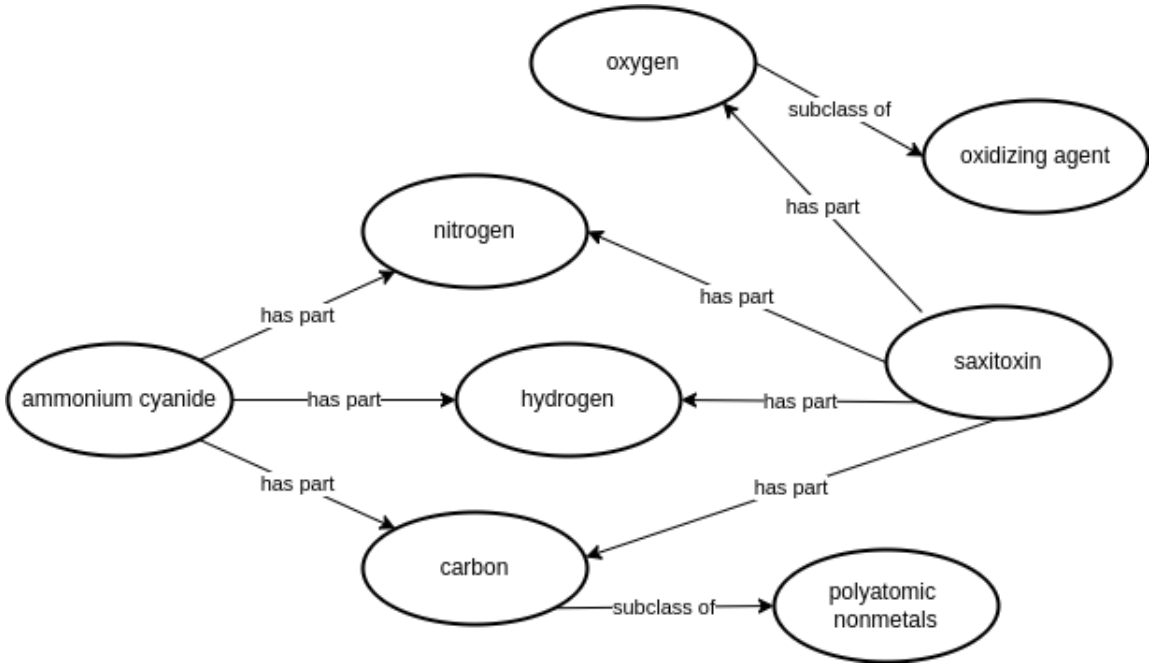


Figure 2.2: An example knowledge graph with entities and relations.

such, KGs provide alternate sources of information compared to LMs (Ji et al., 2021). It is a multi-relational graph composed of entities (nodes) and relations (edges). Each edge is represented as a triple of the form (*head, relation, tail*) (Q. Wang et al., 2017), e.g. '(ammonium cyanide, has part, hydrogen)'. Studies often refer to a 'knowledge base' instead of 'knowledge graph'. There is a minor difference between the two (Ji et al., 2021), but for simplicity this difference is ignored in this research and the terms are used interchangeably. An example KG is shown in Figure 2.2.

The idea of graphical knowledge representation dates back to 1956 as the concept of semantic net (Richens, 1956), but since its use by Google in 2012 (Singhal, 2012) it gained popularity (Ji et al., 2021). They use it to identify and disambiguate entities in text, to enrich search results with semantically structured summaries, and to provide links to related entities in exploratory search (Zou, 2020). Many open knowledge bases are available, such as WordNet (Miller, 1995), DBpedia (Auer et al., 2007), YAGO (Suchanek et al., 2007), Freebase (Bollacker et al., 2008) and Wikidata (Vrandečić & Krötzsch, 2014). KGs have been succesfully applied to applications such as semantic parsing, named entity disambiguation, information extraction and question answering (Q. Wang et al., 2017).

2.4.2 Knowledge Graph Embeddings

KG triplets can effectively represent knowledge, but their symbolic nature makes KGs difficult to manipulate (Q. Wang et al., 2017).

Recently, many studies have focused on mapping entities and relations into low-dimensional vectors while capturing their semantic meanings (Ji et al., 2021): knowledge graph embeddings (KGEs). These KGEs preserve the inherent structure of the KGs, encode rich information about them and are widely utilized by downstream applications (Z. Zhang et al., 2021). This sections describes several important algorithms to create KGEs, categorized by their type of encoding.

2.4.2.1 Translational distance models

Translational distance models are a embedding technique that exploit distance-based scoring functions. *TransE* (Bordes et al., 2013) is one of the most widely used translational distance models. Given a triple *(head, relation, tail)* or *(h, r, t)*, the relation represents a translation vector $\mathbf{r}$ so that the embedding entities $\mathbf{h}$ and $\mathbf{t}$ can follow $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ (Q. Wang et al., 2017). This is based on the assumption that similar entities are likely to have similar relational characteristics and therefore need similar embeddings. There have been multiple extensions on *TransE*.

These translational models have some limitations. They are theoretically inexpressive and cannot capture simple sets of logical rules such as relational hierarchy or inference patterns. To address these limitations, Abboud et al. (2020) propose *BoxE*, which embeds entities as points, and relations as a set of hyper-rectangles, which spatially characterize basic logical properties. It is evaluated for the knowledge graph completion task across multiple datasets and achieves state-of-the-art results.

2.4.2.2 Tensor factorization based models

Tensor factorization based models assume the score of a triple *(h, r, t)* representing semantic similarity can be factorized into several tensors (Z. Zhang et al., 2021). Tensors are multidimensional arrays of numerical values and therefore generalize matrices to multiple dimensions (Rabanser et al., 2017). A general principle of tensor factorization can be denoted as $f_{(h,r,t)} \approx \mathbf{h}^\top \mathbf{M}_r \mathbf{t}$, where $f_{(h,r,t)}$ denotes the score of triple *(h, r, t)* and $\mathbf{M}_r$ is the mapping matrix for relations (Ji et al., 2021).

RESCAL (Nickel et al., 2011) represents each relation as a square matrix. *DistMult* (B. Yang et al., 2014) simplifies *RESCAL* by only using diagonal matrices to represent relations ($\mathbf{M}_r$ is a diagonal matrix). *ComplEx* (Trouillon et al., 2016) extends *DistMult* by introducing complex-valued embeddings instead of real-values embeddings to better model asymmetric relations.

Holographic Embeddings (*HolE*) (Nickel et al., 2016) combine the expressive power of *RESCAL* with the efficiency of *DistMult*. It is related to holographic models of associative memory where circular correlation is used to create compositional representations. Circular correlation can be interpreted as a compression of the tensor product: in calculating the correlation each component corresponds to the sum of a fixed partition of pairwise interactions in the tensor. This can be effective since it allows to share weights for semantically similar interactions. *HolE* can capture rich interactions of a KG, while it remains efficient to compute and easy to train (Nickel et al., 2016). *HolE* outperformed earlier models on the link prediction and relational learning task.

Balažević et al. (2019) propose *TuckER*, a linear model based on Tucker decomposition of the binary tensor representation of knowledge graph triples. Each element in the tensor corresponds to a triple. Tucker decomposition factorizes this tensor into a core tensor multiplied by a matrix along each mode. Rows of the matrices contain entity and relation embeddings, while entries of the core tensor determine the level of interaction between them. *TuckER* outperforms previous state-of-the-art models across standard link prediction datasets, acting as a strong baseline for more elaborate models (Balažević et al., 2019).

DistMA (Shi & Xiao, 2019) addresses the unsuitability of previous models to encode multi-mapping relations such as 1-N, N-1 and N-N relations by defining dot product-based functions over embeddings. It jointly embeds entities and relations from different KGs into a unified embedding space for cross-lingual entity alignment.

2.4.2.3 Neural network based models

Another approach to build KGE is with neural network based models. These approaches have yielded remarkable predictive performance in recent studies (Ji et al., 2021). *ConvE* (Dettmers et al., 2018) applies 2D convolution directly on embeddings, inducing spatial structure in embedding space. *ConvKB* (Nguyen et al., 2017) employs a CNN so that it can capture global relationships and transitional characteristics between entities and relations in KGs. L. Guo et al. (2019) learn to exploit long-term relational dependencies in KGs with recurrent skipping networks, which integrate RNNs with residual learning.

With *CoKE*, Q. Wang et al. (2019) employs transformers to learn contextualized entity and relation embeddings from sequences of edges and paths. *KG-BERT* (Yao et al., 2019) treats triplets in KGs as textual sequences and fine-tunes BERT (Devlin et al., 2018) on these sequences as a sequence classification problem. As in *CoKE*, *KG-BERT* can learn context-aware text embeddings with rich language information via pre-trained LMs.

2.4.2.4 Reinforcement learning based models

Z. Zhang et al. (2021) propose a general multi-task reinforcement learning framework for KGE that considers noise and knowledge conflicts within the KG. Their framework chooses high-quality knowledge triplets, while filtering out noisy ones. This approach is able to enhance existing KGE models of all three previously discussed encoding types.

2.4.3 Improving LMs through KGs

Recently, many studies have tried to augment LMs with KGs in order to integrate the advantages of KGs. A popular approach is to integrate the advantages of KGs with LMs through their embeddings. Z. Zhang et al. (2019) achieve significant improvements on knowledge-driven tasks by enriching pre-trained LMs with embedding information retrieved from KGs. Named entities are retrieved from text and linked to their corresponding entities in Wikidata. The graph structure of the KG is encoded with knowledge embeddings, which are taken as input for their enhanced language representation model (ERNIE). As in BERT, ERNIE uses the MLM as pre-training objective. In addition to this objective, named entity alignments in the input text are masked for which the objective is to predict their original entities to complete alignments. As a result of these objective functions, ERNIE can exploit lexical, syntactic and knowledge information simultaneously.

Ostendorff et al. (2019) also enrich their text representations with KG embeddings that are based on Wikidata. They improve BERT on a classification task of books using short descriptive texts and additional metadata such as title, author(s), publishing date, etc. Automatically generated graph embeddings are utilized as author representations. These are derived by matching the author names to the corresponding Wikidata items.

Peters et al. (2019) propose a method to embed knowledge bases into BERT. Their approach learns entity linkers with self-supervision on unlabeled data, which results in more generally transferable representations that can be used to improve downstream tasks. For *KnowBert*, they integrate WordNet and a subset of Wikipedia into BERT. In KnowBert-Wiki the nodes in the KG are Wikipedia page titles, not Wikidata relations. Actually, early experiments by the authors with embeddings derived from Wikidata did not improve results.

Instead of directly injecting knowledge into BERT, R. Wang et al. (2020) propose *K-Adapter*, a method to inject knowledge into different neural models individually while keeping the original pre-trained model frozen. This addresses the catastrophic forgetting which occurs when injecting knowledge directly.

An approach similar to Z. Zhang et al. (2019) and Peters et al. (2019) is to embed knowledge into BERT by injecting factual knowledge about entities. Poerner et al. (2020) align Wikipedia2Vec entity vectors with BERT's native wordpiece vector space and use the aligned entity vectors as if they were wordpiece vectors to create an entity-enhanced version of BERT.

In KEPLER, X. Wang et al. (2021) encode textual entity descriptions with a pre-trained LM as their embeddings, and then jointly optimize the knowledge embeddings and masked language modeling objectives to integrate factual knowledge. By injecting the sentences that form the basis for their K-BERT model with KG triplets, W. Liu et al. (2020) give the model access to specific domain knowledge. K-BERT works without pre-training as it combines the pre-trained embeddings of BERT directly with a KG.

Agarwal et al. (2020) train a T5 on linearized Wikidata triplets to generate natural language versions of the entire Wikidata KG. Training data for the T5 model TeKGen is created by aligning Wikidata triplets to Wikipedia texts. The TeKGen model is then utilized to build a synthetic corpus that captures the KG in natural language sentences. The resulting KeLM corpus can be used to augment existing pre-training corpora with relational Wikidata knowledge, enabling the integration into the pre-training of language models without architectural changes. It offers ∼15M synthetically generated sentences and improves over previous approaches where KG benchmark datasets are primarily domain specific.

In their recent paper, Safavi & Koutra (2021) categorize recent progress in KG-augmented LMs according to the level of knowledge base supervision provided to the LM. LMs can be build using word-level supervision, where knowledge can be extracted and utilized by cloze prompting and statement scoring. In cloze prompting, triplets are converted to natural language assertions where the token(s) corresponding to the object entity are held out. A LM then predicts the probabilities of candidate tokens for the empty slots. In statement scoring a LM is fed natural language statements corresponding to triplets to predict a score.

For entity-level supervision, knowledge is acquired by focusing on entities. Research focusing on entity-level supervision can be ordered by the amount of supervision it uses. This varies from modeling entities without linking to the KB's entities to training a LM where entity information is embedded at earlier stages of text encoding. Supervision can also be implemented on the level of relations in a KB. Relation-level supervision, contains methods that utilize knowledge base triplets or paths to acquire knowledge. These triplets and paths can be treated as a natural language, posing relations as templated assertions or linearized knowledge base triplets. Other relation-level supervision methods learn distinct embeddings for relations types.

2.4.3.1 BERT as Knowledge Graph

Many recent studies have focused on what knowledge is present in BERT and ask whether it can function as a KG for interpretation and inference over facts. Jiang et al. (2020) investigate what knowledge is present in BERT by using prompts. They propose mining-based and paraphrasing-based methods to automatically generate these prompts use to retrieve factual knowledge from LMs. As their prompt increase knowledge retrieval, they report that LMs are more knowledgeable than initially indicated by previous results.

Rogers et al. (2020) provide an overview of studies on what is driving BERT's performance. They categorize what knowledge is encoded in BERT weights in *syntactic knowledge*, *semantic knowledge* and *world knowledge*. Regarding syntactic knowledge, BERT learns some syntactic information (Z. Wu et al., 2020) and its embeddings encode information about parts of speech, syntactic chunks and roles (Tenney et al., 2019). Research also suggests that BERT has some knowledge of semantic roles (Ettinger, 2020) and that it encodes information about entity types (Tenney et al., 2019).

Probing BERT can give an indication of world knowledge present in the model. By doing so, Ettinger (2020) show that it struggles with challenging inference and role-based event prediction. Petroni et al. (2019) study whether LMs are capable of encoding knowledge in a way similar to KGs by comparing their performance on knowledge-driven question answering tasks. They find that a pre-trained BERT model is competitive with KG-based models for some relation types.

2.4.3.2 Criticism

Instead of probing general knowledge, Sung et al. (2021) test if it is possible to use BERT (and two biomedical variants of BERT) to retrieve biomedical expert knowledge. They report these models do not seem capable to be used as domain-specific knowledge bases. Chalkidis et al. (2020), on the other hand, focus on the legal domain and improve BERT on several tasks within this domain.

Not all research applauds the potential of LMs as knowledge bases. Cao et al. (2021) explore the predicting mechanisms of language models by studying three types of probing: prompts, cases and contexts. They find that previous performance is mainly thanks to the biased prompts and that incorporating external contexts improve predictions on knowledge-driven tasks mainly a result of entity type guidance and answer leakage.

2.5 Evaluation metrics

There are two different imbalanced datasets used in this research: a multi-class dataset (one label per description) and a multi-label dataset (multiple labels per description possible). This sections briefly discusses evaluation metrics for such classification problems.

The most popular metric for the multi-class classification problem is the percentage a classifier predicts correctly, or accuracy (Ghanem et al., 2010). For imbalanced datasets accuracy tends to be overwhelmed by the majority class and ignore the minority class (X. Guo et al., 2008). Common metrics used to account for the imbalance problem are the area under the curve (AUC) of the receiver operating characteristics (ROC) and the the F1-score.

The AUC curve is a measure of the discriminability of a pair of classes (Fawcett, 2006) and can be calculated on the confusion matrix directly (X. Guo et al., 2008). The F1-score is the weighted average between the precision and recall and can be calculated from these metrics. For both the the multi-class (Grandini et al., 2020) and multi-label (Tsoumakas et al., 2006) classification problem, calculating the F1-score and the AUC involves calculating metrics for each class and averaging over them. There are two common averaging approaches: macro- and micro-averaging. In the macro approach all the classes have the same weight in the average, so that there is no distinction between highly and poorly populated classes (Grandini et al., 2020). In the micro approach possible differences between classes are not corrected for: all units are considered together taking possible class differences into account. The micro F1-score boils down to the accuracy formula (Grandini et al., 2020).

Another common evaluation metric for multi-class classification problems is cross entropy. Cross entropy loss is calculated directly on the labels and the predictions and only takes into account the probability of the correct class. This can lead to issues when a label is misclassified, but it is still used thanks to its fast calculation (Grandini et al., 2020). For the multi-label classification problem, binary cross entropy is often used instead (T. Wu et al., 2020), as the probability for each class do not necessarily add up to 1.

Chapter 3

Data and Methodology

The goal of this research is to investigate whether a contextual LM can be used to identify dual-use goods and whether relational dual-use knowledge can improve such LM. The identification of dual-use goods will be treated as a text classification task, where the aim is to assign a label (referring to the dual-use category) to a short description of the good.

Section 3.1 explains the taxonomy of labels and describes the construction of both datasets. The *dual-use goods licenses dataset* is constructed from multiple sources and consists of clean, short texts describing a good and its corresponding label. This dataset is used to test a model's performance on the dual-use classification task, reducing noise in the descriptions as much as possible. The *bill of lading dataset* consists of actual, uncleaned good descriptions and their corresponding label. This dataset is used to test the performance of a model on the dual-use goods identification task on empirical data.

In this research BERT will be augmented with relational dual-use knowledge in order to test whether this improves its performance on the dual-use identification task. Two methods for augmenting BERT with relational dual-use knowledge are explored: by augmenting BERT with KGEs and by further pre-training BERT on relevant synthetic sentences. Section 3.2 introduces the knowledge sources that are used to augment BERT with relational knowledge for both approaches. Section 3.3 focuses specifically on the KGE approach and explains how KGEs are created.

Section 3.4 presents the architectures of all models. First a baseline model is introduced to compare the other models to and to measure if they can improve on the dual-use identification task. This is followed by three logistic regression models and four BERT models, varying in their use of relational dual-use knowledge. A schematic overview of the last BERT model, which uses both methods for augmenting with relational dual-use knowledge, is provided in Figure 3.1 for the *dual-use licenses dataset*. This provides a summary of all data sources used with references to the relevant sections. Finally, this chapter discusses the metrics by which models are evaluated in Section 3.5.

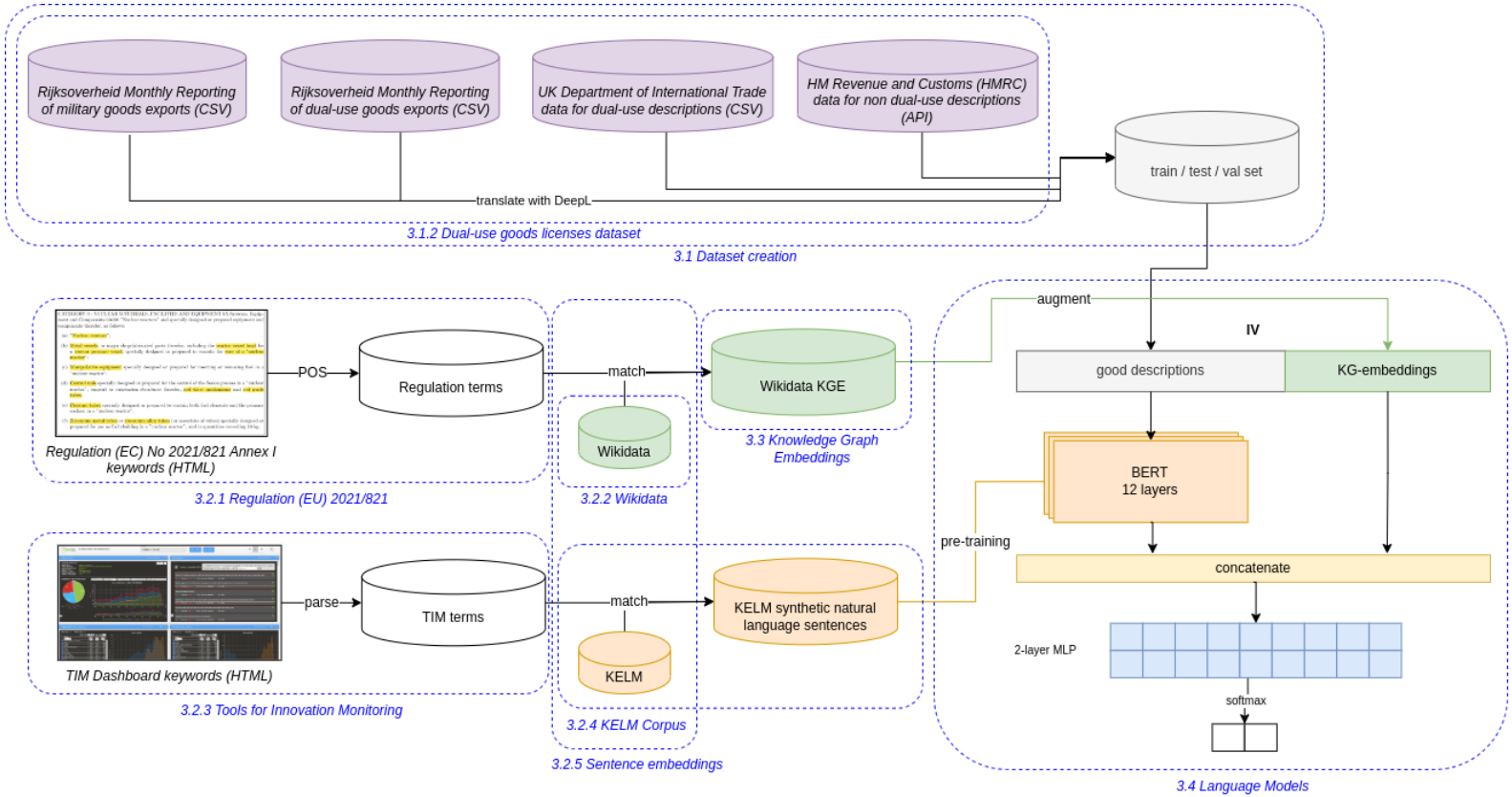


Figure 3.1: A schematic overview of all data sources for IV when using the *dual-use licenses dataset*. In-domain pre-trained BERT. This implementation is further pre-trained (orange) and augmented with KGEs (green). The blue texts indicate the paragraph or section which discusses the area covered by the blue dotted line. For the *bill of lading dataset*, the dataset (purple) would be replaced by the dataset described in Section 3.1.3.

3.1 Datasets creation

This section describes the creation of both datasets. It starts by explaining the taxonomy of labels in Section 3.1.1. Section 3.1.2 describes the creation of the *dual-use goods licences dataset* and Section 3.1.3 describes the construction of the *bill of lading (BoL) dataset*. An overview of all data sources is provided in Table 3.1.

3.1.1 Taxonomy

The labels used in this thesis are based on Regulation (EU) 2021/821 (Council of European Union, 2021) and the Common Military List of the EU (Council of European Union, 2020). One additional category (X) is added for non dual-use (and non-military) goods. This results in the following 12 categories:

0. *Nuclear materials, facilities and equipment*
1. *Special materials and related equipment*
2. *Materials processing*
3. *Electronics*
4. *Computers*
5. *Telecommunications and "information security"*
6. *Sensors and lasers*
7. *Navigation and avionics*
8. *Marine*
9. *Aerospace and propulsion*

ML. *Military goods*

X. *Non dual-use goods*

Control entries in Regulation (EU) 2021/821 are hierarchical and of the form *0A000*, where the first character refers to category 0 to 9 in the list above. The second character is a letter ranging from *A* to *E* which indicate the type of good (e.g. *A* describes *Systems, Equipment and Components*, while *E* describes *Software*). Additional numbers to the right indicate specific paragraphs in the Regulation. For example, the entry *6A001* refers to the first entry under *Systems, Equipment and Components* in category '6. Sensors and lasers'. There can be additional characters added to the right to refer to more specific paragraphs. In this thesis only the top-level categories (0 to 9) will be used, as categories of length two (e.g. *0A, 0B*) result in categories that have little or no entries for the dual-use licenses dataset.

The Common Military List of the European Union uses 22 categories to identify military goods. These are labeled starting with *ML*, followed by their category and their optional paragraph. For example *ML1* refers to "Smooth-bore weapons with a calibre of less than 20 mm, other arms and automatic weapons with a calibre of 12,7 mm (calibre 0,50 inches) or less and accessories". In this thesis the distinction between these 22 categories is ignored: all military goods are labeled as 'ML. Military goods'. This label will not be used in the BoL dataset, as this data is unavailable.

3.1.2 Dual-use goods licences dataset

The dual-use goods licenses dataset consists of clean good descriptions and their corresponding label (including 'ML. Military goods'). This dataset is used to test the performance of a model, while reducing noise in the descriptions as much as possible. The dual-use and military goods descriptions and their corresponding labels are retrieved from two different sources: the UK Department of

International Trade and the Dutch Ministry of Foreign Affairs. The non dual-use goods descriptions used in this dataset are retrieved from Her Majesty's Revenue and Customs (HMRC) trade statistics (see Table 3.1). This section describes the construction of this dataset, categorized according to its source.

3.1.2.1 UK Department of International Trade licenses

The strategic military and dual-use items that require export authorisation in the United Kingdom are summarized in the UK Strategic Export Control Lists, also known as the consolidated list. This list is compiled of various pieces of international legislation which set out what types of goods are controlled (UK Department for International Trade, 2021c), such as Annex I of EU Regulation No. 428/2009. All the UK licence applications are assessed against the consolidated list's licensing criteria. The UK Department of International Trade publishes decisions on these export licensing applications. The published licensing decisions include a case summary and their control entry.

Case summaries are text descriptions used to classify goods on licences. They are standardised descriptions to explain what the goods are without giving away commercially sensitive details about the goods (UK Department for International Trade, 2021b). These are generally short descriptions, such as "magnetometers", "towed hydrophone arrays" or "marine position fixing equipment".

Control entries are the codes assigned to each good on the consolidated list of goods that require export authorisation. It is possible for a license to have multiple dual-use entries or to be both military and dual-use. The control entry of a good on the dual-use list starts with a number between 0 and 9, which reflects the corresponding EU regulation category (as explained in Section 3.1.1). Goods that are on the military list start with *ML* and reflect their categorization on the military list (see Section 3.1.1). Other control entries, such as nationally controlled items indicated by starting with *PL*, are also provided but are not used in this thesis.

Data is available in an accessible format for 2011-2021 from the UK Reports and Statistics Home (UK Department for International Trade, 2021a). A custom request for cases for this time period resulted in a csv file with application type, outcome, number of licenses and the case summary for 52.362 records. Each row can have multiple case summaries, which are then separated by a comma.

Descriptions are cleaned from parts of text that refer to regulation categories directly (e.g. "all items specified by: 5A002") and from indicators that refer to the amount of licenses given (e.g. for "machine guns (324)", the digits and brackets are removed). Descriptions that are longer than 200 characters after splitting are removed as these are descriptions referring to specific control entries (e.g. *6A001*). Spaces at the beginning and end of each description are removed and every character is converted to lowercase.

For records that contain multiple case summaries and control entries of different categories, it is impossible to determine what case summary refers to what control entry as they are in alphabetic order. Therefore, these records are excluded. If a sentence starts with "promoting the supply of", this part is removed from the sentence (e.g. "promoting the supply of body armour" becomes "body armour") as this indirectly refers to military or dual-use goods and might result in answer leakage. Duplicate case summary-control entry combinations are removed.

The resulting set retrieved from the UK Department of International Trade consists of 3.259 unique description-label pairs. Most labels are military goods with 2.391 records. The least common category is '4. Computers' with only 18 records. A few descriptions have multiple labels, as there are 3.229 unique descriptions in this set.

Table 3.1: Overview of used data sources.

Source	**Category**	**Used for**	**Number of records used**
UK Department of International Trade	Licenses dataset	Dual-use and military goods descriptions and labels[1]	3.259 description-label pairs
Rijksoverheid Monthly Reporting of dual-use goods exports	Licenses dataset	Dual-use goods descriptions and labels[1]	2.155 description-label pairs
Rijksoverheid Monthly Reporting of military goods exports	Licenses dataset	Military goods descriptions and labels[1,2]	6.026 descriptions
HM Revenue and Customs	Licenses dataset	Non dual-use goods descriptions and labels[1]	4.064 descriptions
Zauba	BoL dataset	Dual-use and non dual-use goods BoL descriptions and CN codes	626,032 BoL descriptions
Correlation list between TARIC and Dual-use Annex of Regulation (EU) 2021/821	BoL dataset	Matching CN codes to dual-use control entries	6.251 CN-control entry combinations
Regulation (EU) 2021/821	Knowledge source for KGE-BERT	Dual-use terms are retrieved and matched to Wikidata to retrieve relevant triplets	6.724 terms
Wikidata	Knowledge source for KGE-BERT	Retrieving relevant triplets to create KGEs	5.605.994 triplets[3]
Tools for Innovation Monitoring (TIM) Dual-Use	Knowledge source for Pre-Trained BERT	Dual-use terms are retrieved and matched to KELM to retrieve relevant synthetic sentences	9.893 terms
Knowledge Enhanced Language Model (KELM) Corpus	Knowledge source for Pre-Trained BERT	Retrieving relevant synthetic sentences for further pretraining	184.422 sentences

(1) Descriptions from the trainset (in contrast to validation- and testset) in the licenses dataset are optionally used as knowledge source for KGE-BERT. (2) Military goods descriptions not used in the dataset due to undersampling are optionally used as knowledge source for KGE-BERT. (3) Not all triplets are used for creating KGEs (see Section 3.3.2).

3.1.2.2 Dutch Ministry of Foreign Affairs licenses

As a member of the European Union, the Netherlands is subject to Regulation (EU) 2021/821. In the Netherlands a license is required for the export from the European Union of dual-use goods listed in Annex I of the Regulation (EU) 2021/821. For military goods, a license is required for the export and transfer from the Netherlands (Rijksoverheid, 2022a). The Ministry of Foreign Affairs (BZ) is in charge of controlling exports, transit and imports of strategic goods. It has the legal responsibility for the decisions on license applications. Each month, BZ publishes an overview of the issued licenses. It publishes reports on licenses for dual-use goods (Rijksoverheid, 2022b) and military goods (Rijksoverheid, 2022c) separately. The control entries follow the taxonomy of Regulation (EU) 2021/821 and the military list (see Section 3.1.1).

Both reports are available in csv format for 2004-2022. Every record has one or more control entry. The military goods report contains Dutch descriptions of the good. Every description is translated using *DeepI* (DeepL, 2022). Descriptions with less then 4 characters are removed, as these typically contain control entries instead of good descriptions. Special characters present due to conversion of files are removed, as well as wrong entries (e.g. where the control entry is filled in the descriptions column). This results in 6.026 unique military goods descriptions.

The dual-use goods report contains descriptions of the good and its specified end use in additions to its control entries. If the specified end use starts with "*Tbv*" (short for *for the purpose of* in Dutch) it is simply combined with the description. If this is not the case, the word "*voor*" ("*for*" in English) is added in between before combining the description and its end use. Control entries not starting with 0-9 are removed. Descriptions starting with a digit are removed, as these usually are faulty descriptions which contain control entries. Duplicate control entry-good description combinations are removed. This results in 2.155 unique combinations. Category 6 and 2 are most common with 513 and 512 occurrences, while category 4 only has 5 occurrences.

3.1.2.3 Her Majesty's Revenue and Customs trade statistics

HMRC publishes monthly overseas trade statistics (HM Revenue and Customs, 2021). These data provide statistics on UK import and export since 2016. This includes descriptions of the good and their commodity codes using EU's Combined Nomenclature (CN) (European Commission, 2021a). The CN classifies goods at 8 digit level. The first 6 digits are based on the Harmonized System (HS), a global system for classifying goods developed by the World Customs Organisation. The HS is organised into sections divided into chapters (2 digit codes or HS-2), headings (4 digit codes or HS-4) and subheadings (6 digit codes or HS-6). The CN expands the HS subheading code by 2 further digits (HM Revenue and Customs, 2021).

For this dataset the HS-6 code and their descriptions are used, as these resemble the dual-use and military descriptions most. Dual-use and military descriptions are filtered by using their HS-4 code. For example, all HS-6 descriptions corresponding to HS-4 code *8401* ("Nuclear reactors; fuel elements (cartridges), non-irradiated, for nuclear reactors; machinery and apparatus for isotopic separation") are excluded. Descriptions for 64 HS-4 codes are excluded, which results in 324 excluded descriptions. A full list of filtered HS-4 codes and other excluded descriptions are in the Appendix.

Descriptions that are present in the dual-use descriptions are excluded. If a description contains a comma, "excl.", "incl." or "; salts thereof", only the text up the occurrence is taken into account. For example "creosote oils (excl. chemically defined)" becomes "creosote oils" and "toluidines and their derivatives; salts thereof" becomes "toluidines and their derivatives". If a description further contains a semicolon, the description is split into multiple descriptions separated by the semicolon. Thereafter, descriptions that contain "<", ">", "excl.", "incl." or an apostrophe are excluded, as well

as descriptions that are shorter than 5 characters. Duplicate descriptions are removed. This results in 4.064 descriptions that can be used as non dual-use.

3.1.2.4 Train, validation and test set

All descriptions and labels described in previous sections are combined to create the *dual-use goods licenses dataset* which contains dual-use, military and non dual-use goods descriptions and their corresponding labels. Dual-use and military good descriptions are retrieved from the UK Department of International Trade and the Dutch Ministry of Foreign Affairs and are labeled following the taxonomy described in Section 3.1.1. The non dual-use good descriptions are retrieved from HMRC trade statistics and are labeled as 'X. Non dual-use goods'.

The category 'ML. Military goods' is undersampled to 662 records. This means the complete dataset consists of 7.749 clean description-label combinations. The train, validation and test set are split according to a 60%, 20%, 20% stratified split. The dataset remains imbalanced with the 'X. Non dual-use item' category the most frequent (4.064 records) and '4. Computers' the least frequent (23 records).

3.1.3 Bill of lading dataset

A company active in the international logistics sector agreed to share bills of lading for comparison to the dataset described in Section 3.1.2. They confirmed the descriptions in that section are realistic, but often depict the good with too little noise. In practice, descriptions are often more elaborate with each customer having its own textual variations. To help retrieving realistic descriptions, they suggested using the *correlation list between TARIC and Dual-use Annex of Regulation (EU) 2021/821* (European Commision, 2022). The bill of lading (BoL) dataset is the result. The initial examples from the before-mentioned company were only used to validate whether descriptions were realistic, the examples are not used in the BoL dataset.

The BoL dataset consists of actual, uncleaned bill of lading data and their corresponding label. This dataset is used to test the performance of a model on the dual-use goods identification task on empirical data. In contrast to the dataset created in Section 3.1.2, the BoL good descriptions contain more noise and can have multiple labels. This dataset does not use category 'ML. Military goods'.

3.1.3.1 Descriptions and labels

Bill of lading data is retrieved from Zauba, an online database that provides access to search import and export shipment records (Zauba Technologies, 2022), and combined with the *correlation list between TARIC and Dual-use Annex of Regulation (EU) 2021/821* (European Commision, 2022) to retrieve its labels. This correlation list is a lookup table for CN codes and their corresponding dual-use control entry. As the correlation table does not cover military goods, there are no records for this category.

Zauba provides bill of lading data categorized by CN code for shipments with the port of discharge in India up to 2016 (Zauba Technologies, 2022). Descriptions are retrieved for all CN codes in the correlation list, if these exists. The correlation list contains 6.251 CN-control entry combinations. One CN can have multiple control entries, which results in this dataset being multilabel (in contrast to the multiclass licenses dataset).

For example, CN code *28441030* describes "Natural uranium: Worked (Euratom)". This CN code links to control entries *0C001* and *1C236* in the correlation list. Following the taxonomy

discussed in Section 3.1.1, all descriptions categorized as this CN code get both the labels '0. Nuclear materials, facilities and equipment' and '1. Special materials and related equipment'.

All CN codes not in the correlation list are interpreted as referring to non dual-use items and receive the corresponding label. The correlation list matches to 431 unique CN codes. The non dual-use items cover 514 unique CN codes. Descriptions are converted to lowercase. Duplicates descriptions with the same label are removed, while duplicate descriptions with different labels are combined.

As mentioned before, no further cleaning is performed on these descriptions. This way the noise present in the descriptions remains, which means models can be evaluated against actual bill of lading descriptions as compared to the clean descriptions in the licenses dataset. As there is no standard format or guideline for these descriptions, variations are plentiful. The (lowercased) example descriptions "break system for dmrc rs10 hose pipes -ii44729/03916nt-co1.2-de 2te-16-nn90-390" and "steel go (sgs) m- 5, 80mm(raw mat.for mnf.toroidal transformers)" clearly show this.

3.1.3.2 Train, validation and test set

This results in a total of 626,032 unique descriptions. The largest category is 'X. Non dual-use goods' with 244,079 descriptions, while the smallest category is category '7. Navigation and avionics' with 35,190 descriptions. The largest dual-use category is '1. Special materials and related equipment' with 175,569 descriptions. The dataset is split into a train set with 506,032 records and a validation and test set of each 60,000 records.

3.2 Knowledge sources

In this thesis, BERT will be augmented with dual-use knowledge in order to improve its performance on the dual-use classification task with two different methods: by augmenting BERT with KGEs and by further pre-training BERT on relevant synthetic sentences. This section describes the knowledge sources used to improve BERT.

First, this section describes the different knowledge sources that are used to augment BERT with KGEs: Regulation EU No. 2021/821 (Section 3.2.1) and Wikidata (Section 3.2.2). Then it describes the knowledge sources used for further pre-training of BERT, namely the Tools for Innovation Monitoring (Section 3.2.3) and the KeLM Corpus (Section 3.2.4). In Section 3.2.5, the matching mechanism between knowledge sources for both augmentation methods is explained.

3.2.1 Regulation (EU) 2021/821

The most important knowledge source for dual-use identification is Regulation (EC) No. 2021/821. This regulation governs the EU's export control regime (Council of European Union, 2021). Annex I contains a list of dual-use items and covers most of the document with about 400 out of 465 pages. This describes all dual-use goods, materials and technologies divided into categories 0 to 9 provided in Section 3.1.1.

In order to extract the correct dual-use terms from Annex I of this regulation, first each of the subnotes ([a-f] in Figure 3.2) of the Annex will be parsed. This results in a sequence of words for each subnote. Each sequence of words is then fed to *English Universal Part-of-Speech Tagger* (Akbik et al., 2018) in *Flair* (Akbik et al., 2019). As this regulation is a well-structured document, there are some document-specific rules to be exploited to extract the correct terms. For example, if the complete sequence consists of nouns and adjectives only (e.g. Figure 3.2a), it is more likely to be an important dual-use term. The same holds if a noun and adjective span the beginning of

Figure 3.2: Excerpt of Regulation (EC) No 2021/821 Annex I Category 0 Nuclear Materials, Facilities and Equipment. Yellow highlights are term extraction results by combining POS Tagging and document-specific rules.

0A001 "Nuclear reactors" and specially designed or prepared equipment and components therefor, as follows:

(a) "Nuclear reactors";

(b) Metal vessels, or major shop-fabricated parts therefor, including the reactor vessel head for a reactor pressure vessel, specially designed or prepared to contain the core of a "nuclear reactor";

(c) Manipulative equipment specially designed or prepared for inserting or removing fuel in a "nuclear reactor";

(d) Control rods specially designed or prepared for the control of the fission process in a "nuclear reactor", support or suspension structures therefor, rod drive mechanisms and rod guide tubes;

(e) Pressure tubes specially designed or prepared to contain both fuel elements and the primary coolant in a "nuclear reactor";

(f) Zirconium metal tubes or zirconium alloy tubes (or assembles of tubes) specially designed or prepared for use as fuel cladding in a "nuclear reactor", and in quantities exceeding 10 kg;

a sentence. The PoS-tagger is combined with such document-specific rules to extract the correct dual-use term from the regulation document.

Adding more handwritten rules does not always result in retrieving more quality dual-use terms. As more rules are added, both wanted and unwanted terms are retrieved more. Results are evaluated manually. This trade-off means some important terms will not be recognized (e.g. Figure 3.2f "zirconium alloy tubes" is not recognised as an important term), while some are marked as important terms while they are not (e.g. Figure 3.2b "shop-fabricated parts" might not be an important dual-use term). A wrong classification in the PoS-tagger can also result in wrongfully adding or skipping terms.

Subsequent words form one dual-use term (e.g. "zirconium metal tubes") with exceptions where words are separated by a conjunction (e.g. "cylindrical or conical tubes" results in two terms: "cylindrical tubes" and "conical tubes"). Duplicate terms and terms shorter than 3 characters are removed. Terms that start with a digit follow by a space or contain a paragraph code (e.g. "A001") are removed. Terms refer to chapters (e.g. "part II") or other textual artifacts (e.g. "technical note", "above specifications" etc.) are also removed.

This approach results in 6.724 dual-use terms retrieved from Regulation EU No. 2021/821. These are used in combination with Wikidata (Section 3.2.2) to create KGEs. These KGEs are used to augment BERT. The mechanism by which these regulation terms are matched to Wikidata is explained in Section 3.2.5.

3.2.2 Wikidata

Wikidata is an open-source knowledge base focused on items which represent any kind of topic, concept, or object (Vrandečić & Krötzsch, 2014). It contains triplets of the form *(subject, property, object)*. The complete Wikidata dump (version 20220103 1.4TB) is parsed for triplets in which the property is "chemical structure", "made from material", "element symbol", "chemical formula", "subclass of", "has use", "has part", "has effect" or "has quality" to reduce its size and focus on the properties most relevant to dual-use good identification.

Triplets with its object starting with "Wiki" or being "human", "family name", "scholarly article", "review article", "meta-analysis", "scientific journal" or "biographical article" are not taken into account. Triplets with its subject starting with "Category" are also not taken into account. This results in 5.605.994 triplets with 3.340.623 unique items (subject or object). How the regulation terms and the Wikidata triplets are matched is explained in Section 3.2.5.

3.2.3 Tools for Innovation Monitoring

Tools for Innovation Monitoring (TIM) Dual-use (European Commission, 2021c) is an initiative of the EC Competence Centre on Text Mining and Analysis. It maps the dual-use technologies listed in Annex I of Regulation (EU) 428/2009 (the predecessor of 2021/821) based on keywords from the regulation with their scientific and technical synonyms from different types of documents, such as articles, book chapters, conference papers, patents etc. Data is provided in accordance with the categories mentioned in the Section 3.1.1. It also maps emerging technologies not listed in this regulation, but with potential dual-use applications.

In the publicly available dashboards, TIM provides the most common automatic keywords. These keywords are the automatically retrieved keywords from the documents for a specific dual-use category. All keywords will be scraped from the *Global Dual-Use Queries* tab, resulting in 1000 keywords per category. Countries, places, organisation names and other terms irrelevant to dual-use research (i.e. "boxing", "painting", "thereof" etc.) are removed. This results in 9.893 TIM dual-use terms. These terms are used to retrieve relevant synthetic sentences from the KeLM corpus. The KeLM corpus is discussed in Section 3.2.4. How these TIM terms are matched to the KeLM corpus is explained in Section 3.2.5.

3.2.4 KeLM Corpus

The Knowledge Enhanced Language Model (KeLM) Pre-training corpus is a generated synthetic corpus that consists of the entire Wikidata KG as natural text sentences (Agarwal et al., 2020). It offers ∼15M triplets of the form *(subject, relation, object)*, serialized triplets of the from *"<subject> <relation> <object>"* and the generated natural language sentence for the triplets based on Wikidata. This verbalized corpus can be used to integrate KGs and natural text corpora by including it as additional pre-training data (Agarwal et al., 2020).

A subset of the KeLM corpus will be used in order to improve BERT with relational knowledge about dual-use goods. Synthetic sentences are selected based on their proximity to the retrieved TIM dual-use terms. How these sentences are retrieved, is discussed in Section 3.2.5.

3.2.5 Sentence embeddings

The terms extracted from Regulation (EU) 2021/821 are matched to Wikidata in order to retrieve relational dual-use goods data (triplets) with which BERT can be augmented. Similarly, the terms extracted from TIM are matched to the KeLM corpus to retrieve relational dual-use goods data (synthetic natural sentences) for further pretraining BERT. As strictly matching terms results in poor coverage (21% and 16.6% for Wikidata and KeLM respectively), terms are matched using their sentence embeddings.

This means for every term its sentence embedding (a 768 dimensional dense vector) is calculated with *all-mpnet-base-v2* (Reimers & Gurevych, 2019), a sentence-transformer model trained on a large and diverse dataset of over 1 billion training pairs. The relevant term is the term which corresponds to the sentence embedding for which the squared distance to the input sentence embedding is the smallest. If a term can be matched exactly both sentence embeddings will be equal,

which results in the lowest possible squared distance of zero. How this matching mechanism works for both approaches is discussed below.

3.2.5.1 Matching regulation terms to Wikidata

In order to match the 6.724 dual-use terms retrieved from Regulation EU No. 2021/821 to the Wikidata triplets, for every 6.724 dual-use term and for every unique 3.340.623 Wikidata item the sentence embeddings are calculated - which results in the same number of sentence embeddings. Then for every dual-use term, the Wikidata term corresponding to the minimal squared distance between the sentence embeddings is retrieved. Finally, every triplet which contains the retrieved Wikidata term as an item is used as an input to creating knowledge graph embeddings (the creation of KGEs is discussed in Section 3.3).

For example, the regulation term "germanium material" is not found by exactly matching the term to Wikidata items. The sentence embedding for this term is closest to the sentence embedding of the Wikidata term "gallium". Therefore the Wikidata term "gallium" is used and all Wikidata triplets containing the term "gallium" are retrieved. For the 6.724 regulation terms, this results in 10.969 triplets. Section 3.3.2 describes what different sets (the regulation terms is one of these sets) are used to match to Wikidata.

3.2.5.2 Matching TIM dual-use terms to KeLM

Matching the 9.893 TIM dual-use terms to KeLM takes place in a similar way. The sentence embeddings for each TIM term is compared to the sentence embedding of each item (subject or object) of the KeLM triplets. After the closest item is retrieved, all synthetic natural sentences consisting of this item (i.e. that contain this item in one of its triplets) are retrieved.

For example, the TIM term "nuclear waste" is not found by exactly matching the term to the items in the KeLM corpus' triplets. The sentence embedding for this term is closest to the sentence embedding of the sentence embedding of the KeLM triplet item "radioactive waste". This item is used in multiple triplets (e.g. ["dry cask storage", "contains", "radioactive waste"]) and is used to generate the sentence "Dry cask storage contains radioactive waste". All sentences using triplets with the item "radioactive waste" are retrieved and are used to further pretrain BERT (Section 3.4.3.2). In total, 184.422 sentences are retrieved from KeLM by matching TIM terms.

3.3 Knowledge Graph Embeddings

This section describes the algorithms (Section 3.3.1) and the sets of triplets (Section 3.3.2) used to create the KGEs for augmenting BERT.

3.3.1 KGE algorithms

Five algorithms are used to create different KGEs to augment BERT (separately): *TransE* (Bordes et al., 2013), *HolE* (Nickel et al., 2016), *BoxE* (Abboud et al., 2020), *DistMA* (Shi & Xiao, 2019) and *TuckER* (Balažević et al., 2019) . All of the models used produce item-level embeddings, which means a KGE of a given size is returned for for every item in the provided triplet set. Subjects and objects are treated equally: there is no distinction between the embeddings of an item depending on whether it appears as a subject or as an object in a particular triple. Producing item-level KGEs facilitates the use of these embeddings in the chosen approach to augment BERT, as it makes it easy to retrieve embeddings for a given input item.

Table 3.3: Overview of KGE triplet sets.

Name	Description	Number of triplets
X_train only	Train set descriptions.	6.592
Regulation only	Terms retrieved from Regulation EU No. 2021/821	10.969
X_train + Regulation	Train set descriptions and terms retrieved from Regulation EU No. 2021/821	17.561
Regulation + Military	Terms retrieved from Regulation EU No. 2021/821 and the descriptions not used in the trainset by undersampling the 'Military goods' category.	14.063
All 1^{st} order	Train set descriptions, terms retrieved from Regulation EU No. 2021/821 and the undersampled descriptions of the 'Military goods' category.	20.655
All 2^{nd} order	Train set descriptions, terms retrieved from Regulation EU No. 2021/821 and the undersampled descriptions of the 'Military goods' category and all the items one step away from these items. For example, "radio receiver" has two "has part" properties: "antenna" and "loudspeaker". This means all triplets where "antenna" or "loudspeaker" is an object are also taken into account.	49.084

Other important factors for choosing these algorithms are their high efficiency and their relatively low memory use. More recent CNN-based approaches to create KGEs are not used as these are too memory-expensive. All KGEs are created with *PyKEEN* (Python KnowlEdge EmbeddiNgs) (Ali et al., 2021), a Python package designed to train and evaluate reproducible knowledge graph embedding models.

All triplet sets are split in a train (80%), validation (10%) and test set (10%). The standard training loop that uses the stochastic local closed world assumption (Ruffinelli et al., 2019) is used. Scoring functions differ per algorithm. Performance of KGE models is evaluated rank-based: the scoring function produces a plausibility score for triplets in the validation and test set, which is then converted into a ranking.

3.3.2 KGE triplet sets

The algorithms mentioned in Section 3.3.1 require triplets to produce item- and property-level embeddings. All triplets used to create KGEs are based on triplets retrieved from Wikidata. This is a subset of 5.605.994 triplets with only certain properties (see Section 3.2.2). Several sets of triplets are created to find the most useful set for dual-use classification. All triplets are retrieved through matching their sentence embedding to closest sentence embedding of a Wikidata item (see Section 3.2.5). All six sets of triplets used to create KGEs are described in Table 3.3.

3.3.3 Sentence embeddings

Every input sample (good description) in the KGE-augmented BERT models requires a matching mechanism between the input sample and a KGE. As in Section 3.2.5, *all-mpnet-base-v2* (Reimers & Gurevych, 2019) is used to find the closest item and retrieve its embeddings for each sample. For example, the input sample "Sodium hexafluoroaluminate synthetic cryolite" is not found directly in the KGE items. The sentence embedding for this description is closest to the sentence embedding of the KGE item "sodium hexafluoroaluminate". Therefore, the KGE for "sodium hexafluoroaluminate" is retrieved and used to augment BERT for this sample.

Table 3.5: Baseline model

If present in text	Prediction
military	ML. Military goods
image or intensifier	6. Sensors and lasers
industry or valves	2. Materials processing
manufacture	1. Special materials and related equipment
semiconductor	3. Electronics
uranium or nuclear	0. Nuclear materials, facilities and equipment
technology	9. Aerospace and propulsion
security or information	5. Telecommunications and "information security"
navigation	7. Navigation and avionics
submersible	8. Marine
computer	4. Computers

3.4 Language Models

This section describes the different language models and its architectures. It starts with a simple baseline model and continues with different implementations of Logistic Regression and BERT. Models are tested on both the *dual-use licenses dataset* and the *BoL dataset*.

3.4.1 Baseline

The baseline model is a simple model that checks if certain words are present. The words chosen are the one or two most common words for that category in the *dual-use licenses dataset*, excluding stopwords (e.g. “for”, “and”) or words that occur frequently in all categories (e.g. “equipment”). For predictions on the *dual-use licenses dataset*, categories are in order of frequency, except the most common category ‘X. Non dual-use’, which is chosen if none of the words is found. All words used in this baseline model and their corresponding predictions are shown in Table 3.5.

For predictions on the multilabel *BoL dataset*, the order does not matter: if a word is present in the description, the prediction will include the corresponding label. If multiple words are present, multiple labels are returned.

3.4.2 Logistic Regression

This section describes the logistic regressions model as a simple improvement on the baseline model. As the logistic regression models are used to test the usefulness of KGEs in more simple models compared to BERT, three implementations are proposed. For the multilabel dataset, each category is treated separately as a binary classification problem to retrieve separate probabilities per class.

3.4.2.1 Logistic Regression TF-IDF

First, the train set is converted into a matrix of TF-IDF features with unigrams, bigrams and trigrams. English stopwords are removed and terms that occur less then four times are ignored. Intuitively, this model improves on the baseline as it takes the importance of a words regarding the corpus into account.

3.4.2.2 Logistic Regression KGE

Instead of using the TF-IDF features described in the previous section, this implementation uses the matched KGE embeddings only. Input terms are matched to the closest KGE-item with the use of sentence embeddings, as described in Section 3.3.3. If the KGEs contain valuable information for classifying dual-use goods, this model should see be able to learn (i.e. perform the classification task the same or better than the baseline).

3.4.2.3 Logistic Regression TF-IDF and KGE

The last logistic regression implementation is a combination of the previous two implementations. Here, the KGEs are concatenated to the TF-IDF features and used to predict the corresponding label using logistic regression.

3.4.3 BERT

This section describes the different architectures proposed to augment BERT with the dual-use KG described in the previous section. It covers a simple plain BERT model and four different model architectures inspired by different studies. These BERT models are referred to as I to IV and their architectures are represented in Figure 3.3.

3.4.3.1 Plain BERT

Plain BERT is a standard pre-trained BERT model, fine-tuned on the train set of each dataset. One (binary) softmax layer will be added to classify the dual-use goods descriptions. The matched KELM dataset and KGEs will not be used for this model.

BERT's model architecture is a multi-layer bidirectional Transformer encoder (shown in Figure 2.1). Devlin et al. (2018) propose a *pre-training* and a *fine-tuning* step. In the former step the model is trained on unlabeled data using MLM. For the latter step, the pre-trained parameters are adjusted with a labeled task-specific dataset. For the baseline BERT in this thesis, the pre-trained parameters from $BERT_{BASE}$ are used without further pre-training. $BERT_{BASE}$ has 12 layers, hidden size 768 and 12 self-attention heads. This results in 110M parameters. The *fine-tuning* step is performed on the dual-use train set. The simplified architecture of this model is depicted under I in Figure 3.3.

3.4.3.2 In-domain pre-trained BERT

C. Sun et al. (2019) propose three steps to fine-tune the pre-trained BERT model to improve its performance for the classification task: (1) further pre-train BERT on within-task training data or in-domain data; (2) optional fine-tuning BERT with multitask learning if several related tasks are available; (3) fine-tune BERT for the target task. In this thesis, these suggested steps are followed, excluding the optional second step as there are no clear related tasks available. This leaves step (1) and (3).

In general, in-domain pre-training can bring better performance than within task pre-training and specifically for small sentence datasets in-domain pre-training achieves better results (C. Sun et al., 2019). For specialised domains further pre-training might prove beneficial (Chalkidis et al., 2020). As the dual-use classification task is in a specialised domain with short descriptions of goods, in-domain pre-training is expected to achieve better results. The synthetic natural language sentences retrieved from the KELM corpus (as discussed in Section 3.2.5) will be used as in-domain

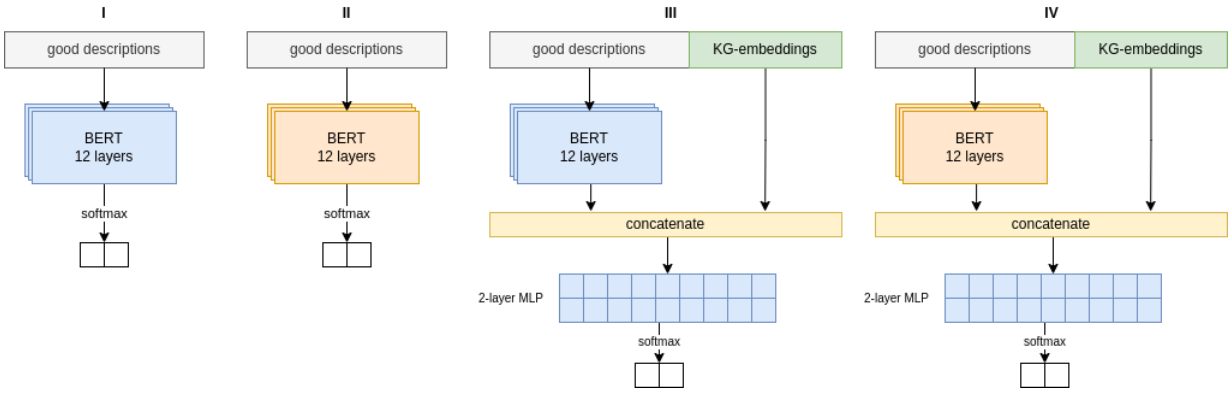


Figure 3.3: Architectures for the proposed LMs. Blue is standard BERT, orange is further pre-trained BERT on in-domain data. Model II follows (C. Sun et al., 2019), model III follows (Ostendorff et al., 2019) and model IV combines these approaches.

data in the first step. As with the first BERT model, the train set will be used for fine-tuning for the target task. Except for the additional pre-training on natural language sentences, there are no adaptations to this model. This means the model's architecture is the same as the architecture of the baseline. This is depicted under II in Figure 3.3.

3.4.3.3 BERT with KGEs

Ostendorff et al. (2019) leverage external knowledge though enriching BERT with knowledge graph embeddings for document classification. Specifically, they include author identity information from Wikidata as dense vector representation to support the genre classification task. Embeddings for each author are retrieved from *PyTorch BigGraph* (Lerer et al., 2019) by matching the author's Wikipedia article to the corresponding Wikidata ID.

In this thesis, the architecture proposed by Ostendorff et al. (2019) is followed as the tasks are comparable (both are text classification tasks). Instead of using the KGEs of the full Wikipedia article, the KGEs will be created with different KGE algorithms on various sets of triplets (as discussed in Section 3.3). For each input description the KGE with the closest sentence-embedding is retrieved. These embeddings will be concatenated to the original pre-trained BERT embeddings and fed to a 2-layer MLP with a softmax activation function. This architecture is represented as Model III in Figure 3.3.

3.4.3.4 In-domain pre-trained BERT with KGEs

This model's architecture is a combination of the architecture in discussed in previous sections. It will be further pre-trained on in-domain data as model II described in Section 3.4.3.2 using synthetic natural language sentences based on TIM data. It will also be augmented with KGE as described in Section 3.4.3.3, using different KGE algorithms on various sets of triplets. This approach is expected to improve on all previous implementations, as it combines all different sources and methods. This is represented as Model IV in Figure 3.3.

3.4.4 Parameter settings

There are many parameters that can influence the performance of the dual-use classification models. Parameter settings are presented in Table 3.7, where single values refer to fixed parameter values

and multiple values refer to grid search parameters (except for the seed). To find the optimal parameter values for each model a grid search is performed on the *dual-use license* validation set.

For the Logistic Regression models using KGEs, the setting with the highest validation macro AUC is chosen as the best setting. The BERT models are run three times with different seeds. The parameter settings with the highest average validation macro AUC over these three runs is chosen as the best setting. For each model, the best setting is evaluated against the test set. For efficiency reasons the validation set of the *BoL dataset* will only used to find the optimal number of epochs. For the other grid search settings, the best settings found on the *dual-use licenses* validation set are used.

3.4.4.1 Difference cutoff

The difference cutoff is a parameter that influences to what extent relevant terms are taken into account. A difference cutoff of 0 means every input term should be exactly matched to the search term (the squared distance between the sentence embeddings of each term is 0). A cutoff of 0.25 implies only KGEs of terms that are very similar are taken into account (the squared distance between the sentence embeddings of each term is smaller than or equal to 0.25). If for a given input term no sentence embedding is found smaller than the cutoff, an embedding of the same size with only zeros is used.

The grid search parameter values for the difference cutoff is based on the distribution of similarities. For the *dual-use licenses dataset*, difference values are smaller as compared to the *BoL dataset*, as descriptions contain less noise. They are about normally distributed with a mean around 0.5, except for a peak at 0, which represents the descriptions that can be exactly matched. The former *dual-use licenses dataset* uses the values presented in Table 3.7, the *BoL dataset* fixes the cutoff to a value equal to 1.

3.5 Evaluation

This section describes the metrics by which the models are evaluated for both the multiclass *dual-use licenses dataset* and the multilabel *bill of lading dataset*.

3.5.1 Evaluation metrics

The evaluation metrics for both datasets have to take into account two important factors. First, the datasets are imbalanced. This might result in metrics that are biased toward the majority class. Second, the dataset have multiple categories, which can make the imbalance more difficult to handle (Ghanem et al., 2010).

For the dual-use classification task, it is important to predict all classes correct regardless of their occurrence in the dataset. Although it can be argued that classes that occur most frequently are more important, this would bias the metrics too much toward the non dual-use item class which accounts for a large part of the test set for each dataset.

This may be at the expense of categories that have just a few samples and are still important to classify correctly, such as category '4. Computers' in the *dual-use licenses dataset*. Besides, although this category is very sparse in this dataset, this does not imply goods of this category also occur infrequently in practice. The sparcity of some classes in the dataset should be seen as an artifact of this dataset, not as a representation of the distribution expected in practice. In fact, for the *BoL dataset* '4. Computers' is not the smallest category.

Table 3.7: Grid search parameters and values for KGE models

	parameters	**values**
Logistic Regression	Minimum DF[1]	4
	Sublinear TF scaling	*True*
	Penalty	*L2*
	Solver	*liblinear*
	n-gram range[2]	(1,3)
KGE	Algorithm	*TransE*, *HolE*, *BoxE*, *DistMA*, *TuckER*
	Epochs	10
	Embedding size[3]	64, 128, 256, 512, 1024
	Triplets set	'X_train only', 'Regulation only', 'X_train + Regulation', 'Regulation + Military', 'all 1st order', 'all 2nd order'
	Seed	2022
BERT	Epochs	7
	Max sentence length	64
	Optimizer	AdamW
	Learning rate	2e-5
	ϵ	1e-8
	Difference cutoff	0, 0.25, 0.5, 0.75
	Seed	2022, 2023, 2024

(1) Minimum document frequency is set to 1,000 for the *BoL dataset*. (2) n-gram range for the *BoL dataset* is (1,1) to reduce memory usage. (3) Embedding size 1024 is not used for the *TuckER* algorithm due to memory usage.

As macro-averaging gives all classes the same weight in the metric, the macro-averaged metrics are considered most important and all models will be evaluated according to the macro AUC and macro F1 score. During training cross entropy will be used as loss function on the multi-class dataset. For the multi-label dataset, binary cross entropy with logits loss is used to account for categories being orthogonal. Micro AUC and micro F1 scores will be presented for the results on the test sets.

Finally, AUC and F1 test scores will be presented for the binary and the partial case for the multiclass *dual-use licenses dataset*. The binary case combines all dual-use categories into one category, which results in a binary classification of dual-use and non dual-use items. The partial case excludes the non dual-use category and focuses on the dual-use and military categories. Confusion matrices are also presented for the *dual-use licenses dataset*.

Chapter 4

Results

This section provides the results of all experiments. For both the *dual-use licenses dataset* (Section 3.1.2) and the *bill of lading dataset* (Section 3.1.3), results are shown for the baseline (Section 3.4.1), logistic regression with TF-IDF (Section 3.4.2.1), logistic regression with KGEs (Section 3.4.2.2), logistic regression with both TF-IDF and KGEs (Section 3.4.2.3), plain BERT (Section 3.4.3.1), pre-trained BERT (Section 3.4.3.2), KGE-BERT (Section 3.4.3.3) and pre-trained KGE-BERT (Section 3.4.3.4).

Table 4.1 shows the best parameter settings found on the *dual-use licenses* validation set. Table 4.3 shows the results of the experiments with these best settings on the *dual-use licenses test set*. Table 4.5 shows the results for the same experiments on the *bill of lading test set*. The F1 scores per category are presented in Table 4.7.

4.1 Validation results

This section provides the results on the *dual-use license validation set*, which was used to find the best parameter settings. It starts with the best settings and continues with other findings regarding these parameters.

4.1.1 Best settings

Table 4.1 shows the best parameter settings found on the *dual-use licenses* validation set. For both logistic regression models which use KGEs, *DistMA* performs best in combination with the 'Regulation only' triplet set. For the logistic regression model using both TF-IDF and KGE, the best embedding size is found to be 512. For the logistic regression model using only KGEs, this is 1024. For all BERT models, the most important grid search parameter is the number of epochs. Given the best BERT epoch, implementations in the grid search have a comparable performance on the *dual-use licenses dataset*. The best number of epochs is found to be 3 for plain BERT, pre-trained BERT and pre-trained KGE-BERT. For KGE-BERT the best number of epochs is 5.

4.1.2 Logistic Regression

For the KGE logistic regression models the results on the validation set indicate a positive relationship between the embedding size and predictive performance. Figure 4.1 (top left) visualises this relationship by averaging the AUC over the different triplet sets per algorithm and embedding size. The smallest embedding size, 64, performs worse compared to the baseline model for all algorithms.

The best setting uses the *DistMA* algorithm with the largest embedding size, 1024, and has a greater validation set AUC as compared to using TF-IDF only.

DistMA outperforms the other KGE algorithms on the *dual-use licenses validation set*. For the logistic regression model using TF-IDF and KGE as inputs, the improvement in performance for *DistMA* diminishes with each increase in embedding size. The only notable improvements in performance for this implementation are for *DistMA* when increasing embedding size from 64 to 128 and for *TuckER* when increasing embedding size from 256 to 512. *DistMA* is the only KGE algorithm that outperforms the TF-IDF only logistic regression.

4.1.3 BERT

Variation in validation AUC between different seeds is similar to the variation between KGE settings, indicating KGE settings such as type of algorithm and embedding size do not influence the performance noticeably. The best settings for the BERT models using KGEs are more arbitrary than a sign of one setting outperforming others on this dataset.

BERT performs better compared to all logistic regression models on the validation set. On the *dual-use licenses dataset* differences between the different implementations of BERT are negligible: the variation in AUC between different implementation is similar to the variation within implementations for different seeds. This can be seen in Figure 4.2. Augmentation with KGEs does not improve the performance of BERT for different embedding sizes, as visualised in Figure 4.1 (bottom left) and 4.1 (bottom right).

4.2 Test results

This section presents the results on the test sets. Table 4.3 and Table 4.5 present the AUC and F1-scores for all experiments on the *dual-use licenses* and *bill of lading test set* respectively. All logistic regression models outperform the baseline. The best logistic regression implementation uses the combination of TF-IDF and KGEs. All BERT models outperform the logistic regression models. Differences in performance between the different implementations of BERT are negligible.

4.2.1 Logistic regression

For the *dual-use licenses* dataset, the logistic regression model with TF-IDF and KGE features has the highest mean test macro AUC (0.842) and test macro F1 (0.706), while the logistic regression model using only TF-IDF has the highest micro AUC (0.906) and the highest micro F1 (0.827). Comparing F1-scores per category, the logistic regression model with TF-IDF and KGE has a higher score for the categories with very few samples (category '4. Computers' and '8. Marine') compared to the TF-IDF only implementation.

For the *bill of lading test set*, the logistic regression model using TF-IDF and KGE improves on both other logistic regressions for both macro and micro metrics. Contrary to the other dataset, this model displays the highest F1-score for category '4. Computers' (0.637). Category '7. Navigation and avionics' is hardest to predict correct for all logistic regression models, with the implementation using only KGEs only having an F1-score of 0.192 and the implementation with using TF-IDF and KGEs having an F1-score of 0.382.

4.2.2 BERT

All BERT models outperform the baseline and the logistic regression models on both datasets. There is no clear difference in performance between the different BERT implementations.

Test macro AUC is around 0.907 (with a minimum of 0.903 to a maximum of 0.910) for the *dual-use license* dataset and around 0.973 (ranging from 0.973 to 0.974) for the *bill of lading* dataset. Test micro AUC is around 0.954 (ranging from 0.953 to 0.955) for the *dual-use license* dataset and around 0.977 (ranging from 0.976 to 0.977) for the *bill of lading* dataset. This means AUC scores for the *bill of lading* dataset are higher compared to the *dual-use licenses* dataset and the difference between micro and macro AUC are larger for the *dual-use licenses* dataset.

Macro F1-scores for the *bill of lading* dataset are higher (0.835 to 0.837) as compared to the *dual-use licenses* dataset (0.807 to 0.826), while the micro F1-scores are higher for the *dual-use licenses* dataset. The difference between micro and macro F1-scores is smaller for the *bill of lading* dataset.

The F1-scores per category are presented in Table 4.7. With few exceptions in the *dual-use licenses* dataset (category '4. Computers' and category '8. Marine'), all BERT implementations improve F1-scores for each category compared to the logistic regression models.

For the *dual-use licenses* dataset, BERT models perform remarkably well on category '0. Nuclear materials, facilities and equipment' (minimum F1 of 0.945) and category 'X. Non dual-use goods' (minimum F1 of 0.978). Category '8. Marine' has the lowest F1-scores for all BERT models for this dataset (maximum of 0.563). This does not hold for the *bill of lading* dataset. Although higher than other categories, the F1-scores of category '0. Nuclear materials, facilities and equipment' and 'X. Non dual-use goods' do not exceed 0.854 and 0.862 respectively. They also do not perform notably worse on category 8, which has a minimum F1-score of 0.796 for this dataset.

For the *bill of lading* dataset, there is little variation in per category F1-scores for the different BERT implementations. Where per category F1-scores show some differences between implementation for the *dual-use licenses* dataset (e.g. category 8 F1-score for plain BERT is 0.563 compared to 0.370 for KGE-BERT), these differences are not present for the *bill of lading* dataset. In fact, for this dataset the largest difference in F1-scores for BERT models is 0.008 (0.792 for pre-trained BERT minus 0.784 for pre-trained KGE-BERT).

Comparing confusion matrices in Figure 4.3 and Figure 4.4, all BERT implementations reduce classification errors in all categories compared to logistic regression models. The BERT models show some consistent classification errors for predicted category '1. Special materials and related equipment' and '2. Materials processing', where the true label is 'X. Non dual-use goods'. These errors are not found in the baseline or the TF-IDF logistic regression.

Relatively, the confusion matrices show category '7. Navigation and avionics' is often mistakenly classified as category '6. Sensors and lasers' or '8. Marine' by plain BERT and pre-trained BERT. The BERT implementations that use KGEs are performing better on this category. Plain BERT classifies 25 descriptions correctly, 7 incorrectly as category 6 and 4 incorrectly as category 8. Pre-trained BERT respectivily classifies 26 correctly, 7 incorrectly as category 6 and 3 incorrectly as category 8. KGE-BERT classifies 27 correctly, but incorrectly classifies 3 descriptions as category 9. Pre-trained KGE-BERT performs best on this category with 31 descriptions classified correctly.

Another category that is incorrectly classified relatively frequently is category '5. Telecommunications and "information security"'. All BERT implementation incorrectly classify some descriptions of this category as category '3. Electronics' and category '6. Sensors and lasers'. The same holds for category '3. Electronics', which is incorrectly labeled as '2. Materials processing' and '6. Sensors and lasers'.

For the binary classification task of which results are presented in Table 4.9, shows higher scores

Table 4.1: Best setting for all models found on *dual-use licenses* validation set.

Model	Algorithm	Triplet set	Embedding size	Epochs	Difference cutoff
LR KGE	*DistMA*	Regulation only	1024	-	-
LR TF-IDF KGE	*DistMA*	Regulation only	512	-	-
Plain BERT	-	-	-	3	-
Pre-trained BERT	-	-	-	3	-
KGE-BERT	*BoxE*	Regulation + Military	512	5	0.25
Pre-trained KGE-BERT	*TransE*	Xtrain only	256	3	0.25

Table 4.3: Test AUC and F1-scores for all models on the *dual-use licenses dataset*. BERT models are averaged over 3 runs with different seeds.

Model	Test AUC macro	Test F1 macro	Test AUC micro	Test F1 micro
Baseline	0.616	0.339	0.792	0.618
LR TF-IDF	0.803	0.691	0.906	0.827
LR KGE	0.815	0.654	0.873	0.766
LR TF-IDF KGE	0.842	0.706	0.891	0.801
Plain BERT	0.904	0.820	0.954	0.917
Pre-trained BERT	0.909	0.820	0.954	0.916
KGE-BERT	0.910	0.826	0.955	0.917
Pre-trained KGE-BERT	0.903	0.807	0.953	0.913

as compared to the 12 category multiclass classification task. Increase is greater for macro scores as compared to micro scores. Excluding the non dual-use category for the *dual-use licenses* dataset descreases performance measured in both macro AUC, micro AUC and macro F1 slightly. The most notable decrease for BERT models is found for micro F1 which decreases from around 0.913-0.917 to 0.860-0.869.

Table 4.5: Test AUC and F1-scores for BERT models on the *bill of lading dataset*.

Model	Test AUC macro	Test F1 macro	Test AUC micro	Test F1 micro
Baseline	0.504	0.067	0.576	0.281
LR TF-IDF	0.862	0.496	0.884	0.526
LR KGE	0.811	0.418	0.849	0.461
LR TF-IDF KGE	0.877	0.539	0.896	0.568
Plain BERT	0.973	0.837	0.976	0.845
Pre-trained BERT	0.973	0.836	0.977	0.844
KGE-BERT	0.973	0.835	0.977	0.843
Pre-trained KGE-BERT	0.974	0.835	0.977	0.844

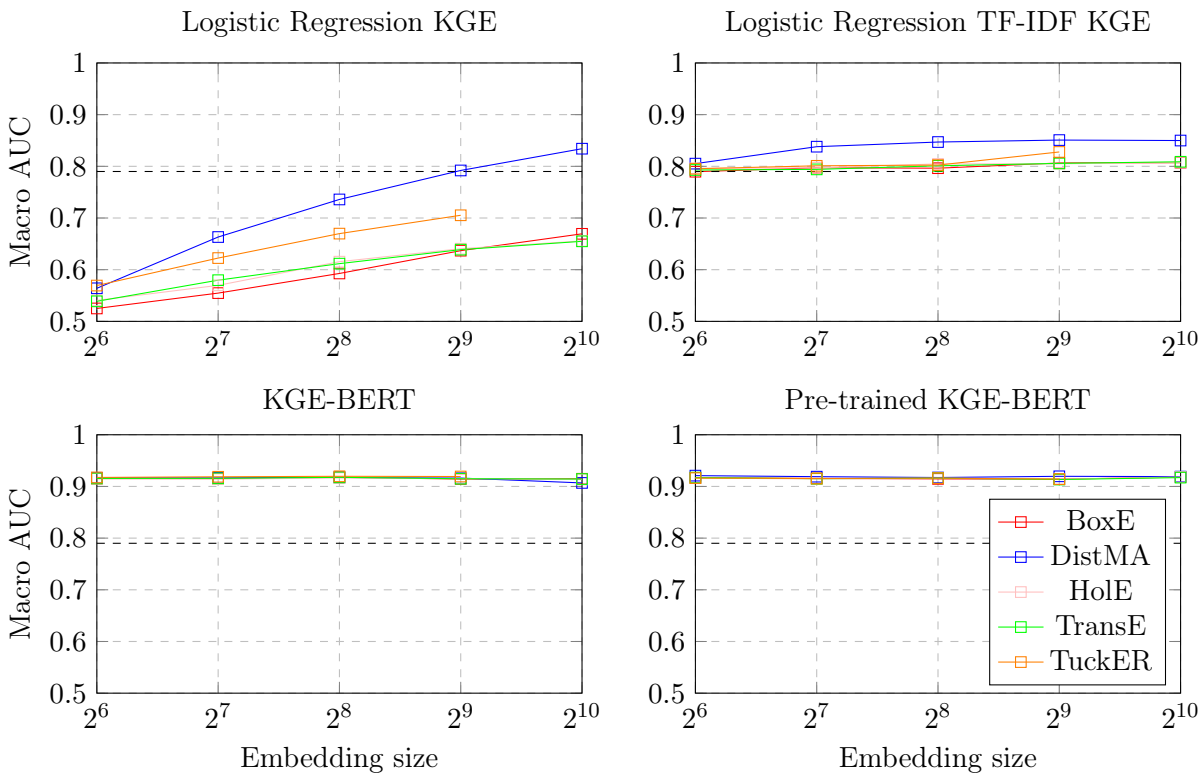


Figure 4.1: Mean validation macro AUC for models using KGEs. *TuckER* with embedding size 1024 could not be run due to its memory usage. Striped line is performance of Logistic Regression TF-IDF.

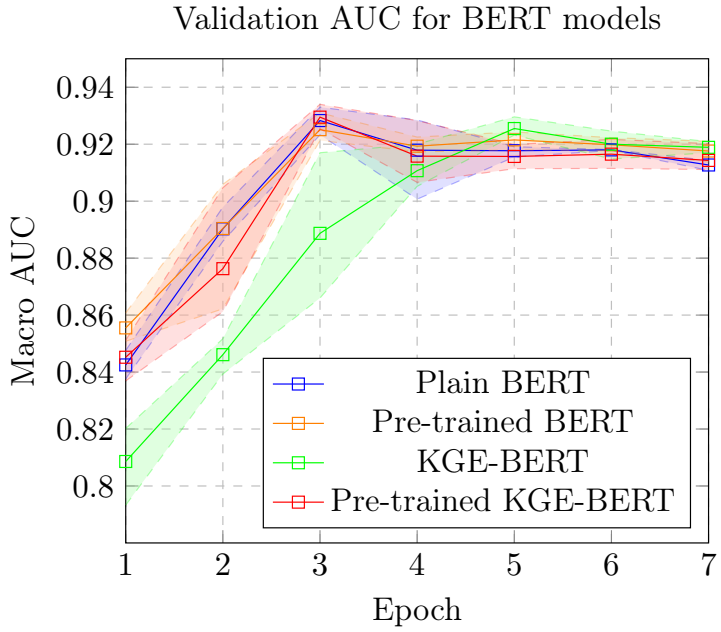
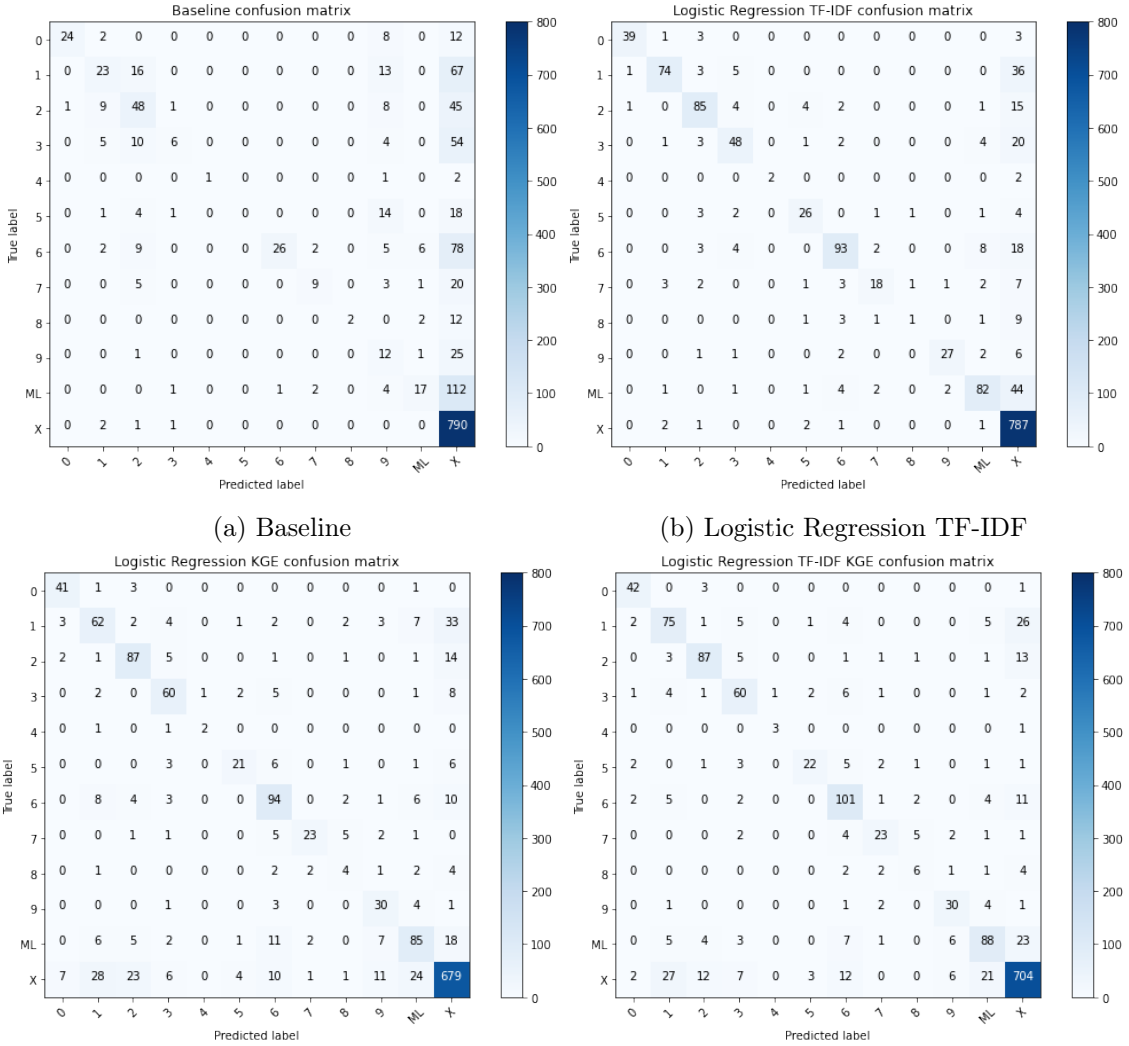


Figure 4.2: Mean validation macro AUC for BERT models on the *dual-use licenses dataset*. Solid line is the mean validation macro AUC. Shaded area indicates the minimum and maximum validation macro AUC.



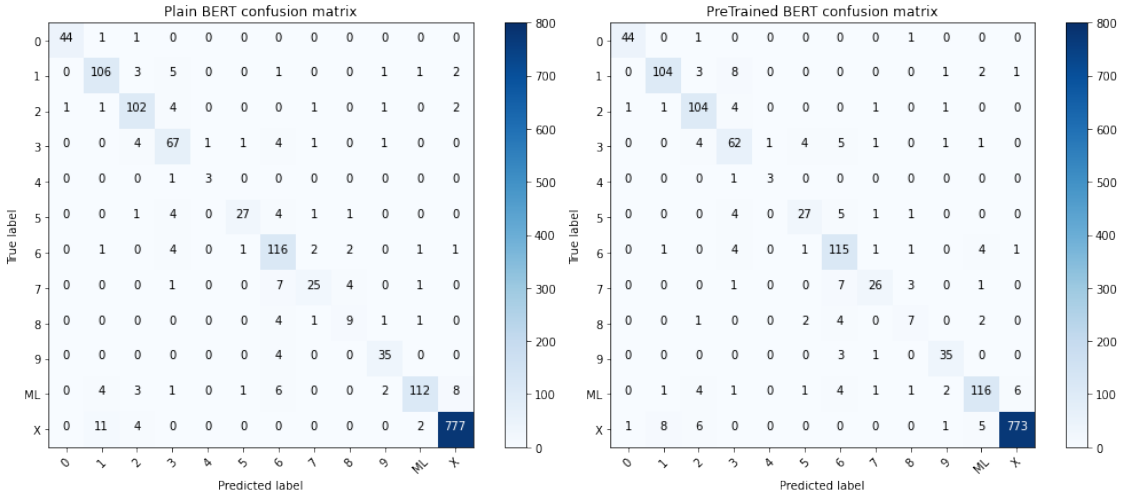
(a) Baseline

(b) Logistic Regression TF-IDF

(c) Logistic Regression KGE

(d) Logistic Regression TF-IDF KGE

Figure 4.3: Confusion matrices for the baseline and the logistic regression models on the *dual-use licenses test set*.



(a) Plain BERT

(b) Pre-trained BERT

(c) KGE-BERT

(d) Pre-trained KGE-BERT

Figure 4.4: Confusion matrices for the BERT models on the *dual-use licenses test set*.

Table 4.7: F1-scores per category on *dual-use licenses* (top) and *bill of lading* (bottom) test set for all models.

Model	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**ML**	**X**
Baseline	0.676	0.282	0.466	0.135	0.400	0.000	0.335	0.353	0.222	0.216	0.207	0.779
LR TF-IDF	0.897	0.736	0.787	0.667	0.667	0.703	0.782	0.581	0.105	0.783	0.686	0.902
LR KGE	0.833	0.546	0.737	0.720	0.571	0.618	0.701	0.697	0.303	0.638	0.620	0.868
LR TF-IDF KGE	0.866	0.628	0.787	0.723	0.750	0.667	0.745	0.648	0.387	0.714	0.667	0.890
Plain BERT	0.967	0.872	0.887	0.807	0.750	0.794	0.847	0.725	0.563	0.875	0.878	0.981
Pre-trained BERT	0.957	0.889	0.885	0.756	0.750	0.740	0.849	0.743	0.467	0.875	0.866	0.982
KGE-BERT	0.945	0.845	0.871	0.814	0.857	0.783	0.856	0.761	0.370	0.828	0.880	0.981
Pre-trained KGE-BERT	0.947	0.863	0.849	0.790	0.889	0.761	0.866	0.729	0.444	0.909	0.884	0.978

Model	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**ML**	**X**
Baseline	0.000	0.008	0.009	0.013	0.117	0.010	0.010	0.006	0.001	0.001		0.562
LR TF-IDF	0.595	0.510	0.507	0.538	0.595	0.542	0.407	0.335	0.357	0.489		0.583
LR KGE	0.535	0.451	0.425	0.510	0.506	0.458	0.353	0.192	0.261	0.369		0.543
LR TF-IDF KGE	0.614	0.557	0.544	0.599	0.637	0.580	0.449	0.382	0.417	0.518		0.631
Plain BERT	0.854	0.855	0.839	0.862	0.888	0.853	0.790	0.785	0.806	0.823		0.862
Pre-trained BERT	0.851	0.854	0.836	0.862	0.886	0.851	0.792	0.782	0.804	0.822		0.860
KGE-BERT	0.849	0.853	0.835	0.858	0.881	0.851	0.786	0.778	0.796	0.820		0.861
Pre-trained KGE-BERT	0.850	0.854	0.835	0.863	0.884	0.852	0.784	0.780	0.802	0.821		0.862

Table 4.9: Test binary AUC and binary F1-scores for all models on the *dual-use licenses dataset*. BERT models are averaged over 3 runs with different seeds. Binary scores are calculated by combining all dual-use categories into one category. This gives two categories: dual-use and non dual-use.

Model	**Test AUC macro**	**Test F1 macro**	**Test AUC micro**	**Test F1 micro**
Baseline	0.688	0.673	0.688	0.709
LR TF-IDF	0.875	0.880	0.875	0.883
LR KGE	0.862	0.861	0.862	0.862
LR TF-IDF KGE	0.879	0.878	0.879	0.879
Plain BERT	0.972	0.971	0.972	0.972
Pre-trained BERT	0.970	0.969	0.970	0.969
KGE BERT	0.972	0.971	0.972	0.971
Pre-trained KGE BERT	0.968	0.967	0.968	0.967

Table 4.11: Test partial AUC and partial F1-scores for all models on the *dual-use licenses dataset*.. BERT models are averaged over 3 runs with different seeds. Partial scores are calculated by excluding the non dual-use category for both predicted and actual labels. This gives 11 categories.

Model	**Test AUC macro**	**Test F1 macro**	**Test AUC micro**	**Test F1 micro**
Baseline	0.608	0.299	0.763	0.540
LR TF-IDF	0.796	0.672	0.912	0.836
LR KGE	0.810	0.635	0.874	0.767
LR TF-IDF KGE	0.838	0.689	0.895	0.799
Plain BERT	0.901	0.815	0.932	0.869
Pre-trained BERT	0.895	0.798	0.927	0.860
KGE-BERT	0.891	0.801	0.927	0.860
Pre-trained KGE-BERT	0.909	0.812	0.929	0.864

Chapter 5

Discussion

The aim of this research project is to retrieve useful information from different knowledge sources to build a model that can identify dual-use goods based on a short description of the good. More specifically, I use the contextual language model BERT for the dual-use goods classification task and investigate whether this model can be improved by augmenting it with relational dual-use knowledge. This results in the following research questions:

1. *Can a contextual language model such as BERT be used to identify dual-use goods based on their description?*
2. *Can augmenting this model with relational dual-use good knowledge improve its performance on the dual-use good classification task?*

The use of KGEs can improve the performance of a logistic regression model on the dual-use classification task. On average, these models show a positive relationship between its embedding size and its performance and especially small categories seem to profit from using KGEs. All implementations of BERT perform well on the dual-use identification task and have a better performance compared to the logistic regression models. None of the BERT implementations augmented with relational dual-use knowledge outperformed the plain implementation of BERT. These results will be interpreted and discussed in this chapter.

5.1 Interpretations

5.1.1 Logistic Regression

The logistic regression results on the *dual-use licenses dataset* indicate that KGEs can contain useful information for the dual-use classification task: both using only these embeddings in a logistic regression model and complementing KGEs with TF-IDF features can improve the performance of the logistic regression model. The type of algorithm is important, as the *DistMA* model improved over the other KGE algorithms. This could be due to its ability to capture multi-mapping relations (Shi & Xiao, 2019), as many triplets retrieved from Wikidata are of the form 1-N and N-1.

The embedding size of the knowledge graph was also found to be an important factor. Only larger embedding sizes, *512* and *1024*, were able to improve the performance of the KGE logistic regression models compared to the TD-IDF logistic regression. This supports the idea that knowledge graphs can contain useful information for the dual-use classification task, but need space to store information. Smaller embeddings might compress information too much to be helpful for this task.

The improvement in macro AUC is primarily thanks to the improvement of on the classes with few samples when using KGEs for logistic regression. The two smallest classes, category '4. Computers' and '8. Marine', get more weight in the macro scores compared to the micro scores. Logistic regression without KGEs has F1-scores of 0.667 and 0.105 for these categories respectively, while the implementation with both TF-IDF and KGEs improves to 0.750 and 0.387. This could indicate KGEs are especially useful for categories where little information is available, which is in line with one of the reasons to augment LMs with KGEs, namely to improve their limited coverage of knowledge.

Some examples can illustrate this argument. The description "submersible equipment" is correctly recognised by all logistic regression implementations as category '8. Marine', arguably thanks to the relatively frequent occurrence of the word "submersible" in the train set. For a somewhat more complex description in the same category, "optical fiber cable for salvage operations", no such relatively frequent word is present (the word "salvage" does not occur in the train set). It is misclassified by the TF-IDF logistic regression as 'X. Non dual-use goods', while logistic regression implementations using KGEs correctly classify this example as '8. Marine', suggesting the KGEs contain some useful knowledge.

The triplet set used for KGE creation in these cases, the 'regulation only' triplet set, does indeed contain triplets regarding optical fibers, such as ("optical fiber", "subclass of", "optical waveguide"), as well as a triplet related to (marine) salvage operations: ("marine salvage operation", "has use", "marine salvage"). It is not hard to argue that the *DistMA* algorithm was able to embed (part of) this knowledge effectively, which made it possible to classify this description correctly.

5.1.2 BERT

The use of relational knowledge was not found to improve BERT for the dual-use classification task. None of the implementations using relational knowledge notably improved its performance compared to the plain implementation of BERT. The variation in AUC between implementations are similar to variation in AUC within implementations for different seeds. This suggests that BERT already encodes, or is at least able to substitute, the knowledge encoded in the KGEs which was able to complement the TF-IDF features in the logistic regression models.

Taking the same example from the *dual-use licenses dataset* as before, plain BERT classifies "optical fiber cable for salvage operations" correctly, suggesting it does not require additional knowledge from KGEs for a correct prediction. Focusing on the F1-score for the small category '8. Marine',

plain BERT outperforms all logistic regression implementations. The descriptions that seem to be profit from KGEs in the logistic regression models in this category are correctly classified by plain BERT. In addition to this, plain BERT classifies some examples correct in this category where all logistic regression models were wrong, such as "power generation systems for marine vessels", which was classified as 'X. Non dual-use good' by the TF-IDF logistic regression model and as '6. Sensors and lasers' by the other logistic regression models.

As mentioned before, differences between BERT implementations are negligible for both the *dual-use licenses* and the *bills of lading dataset*. The F1-scores per category (Table 4.7) do not show one BERT implementation clearly outperforming others for any category. All BERT implementations improve F1-scores for each category compared to logistic regression, except for category '4. Computers' and '8. Marine' in the *dual-use licenses* dataset. Here, some BERT implementation perform equal to (0.750 F1 for category 4) or worse than the best logistic regression implementation (KGE-BERT 0.370 F1 compared to LR TF-IDF KGE 0.387 for category 8). This can be attributed to these categories having little observations, so incorrect classifications have relatively more impact on the per category F1-score. Category 4 for example, occurs 4 times in the test set. Each model that misclassifies one sample receives a F1-score of 0.750 for this category. The *bill of lading* dataset, which consists of many more descriptions, does not show this behaviour. For this dataset, all BERT implementations outperform all logistic regression implementations for all categories.

Regarding the predictions of BERT models for the *bill of lading dataset*, there is no clear pattern in errors found. Again, there are descriptions which are labeled correct by all BERT models (e.g. "NICKEL TUBE,SIZES:19X1.65X4000 MM,ASTM B163 NI200"), labeled correct by plain BERT but wrong by the augmented implementations (e.g. "OFFICE CHAIR MODEL NO.-KTM234(FOC)"), vice verse (e.g. "VR 3D GLASS (VR HEADSET)") or labeled incorrect by all (e.g. "3128 0013 05 PARM GRIP (SPARE PARTS FOR TUNNELS DRILLING MACHINE)").

5.2 Limitations

5.2.1 Datasets

The two datasets, the *dual-use licenses* and the *bill of lading dataset* are the building blocks of this thesis. The former has been the original dataset from the research proposal, the latter was later created with help of an expert in the international logistics sector. As both have their own benefits and limitations, both datasets were used.

5.2.1.1 Dual-use licenses dataset

There are some limitations that arise from the construction of the *dual-use licenses dataset* as described in Section 3.1. First, the dataset is small and consists of clean descriptions. Using clean descriptions to train, validate and test the models might result in models that do not generalize well in practice.

The imbalanced characteristic of this dataset results in certain categories having very few samples. This, in turn, can result in more sensitive F1 scores for these categories, as with category '4. Computers' and '8. Marine'. Adding to this, one of the data sources is translated from Dutch to English, which could result in some translation errors. Some other descriptions are not necessarily translated incorrectly, but could be translated with terminology not often used in practice. Incorrect or strange translations could be disadvantageous to training the model and predicting descriptions correctly.

Another limitation arising from constructing the dataset is with the 'X. Non dual-use goods' category. While licensing records for dual-use and military goods are publicly available, the 'X. Non dual-use goods' category was constructed by retrieving all traded goods and filtering the dual-use and military goods. This does not guarantee the 'non dual-use goods' category contains no dual-use or military goods. A clear example of this is "electric detonators", which is labeled as non dual-use in the dataset. Regulation (EU) 2021/821 categorizes "Electrically driven explosive detonators" under '1. Special materials and related equipment', which is the category predicted by all BERT models.

Inspecting the confusion matrices of the BERT models in Figure 4.4, shows more of these examples in the bottom row of each matrix. Some can be found to be mislabeled by checking Regulation 2021/821 (e.g. "polyurethanes"), while the correct label for other examples are harder to determine (e.g. "ignition magnetos"). Being unable to easily determine the correct label for a description is a limitation to this research. Incorrectly classified labels influence the performance metrics and might under- or overestimate the performance of certain models. It does, on the other hand, stress why a proper dual-use identification model could be helpful.

As mentioned in the results (Section 4.2.2), the confusion matrices for BERT implementations show some other consistent misclassification errors for category '7. Navigation and avionics' predicted as '6. Sensors and lasers' or '8. Marine'. An example is "hydrographic survey equipment for oil and gas industry", which is incorrectly classified as '6. Sensors and lasers' by all LR models and 3 BERT models. Pre-trained KGE-BERT classifies this description correctly. The misclassification of this description by the other BERT implementations could be the result of the train set: "hydrographic" or "oil and gas" occurs twice as much in category 6 as compared to category 7. More importantly, comparable descriptions seem to be present in categories 6, 7 and 8. This points to the next limitation.

The last limitation of the *dual-use licenses dataset* is its multiclass characteristic. The original records are multilabel, but due to the sorting of the records in alphabetical order, it is impossible to determine what description belongs to what label when multiple labels are present. For this dataset,

this limitation is simply ignored. Only descriptions that have one label are taken into account and the problem is treated as a multiclass problem where only one label can be correct.

A clear example where this is undesirable is the description 'technology for unmanned air vehicles', which is labeled as 'ML. Military goods' while all BERT models label this description as '9. Aerospace and propulsion'. Simply checking the metrics in this case would result in all BERT models being incorrect, while "Unmanned aerial vehicles (UAVs)" can also be found under '9. Aerospace and propulsion' in Regulation 2021/821.

This limitation can also influence the performance metrics and under- or overestimate the performance of certain models. As shown in the results, macro AUC, macro F1 and micro AUC are higher for the *bill of lading test set* as compared to the *dual-use licenses test set*. An increase in performance is not usually expected when shifting from clean data to empirical data. In this case, the increase in performance probably reflects the limitations of the *dual-use licenses* dataset described above. With many of these limitations only present in this dataset, performance increases when using the *bill of lading* dataset.

5.2.1.2 Bill of lading dataset

The limitations regarding the *dual-use licenses dataset* are at least partly addressed by the *bill of lading dataset*: descriptions are not cleaned or translated, it has many more samples and it can contain multiple labels per description. It does, however, have its own limitations. There can still be mislabeled descriptions, as CN-codes are not always applied correctly or do not always fit the specific good. For example, some descriptions seem mislabeled as a results of the correlation table. An example is "PLASTIC CHAIR (55 KGS)" being labeled as dual-use category '2. Materials processing', as its CN-code *94037000* describing 'Furniture of plastics' is mentioned on the correlation table. It refers to control entry *2B352f* 'Protective and containment equipment', which does not seem to match.

Another limitation might arise from this dataset. Zauba provides bill of lading data for shipments with the port of discharge in India up to 2016. Both the location and time could results in some bias. Possibly descriptions for cargo with port of discharge in India is different from other countries or good descriptions might have changed over time. Such differences introduce bias and could be detrimental to the generalisability of the model.

This dataset lacks the category 'ML. Military goods'. This is a special category. Military goods are controlled under a different regulation and different legislation might apply to the export, transit and brokering of these type of goods compared to dual-use goods. The usefulness of an 'military goods identification model' is arguably little as these goods are easier to identify as an non-expert. The lack of this category does therefore not decrease the quality of this dataset. It might be different if military good descriptions were present in the *bill of lading dataset* labeled as 'X. Non dual-use goods'. Luckily, this dataset does not seem to contain military goods by checking CN-codes or checking examples from the other dataset (i.e. "anti-aircraft systems"). Some related examples are categorized as dual-use, such as "ALUMINIUM CASTING (MFRG-DEFENCE AIRCRAFT) AS PER INV & DEC" which is labeled as '1. Special materials and related equipment', '2. Materials processing' and '9. Aerospace and propulsion'.

Grid search is performed on the *dual-use licenses* dataset only, as this would take too much time on the *bill of lading* dataset. This means the best parameters found with the grid search are not necessarily the best parameters for the *bill of lading* dataset. Regardless, the logistic regression implementation with TF-IDF and KGE improves over the implementation without KGEs on this dataset.

For the difference cutoff, a different approach is taken. As descriptions in the *bill of lading* dataset

contain more noise as compared to the *dual-use licenses* dataset, their difference is larger when comparing sentence embeddings. To account for this, the difference cutoff is set higher compared to the best value found on the *dual-use licenses* dataset. Where a cutoff of 1 would include almost all terms in the *dual-use licenses* dataset, it includes somewhat more than half the terms in the *bill of lading* dataset. Having the difference cutoff set manually without grid search poses the risk of excluding or including too many KGEs.

5.2.2 Knowledge Graph Embeddings

Two methods of augmenting BERT with relational knowledge were investigated. The augmentation method through further pre-training with KELM was partly suggested in Agarwal et al. (2020) and the augmentation method using KGEs followed the approach in Ostendorff et al. (2019). This section discusses the latter. The former is discussed in the next section.

Although the logistic regression models were initially proposed as easy-to-implement improvements on the baseline model, they provided some valuable insight in KGEs. Other than this, augmenting models with KGEs did not improve performance for BERT on any the two datasets.

Ostendorff et al. (2019) improve BERT's performance on classifying the genre of a book by augmenting it with KGEs that encode information about the author. Information about the author is retrieved from their Wikipedia page and is largely available. The approach in this thesis differs on some important aspects. First, while the Wikipedia page of an author is widely available for many authors, this is not the case for the dual-use identification task. In theory, a short description of the good can contain anything. This means finding its corresponding KGE is more difficult.

Inputs (descriptions) and KGEs are matched by comparing their sentence embeddings (as explained in Section 3.2.5 and 3.3.3). Manually comparing descriptions with their retrieved items, this approach seemed to work reasonably well. It will, however, always work less well as compared to directly retrieving a KGE as proposed by Ostendorff et al. (2019). The advantage of using sentence embeddings in this case is that any input description can be matched to any other term, avoiding matching problems for unseen descriptions. On the other hand, this might result in matching terms that are not closely related and makes this process dependent on the *all-mpnet-base-v2* model.

Matching descriptions to terms that are not closely related can be avoided by setting the difference cutoff right. But, with no clear difference in any of the grid search results for this cutoff, choosing the right parameter value proved difficult. Adding to this, the dependency on the *all-mpnet-base-v2* model might have introduced some other weaknesses. The *all-mpnet-base-v2* model is not trained on such very short (dual-use) descriptions and might not always result in the best closest term. It also increased processing time significantly.

A second important difference between the approach in Ostendorff et al. (2019) and this thesis is the use of Wikidata instead of the Wikipedia page. Both can be used to create KGEs. As for many dual-use items, there often in no (elaborate) Wikipedia page. An advantage of using Wikidata triplets is that many properties could be retrieved and used in KGE creation. A large disadvantage is that a lot of information is not present in Wikidata triplets compared to Wikipedia. Adding to this, only certain properties are used in this thesis. It could be possible that in order to improve BERT with KGEs, more information is required in these KGEs, either by creating them from Wikipedia articles or by retrieving all properties.

Interestingly, the grid search found the 'Regulation only' triplet set with terms retrieved from Regulation (EU) No. 2021/821 performing best for both logistic regression models using KGEs. As the 'Regulation only' triplet set is a subset of other triplet sets, this suggests simply adding more items to the KGEs (and thus more KGEs) is not always useful. However, no clear evidence is found for one triplet set outperforming the others. Embedding size and algorithm are the most important

factors for performance of logistic regression models using KGEs.

Some KGE parameters remain relatively unexplored in this thesis. The size of the KGE is used as a parameter, as several embedding sizes are evaluated. For most other settings, such as sampling methods, regulizer settings or train, validation and test split size are not further investigated. The usefulness of KGEs is tested according to their performance on the dual-use goods identification task. This gives some insight in the difference between algorithms and embedding sizes, but it does not give insight in why *DistMA* outperforms the others for logistic regression models, or how this information is embedded in BERT.

Given the results of this thesis, it can be questioned whether leveraging a subset of Wikidata triplets to create KGEs is the best approach. Another approach to augmenting BERT with relational dual-use knowledge is by leveraging the retrieved triplets directly instead of creating KGEs. Graph traversal algorithms could prove useful in this case.

5.2.3 Language Models

Further pre-training on in-domain data is a common approach to improve the performance of BERT on a given (classification) task. In this thesis, a subset of KeLM was used in further pre-training in order to augment BERT with the relational knowledge present in KeLM. But there are some differences between the approach in this thesis and the augmentation methods suggested in Agarwal et al. (2020) which might affect the performance on the dual-use classification task.

For the KeLM dataset, the authors suggested using the full dataset to leverage relational KG data in BERT. In this thesis, only a subset of synthetic sentences based on TIM terms is used. The idea behind using a subset was not only to make pre-training faster. By focusing on synthetic sentences related to dual-use goods, it might become easier for a model to leverage this knowledge. However, this approach did not improve BERT's performance on the dual-use goods identification task. Possibly, the whole KeLM dataset or a larger subset would be able to so.

On the other hand, it can be questioned whether the KeLM is the best option for further pre-training BERT for the dual-use goods identification task. Inspecting the retrieved synthetic sentences, there are (lowercased) sentences that might be useful (e.g. "uraninite is a mineral that is a class of uranium dioxide"). But there are quite some sentences that seem less relevant (e.g. "uranium mining is the main topic of uranium mining in kazakhstan"). Possibly, real (not synthetic) texts such as regulations, jurisprudence, news articles or research articles would be a better fit for further pre-training BERT for the task at hand.

So far, no research has been published except for the original paper by Agarwal et al. (2020) that uses the KeLM corpus for further pre-training for any downstream task. If so, it would be interesting to know if any cleaning or selection process is done on the corpus before using it.

As a final remark on the limitations of this research, plain BERT performed unexpectedly well for the dual-use identification task. As this helps answering the first research question, it makes answering the second research question much harder. The better BERT performs out-of-the-box, the harder it becomes to improve its performance with relational knowledge. The relational knowledge used to augment BERT for this task should not simply encode general dual-use knowledge, it might need very specific knowledge not encoded in BERT. Although augmentation was insufficient to improve BERT in this thesis, future attempts might be successful.

5.3 Future work

All implementations of BERT performed well on the dual-use classification task. This performance remained on the empirical *bill of lading dataset*. Although none of the BERT implementations augmented with relational dual-use data improved its performance, the results and limitations of this research suggest some approaches for further research.

Results of the logistic regression models indicate KGEs can contain relational dual-use knowledge to improve the performance of a simple model on the dual-use identification task. Given the relatively small subset of data used in both augmentation approaches (i.e. a subset of KeLM based on TIM terms and using a subset of Wikidata triplets based on regulation terms for KGE creation), it would be an interesting approach to scale up the (dual-use) knowledge that can be encoded and thereby possibly encoding knowledge that is not present in BERT. Using the same implementations as in this thesis, an approach could be to use the *dual-use licenses dataset* as a knowledge source instead of a dataset. Given the limitations of both datasets, future research could also benefit from more recent data which covers more than one port of discharge.

Research into why *DistMA* outperformed the other algorithms for the logistic regression models might provide interesting findings. Finding the driving factor behind *DistMA*'s performance could provide stepping stones for research on augmenting LMs with KGEs or even improving KGE models on other tasks. It would also be interesting to see what the performance of *TuckER* with embedding size *1024* would be, as this setting could not be run due to lack of memory and embedding size *512* increased performance compared to embedding size *256*. Testing promising CNN-based KGE models, such as *ConvE* and *ConvKB*, for augmentation of BERT for the dual-use identification task would also be interesting, as these models showed great improvements for long-term relational dependencies in KGs - a characteristic dual-use good identification might benefit greatly from.

For further research into improving BERT on the dual-use goods identification task without augmenting with relational knowledge per se, it would be interesting to further pre-train BERT on regulations or scientific articles instead of the KeLM corpus.

As mentioned before, instead of converting triplets into KGEs for augmenting BERT, it is possible to leverage these triplets directly. Not only might this improve BERT's performance, it could also play a role in increasing the explainability of such model. Explainability is not investigated in this thesis, but it plays an important role in both artificial intelligence and regulatory compliance.

Finally, future work not related to academic research involves building an application out of the BERT models proposed in this research. These models perform well enough to initiate a proof of concept with human supervision. They could be used to help employees involved in international trade make decisions about applying for a dual-use goods export license. They should not be used in an automatic decision making process involving these licenses or other export activities. The company active in the international logistics sector mentioned in Section 3.1.3 acknowledged that a dual-use identification application would be useful in their field of work.

Other sectors engaged in international trade for which a dual-use identification application would be useful are trade finance (the financing of goods or services in a trade or transaction) and payment service providers (providing payment methods and its technical processing to online shops). The former can implement any of the BERT models discussed in this thesis, as the bill of lading is commonly used in evaluating the financing of a trade. Payment service providers do not use the bill of lading, so applying the model to this sector might require adaptations to the model's training process.

References

Abboud, R., Ceylan, I., Lukasiewicz, T., & Salvatori, T. (2020). BoxE: A box embedding model for knowledge base completion. *Advances in Neural Information Processing Systems*, *33*, 9649–9661.

Adhikari, A., Ram, A., Tang, R., & Lin, J. (2019). DocBERT: BERT for document classification. *arXiv preprint arXiv:1904.08398*.

Agarwal, O., Ge, H., Shakeri, S., & Al-Rfou, R. (2020). Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. *arXiv preprint arXiv:2010.12688*.

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., & Vollgraf, R. (2019). FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)* (pp. 54–59).

Akbik, A., Blythe, D., & Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics* (pp. 1638–1649).

Ali, M., Berrendorf, M., Hoyt, C. T., Vermue, L., Sharifzadeh, S., Tresp, V., & Lehmann, J. (2021). PyKEEN 1.0: A Python Library for Training and Evaluating Knowledge Graph Embeddings. *Journal of Machine Learning Research*, *22*(82), 1–6. Retrieved from http://jmlr.org/papers/v22/20-825.html

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2007). DBpedia: A nucleus for a web of open data. In *The semantic web* (pp. 722–735). Springer.

Balažević, I., Allen, C., & Hospedales, T. M. (2019). TuckER: Tensor factorization for knowledge graph completion. *arXiv preprint arXiv:1901.09590*.

Baoli, L., Qin, L., & Shiwen, Y. (2004). An adaptive k-nearest neighbor text categorization strategy. *ACM Transactions on Asian Language Information Processing (TALIP)*, *3*(4), 215–226.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data* (pp. 1247–1250).

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, *26*.

Brants, T. (2000). TnT-a statistical part-of-speech tagger. *arXiv preprint cs/0003055*.

Brill, E. (1992). A simple rule-based part of speech tagger. In *3rd Conference on Applied Natural Language Processing (ANLC)*.

Cao, B., Lin, H., Han, X., Sun, L., Yan, L., Liao, M., ... Xu, J. (2021). Knowledgeable or educated guess? revisiting language models as knowledge bases. *arXiv preprint arXiv:2106.09231*.

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., & Androutsopoulos, I. (2020). LEGAL-BERT: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*.

Chen, J., Huang, H., Tian, S., & Qu, Y. (2009). Feature selection for text classification with Naïve Bayes. *Expert Systems with Applications*, *36*(3), 5432–5435.

Church, K. W. (1989). A stochastic parts program and noun phrase parser for unrestricted text. In *International Conference on Acoustics, Speech, and Signal Processing* (pp. 695–698).

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning* (pp. 160–167).

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Council of European Union. (2020). *Common Military List of the European Union.*

Council of European Union. (2021). *Council Regulation (EU) No 2021/821.*

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), 21–27.

Cutting, D., Kupiec, J., Pedersen, J., & Sibun, P. (1992). A practical part-of-speech tagger. In *Third Conference on Applied Natural Language Processing* (pp. 133–140).

DeepL. (2022). *Deepl translator.* Retrieved 2022-06-15, from https://www.deepl.com/translator

Dettmers, T., Minervini, P., Stenetorp, P., & Riedel, S. (2018). Convolutional 2d knowledge graph embeddings. In *Thirty-second AAAI Conference on Artificial Intelligence.*

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dow Jones. (2019). *The challenge of tracking dual-use goods for trade finance compliance.* Retrieved 2021-12-08, from https://visit.dowjones.com/risk/content/dual-use-goods/

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211.

Ettinger, A. (2020). What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, *8*, 34–48.

European Commision. (2022). *Correlation list between TARIC and Dual-use Annex of Regulation (EU) 2021/821.* Retrieved 2022-06-15, from https://trade.ec.europa.eu/doclib/html/160003.htm

European Commission. (2021a). *The Combined Nomenclature.* Retrieved 2022-06-15, from https://ec.europa.eu/taxation_customs/business/calculation-customs-duties/customs-tariff/combined-nomenclature_en

European Commission. (2021b). *Dual-use trade controls.* Retrieved 2021-12-08, from https://ec.europa.eu/trade/import-and-export-rules/export-from-eu/dual-use-controls/

European Commission. (2021c). *Tools for innovation monitoring.* Retrieved 2022-01-16, from https://knowledge4policy.ec.europa.eu/text-mining/topic/tim_analytics_en

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, *27*(8), 861–874.

Fukushima, K., & Miyake, S. (1982). Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets* (pp. 267–285). Springer.

Ghanem, A. S., Venkatesh, S., & West, G. (2010). Multi-class pattern classification in imbalanced data. In *2010 20th International Conference on Pattern Recognition* (pp. 2881–2884).

Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*.

Guo, L., Sun, Z., & Hu, W. (2019). Learning to exploit long-term relational dependencies in knowledge graphs. In *International conference on machine learning* (pp. 2505–2514).

Guo, X., Yin, Y., Dong, C., Yang, G., & Zhou, G. (2008). On the class imbalance problem. In *2008 Fourth International Conference on Natural Computation* (Vol. 4, pp. 192–201).

He, Z., Wang, Z., Wei, W., Feng, S., Mao, X., & Jiang, S. (2020). A survey on recent advances in sequence labeling from deep learning models. *arXiv preprint arXiv:2011.06727*.

HM Revenue and Customs. (2021). *UK Trade Info.* Retrieved 2021-12-08, from https://www.uktradeinfo.com/

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Ji, S., Pan, S., Cambria, E., Marttinen, P., & Philip, S. Y. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*.

Jiang, Z., Xu, F. F., Araki, J., & Neubig, G. (2020). How can we know what language models know? *Transactions of the Association for Computational Linguistics*, *8*, 423–438.

Jo, T. (2017). Using K Nearest Neighbors for text segmentation with feature similarity. In *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)* (pp. 1–5).

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *European Conference on Machine Learning* (pp. 137–142).

Joachims, T., et al. (1999). Transductive inference for text classification using support vector machines. In *Icml* (Vol. 99, pp. 200–209).

Johnson, R., & Zhang, T. (2014). Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058*.

Johnson, R., & Zhang, T. (2016). Supervised and semi-supervised text categorization using LSTM for region embeddings. In *International Conference on Machine Learning* (pp. 526–534).

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Klein, S., & Simmons, R. F. (1963). A computational approach to grammatical coding of English words. *Journal of the ACM (JACM)*, *10*(3), 334–347.

Kneppelhout. (2021). *Export controls & sanctions.* Retrieved 2021-12-08, from https://kneppelhout.com/news/export-controls-sanctions/

Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information*, *10*(4), 150.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *International Conference on Machine Learning* (pp. 1188–1196).

Lerer, A., Wu, L., Shen, J., Lacroix, T., Wehrstedt, L., Bose, A., & Peysakhovich, A. (2019). PyTorch-BigGraph: A Large-scale Graph Embedding System. In *Proceedings of the 2nd SysML Conference.* Palo Alto, CA, USA.

Li, Q., Peng, H., Li, J., Xia, C., Yang, R., Sun, L., ... He, L. (2020). A survey on text classification: From shallow to deep learning. *arXiv preprint arXiv:2008.00364*.

Liu, N., Zhang, B., Yan, J., Chen, Z., Liu, W., Bai, F., & Chien, L. (2005). Text representation: From vector to tensor. In *Fifth IEEE International Conference on Data Mining (ICDM'05)* (pp. 4–pp).

Liu, P., Qiu, X., & Huang, X. (2016). Recurrent neural network for text classification with multi-task learning. *arXiv preprint arXiv:1605.05101*.

Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., & Wang, P. (2020). K-BERT: Enabling language representation with knowledge graph. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, pp. 2901–2908).

Ma, X., & Hovy, E. (2016). End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. *arXiv preprint arXiv:1603.01354*.

Maron, M. E. (1961). Automatic indexing: an experimental inquiry. *Journal of the ACM (JACM)*, *8*(3), 404–417.

McCallum, A., Nigam, K., et al. (1998). A comparison of event models for Naïve Bayes text classification. In *AAAI-98 workshop on learning for text categorization* (Vol. 752, pp. 41–48).

Melamud, O., Goldberger, J., & Dagan, I. (2016). context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of the 20th SIGNLL conference on computational natural language learning* (pp. 51–61).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, *38*(11), 39–41.

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning–based text classification: A comprehensive review. *ACM Computing Surveys (CSUR)*, *54*(3), 1–40.

Nederlandse Vereniging van Banken. (2020). *AML, CTF & Sanctions Guidance Part II.*

Nguyen, D. Q., Nguyen, T. D., Nguyen, D. Q., & Phung, D. (2017). A novel embedding model for knowledge base completion based on convolutional neural network. *arXiv preprint arXiv:1712.02121*.

Nickel, M., Rosasco, L., & Poggio, T. (2016). Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 30).

Nickel, M., Tresp, V., & Kriegel, H.-P. (2011). A three-way model for collective learning on multi-relational data. In *Icml.*

Ostendorff, M., Bourgonje, P., Berger, M., Moreno-Schneider, J., Rehm, G., & Gipp, B. (2019). Enriching BERT with knowledge graph embeddings for document classification. *arXiv preprint arXiv:1909.08402*.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1532–1543).

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *NAACL.*

Peters, M. E., Neumann, M., Logan IV, R. L., Schwartz, R., Joshi, V., Singh, S., & Smith, N. A. (2019). Knowledge enhanced contextual word representations. *arXiv preprint arXiv:1909.04164*.

Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., & Riedel, S. (2019). Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Poerner, N., Waltinger, U., & Schütze, H. (2020, November). E-BERT: Efficient-yet-effective entity embeddings for BERT. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (pp. 803–818). Online: Association for Computational Linguistics. Retrieved from https://aclanthology.org/2020.findings-emnlp.71 doi: 10.18653/v1/2020.findings-emnlp.71

Rabanser, S., Shchur, O., & Günnemann, S. (2017). Introduction to tensor decompositions and their applications in machine learning. *arXiv preprint arXiv:1711.10781*.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.

Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Reimers, N., & Gurevych, I. (2019, 11). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. Retrieved from https://arxiv.org/abs/1908.10084

Richens, R. H. (1956). Preprogramming for mechanical translation. *Mech. Transl. Comput. Linguistics*, *3*(1), 20–25.

Rijksoverheid. (2019). *Guidelines for compiling an Internal Compliance Programme.*

Rijksoverheid. (2022a). *Handboek strategische goederen en diensten.* Retrieved 2022-03-10, from https://www.rijksoverheid.nl/documenten/rapporten/2006/10/23/handboek-strategische-goederen

Rijksoverheid. (2022b). *Maandelijkse rapportage uitvoer dual-use-goederen.* Retrieved 2022-03-10, from https://www.rijksoverheid.nl/onderwerpen/exportcontrole-strategische-goederen/documenten/rapporten/2016/10/01/overzicht-dual-use-vergunningen

Rijksoverheid. (2022c). *Maandelijkse rapportage uitvoer militaire goederen.* Retrieved 2022-03-10, from https://www.rijksoverheid.nl/documenten/rapporten/2016/10/01/overzicht-uitvoer-militaire-goederen

Rogers, A., Kovaleva, O., & Rumshisky, A. (2020). A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, *8*, 842–866.

Ruffinelli, D., Broscheit, S., & Gemulla, R. (2019). You can teach an old dog new tricks! on training knowledge graph embeddings. In *International Conference on Learning Representations.*

Safavi, T., & Koutra, D. (2021). Relational world knowledge representation in contextual language models: A review. *arXiv preprint arXiv:2104.05837*.

Shi, X., & Xiao, Y. (2019). Modeling multi-mapping relations for precise cross-lingual entity alignment. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 813–822).

Singhal, A. (2012). *Introducing the Knowledge Graph: things, not strings.* Retrieved 2022-07-04, from https://blog.google/products/search/introducing-knowledge-graph-things-not/

Socher, R., Huval, B., Manning, C. D., & Ng, A. Y. (2012). Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning* (pp. 1201–1211).

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., & Manning, C. D. (2011). Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 151–161).

Suchanek, F. M., Kasneci, G., & Weikum, G. (2007). Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web* (pp. 697–706).

Sun, A., Lim, E.-P., & Liu, Y. (2009). On strategies for imbalanced text classification using svm: A comparative study. *Decision Support Systems*, *48*(1), 191–201.

Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to fine-tune BERT for text classification? In *China National Conference on Chinese Computational Linguistics* (pp. 194–206).

Sung, M., Lee, J., Yi, S., Jeon, M., Kim, S., & Kang, J. (2021). Can language models be biomedical knowledge bases? *arXiv preprint arXiv:2109.07154*.

Tai, K. S., Socher, R., & Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., ... others (2019). What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., & Bouchard, G. (2016). Complex embeddings for simple link prediction. In *International Conference on Machine Learning* (pp. 2071–2080).

Tsoumakas, G., Katakis, I., & Vlahavas, I. (2006). A review of multi-label classification methods. In *Proceedings of the 2nd ADBIS workshop on data mining and knowledge discovery (ADMKD 2006)* (pp. 99–109).

UK Department for International Trade. (2021a). *Reports and Statistics Home.* Retrieved 2021-12-08, from https://www.exportcontroldb.trade.gov.uk/

UK Department for International Trade. (2021b). *Strategic export controls: licensing data annual reports.* Retrieved 2021-12-08, from https://www.gov.uk/government/collections/strategic-export-controls-licensing-data-annual-reports

UK Department for International Trade. (2021c). *UK strategic export control lists - the consolidated list of strategic military and dual-use items that require export authorisation from great britain and northern ireland.* Retrieved 2021-12-09, from https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/948279/uk-strategic-export-control-list.pdf

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998–6008).

Vrandečić, D., & Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, *57*(10), 78–85.

Wang, Q., Huang, P., Wang, H., Dai, S., Jiang, W., Liu, J., ... Wu, H. (2019). CoKE: Contextualized knowledge graph embedding. *arXiv preprint arXiv:1911.02168*.

Wang, Q., Mao, Z., Wang, B., & Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, *29*(12), 2724–2743.

Wang, R., Tang, D., Duan, N., Wei, Z., Huang, X., Cao, G., ... others (2020). K-ADAPTER: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*.

Wang, X., Gao, T., Zhu, Z., Zhang, Z., Liu, Z., Li, J., & Tang, J. (2021). KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, *9*, 176–194.

Wolfsberg Group. (2019). *Trade finance principles 2019.*

Wu, T., Huang, Q., Liu, Z., Wang, Y., & Lin, D. (2020). Distribution-balanced loss for multi-label classification in long-tailed datasets. In *European Conference on Computer Vision* (pp. 162–178).

Wu, Z., Chen, Y., Kao, B., & Liu, Q. (2020). Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. *arXiv preprint arXiv:2004.14786*.

Yang, B., Yih, W.-t., He, X., Gao, J., & Deng, L. (2014). Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information retrieval*, *1*(1), 69–90.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1480–1489).

Yao, L., Mao, C., & Luo, Y. (2019). KG-BERT: BERT for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.

Yong, Z., Youwen, L., & Shixiong, X. (2009). An improved KNN text classification algorithm based on clustering. *Journal of computers*, *4*(3), 230–237.

Zauba Technologies. (2022). *Import export data.* Retrieved 2022-06-15, from https://www.zauba.com/

Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019). ERNIE: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*.

Zhang, Z., Zhuang, F., Zhu, H., Li, C., Xiong, H., He, Q., & Xu, Y. (2021). Towards robust knowledge graph embedding via multi-task reinforcement learning. *IEEE Transactions on Knowledge and Data Engineering*.

Zhou, X., Wan, X., & Xiao, J. (2016). Attention-based LSTM network for cross-lingual sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 247–256).

Zhu, X., Sobihani, P., & Guo, H. (2015). Long short-term memory over recursive structures. In *International Conference on Machine Learning* (pp. 1604–1612).

Zou, X. (2020). A survey on application of knowledge graph. In *Journal of Physics: Conference Series* (Vol. 1487, p. 012016).

Appendix A

Filtered Harmonized System codes

This table presents the Harmonized System (HS) codes filtered from the non dual-use good descriptions in Section 3.1.2.3.

HS-code	Description
2529	Feldspar; leucite; nepheline and nepheline syenite; fluorspar
2610	Chromium ores and concentrates
2611	Tungsten ores and concentrates
2612	Uranium or thorium ores and concentrates
2613	Molybdenum ores and concentrates
2614	Titanium ores and concentrates
2615	Niobium, tantalum, vanadium or zirconium ores and concentrates
2812	Halides and halide oxides of non-metals
2825	Hydrazine and hydroxylamine and their inorganic salts; other inorganic bases; other metal oxides, hydroxides and peroxides
2844	Radioactive chemical elements and radioactive isotopes (including the fissile or fertile chemical elements and isotopes); and their compounds; mixtures and residues containing these products
2845	Isotopes other than those of heading no. 2844; compounds, inorganic or organic, of such isotopes, whether or not chemically defined
2847	Hydrogen peroxide; whether or not solidified with urea
2921	Amine-function compounds
2928	Organic derivatives of hydrazine or of hydroxylamine
2931	Other organo-inorganic compounds
3101	Fertilizers; animal or vegetable, whether or not mixed together or chemically treated; fertilizers produced by the mixing or chemical treatment of animal or vegetable products
3102	Fertilizers; mineral or chemical, nitrogenous
3103	Fertilizers; mineral or chemical, phosphatic
3104	Fertilizers; mineral or chemical, potassic
3105	Fertilizers; mineral or chemical, containing 2 or 3 of the elements nitrogen, phosphorus, potassium; other fertilisers; goods of chapter 31 in tablets or packages of gross weight not exceeding 10kg
3602	Prepared explosives, other than propellent powders
3821	Prepared culture media for the development or maintenance of micro-organisms (including viruses and the like) or of plant, human or animal cells
3822	Reagents; diagnostic or laboratory reagents on a backing and prepared diagnostic or laboratory reagents whether or not on a backing, other than those of heading no. 3002 or 3006; certified reference material
3904	Polymers of vinyl chloride or of other halogenated olefins, in primary forms
8109	Zirconium; articles thereof, including waste and scrap
8112	Beryllium, chromium, germanium, vanadium, gallium, hafnium, indium, niobium (columbium), rhenium and thallium; and articles of these metals, including waste and scrap
8401	Nuclear reactors; fuel elements (cartridges), non-irradiated, for nuclear reactors, machinery and apparatus for isotopic separation

HS-code	Description
8411	Turbo-jets, turbo-propellers and other gas turbines
8414	Air or vacuum pumps, air or other gas compressors and fans; ventilating or recycling hoods incorporating a fan whether or not fitted with filters
8421	Centrifuges, including centrifugal dryers; filtering or purifying machinery and apparatus for liquids or gases
8442	Machinery, apparatus and equipment (excluding machines of headings 8456 to 8465) for preparing or making printing components; plates, cylinders and other printing components; lithographic stones prepared for printing purposes
8456	Machine-tools; for working any material by removal of material, by laser or other light or photon beam, ultrasonic, electro-discharge, electro-chemical, electron beam, ionic-beam, or plasma arc processes; water-jet cutting machines
8484	Gaskets and similar joints of metal sheeting combined with other material or of two or more layers of metal; sets or assortments of gaskets and similar joints, dissimilar in composition, put up in pouches, envelopes or similar packings; mechanical seals
8486	Machines and apparatus of a kind used solely or principally for the manufacture of semiconductor boules or wafers, semiconductor devices, electronic integrated circuits or flat panel displays; machines and apparatus specified in note 9-C to this Chapter
8515	Electric (electrically heated gas) soldering, brazing, welding machines and apparatus, capable or not of cutting, electric machines and apparatus for hot spraying of metals or sintered carbides
8532	Electrical capacitors; fixed, variable or adjustable (pre-set)
8540	Thermionic, cold cathode or photo-cathode valves and tubes (e.g. vacuum, vapour, gas filled valves and tubes, mercury arc rectifying valves and tubes, cathode-ray and television camera tubes)
8541	Diodes, transistors, similar semiconductor devices; including photovoltaic cells assembled or not in modules or panels, light-emitting diodes (LED), mounted piezo-electric crystals
8542	Electronic integrated circuits
8543	Electrical machines and apparatus; having individual functions, not specified or included elsewhere in this chapter
8545	Carbon electrodes, carbon brushes, lamp carbons, battery carbons and other articles of graphite or other carbon; with or without metal, of a kind used for electrical purposes
8710	Tanks and other armoured fighting vehicles; motorised, whether or not fitted with weapons, and parts of such vehicles
8801	Balloons and dirigibles; gliders, hang gliders and other non-powered aircraft.
8802	Aircraft n.e.c. in heading no. 8801 (e.g. helicopters, aeroplanes); spacecraft (including satellites) and suborbital and spacecraft launch vehicles
8803	Aircraft; parts of heading no. 8801 or 8802
8804	Parachutes (including dirigible parachutes and paragliders) and rotochutes; parts thereof and accessories thereto
8805	Aircraft launching gear, deck-arrestor or similar gear, ground flying trainers; parts of the foregoing articles
8901	Cruise ships, excursion boats, ferry-boats, cargo ships, barges and similar vessels for the transport of persons or goods
8902	Fishing vessels, factory ships and other vessels; for processing or preserving fishery products
8903	Yachts and other vessels; for pleasure or sports, rowing boats and canoes
8904	Tugs and pusher craft
8905	Light-vessels, fire-floats, dredgers, floating cranes, other vessels; the navigability of which is subsidiary to main function; floating docks, floating, submersible drilling, production platforms
8906	Vessels; other, including warships and lifeboats, other than rowing boats
8907	Boats, floating structures, other (for e.g. rafts, tanks, coffer-dams, landing stages, buoys and beacons)
8908	Vessels and other floating structures; for breaking up
9020	Breathing appliances and gas masks; excluding protective masks having neither mechanical parts nor replaceable filters and excluding apparatus of item no. 9019.20
9027	Instruments and apparatus; for physical or chemical analysis (e.g. polarimeters, spectrometers), for measuring or checking viscosity, porosity, etc, for measuring quantities of heat, sound or light
9301	Military weapons; other than revolvers, pistols and arms of heading no. 9307
9302	Revolvers and pistols; other than those of heading no. 9303 or 9304

HS-code	Description
9303	Firearms; other similar devices (e.g. sporting shotguns and rifles, muzzle-loading firearms, very pistols, devices for firing flares or blank ammunition, captive bolt humane killers, line throwing guns)
9304	Firearms; (e.g. spring, air or gas guns and pistols, truncheons), excluding those of heading no. 9307
9305	Firearms; parts and accessories of articles of heading no. 9301 to 9304
9306	Bombs, grenades, torpedoes, mines, missiles and similar munitions of war and parts thereof; cartridges and other ammunition, projectiles and parts thereof, including shot and cartridge wads
9307	Arms; swords, cutlasses, bayonets, lances and the like, parts thereof and scabbards and sheaths therefor
9880	Component parts of complete industrial plant in the framework of external trade
9930	Goods delivered to vessels and aircraft goods classified elsewhere
9931	Goods delivered for the crew of the offshore installation or for the operation of the engines, machines and other equipment of the offshore installation

Appendix B

Code

The code for this research can be found on
https://colab.research.google.com/drive/19kyVECD21R579FOqym1UAS4zcpKCst3E?usp=sharing