



SCRE: special cargo relation extraction using representation learning

Vahideh Reshadat¹ · Alp Akcay¹ · Kalliopi Zervanou¹ · Yingqian Zhang¹ · Eelco de Jong²

Received: 2 December 2022 / Accepted: 23 May 2023 / Published online: 18 June 2023
© The Author(s) 2023

Abstract

The airfreight industry of shipping goods with special handling needs, also known as special cargo, often deals with non-transparent data and outdated technology, resulting in significant inefficiency. A special cargo ontology is a means of extracting, structuring, and storing domain knowledge and representing the concepts and relationships that can be processed by computers. This ontology can be used as the base of semantic data retrieval in many artificial intelligence applications, such as planning for special cargo shipments. Domain information extraction is an essential task in implementing and maintaining special cargo ontology. However, the absence of domain information makes instantiating the cargo ontology challenging. We propose a relation representation learning approach based on a hierarchical attention-based multi-task model and leverage it in the special cargo domain. The proposed relation representation learning architecture is applied for identifying and categorizing samples of various relation types in the special cargo ontology. The model is trained with domain-specific documents on a number of semantic tasks that vary from lightweight tasks in the bottom layers to the heavyweight tasks in the top layers of the model in a hierarchical setting. Therefore, it conveys complementary input features and learns a rich representation. We also train a domain-specific relation representation model that relies only on an entity-linked corpus of cargo shipment domain. These two relation representation models are then employed in a supervised multi-class classifier called Special Cargo Relation Extractor (SCRE). The results of the experiments show that the proposed relation representation models can represent the complex semantic information of the special cargo domain efficiently.

Keywords Special cargo shipment · Relation representation · Relation classification

1 Introduction

The use of digital technologies is no longer optional but necessary to become competitive in many industries. The airfreight industry is one of them. However, during the digitization process in the airfreight industry, little attention has been paid to the special cargo transportation [1]. The special cargo (or special freight) refers to the substances which require special handling during transportation (e.g., pharmaceuticals with stringent temperature requirements, live animals, chemical and food products). Managing the process of special cargo transportation depends on a variety of domain data. In practice, this data are often in large volume and variety with different granularities, formats and sometimes ambiguous properties. Ontologies can effectively address these challenges by conceptualizing and structuring the domain knowledge. They are used in many applications such as semantic integration and reasoning, routing and various operational

✉ Vahideh Reshadat
v.reshadat@tue.nl

Alp Akcay
A.E.Akcay@tue.nl

Kalliopi Zervanou
K.Zervanou@tue.nl

Yingqian Zhang
YQZhang@tue.nl

Eelco de Jong
Eelco.deJong@validaide.com

¹ Department of Industrial Engineering and Innovation Sciences, Atlas 5.321, Eindhoven University of Technology, P.O. Box 513, Eindhoven 5600 MB, The Netherlands

² Validaide B.V., Amsterdam, The Netherlands

decisions in special cargo transportation. They also can be exploited in the question-answering task for specific queries about routing special cargos.

For instance, route planning at freight forwarders can benefit from advanced information extraction and ontologies. Route planning for special cargo products (i.e., what carriers, services, additional options, and conditions to choose for special cargo shipment) can be a complex task as the number of possibilities for routing decisions can be overwhelmingly high. A recent example is the transportation of COVID-19 vaccines that needs cautious routing decisions. Airfreight forwarders make these decisions by acquiring the necessary information related to routings with the help of industry partners. Nowadays, most activities are still conducted with human intervention and operators rely heavily on expert knowledge. Therefore, getting all the necessary information can be challenging because of the complex attributes of goods (e.g., chemical characteristics) and the absence of standardization in the services of forwarders and suppliers. Manual data collection is inefficient and costly, negatively affecting the performance of the entire shipment operation [1]. Thus, there is an urgent need to develop a database of special freight capabilities that provides structured data from which specific features of the historical shipments data can be retrieved to build accurate predictive models for routing options. Such predictive models can provide a quick evaluation over different routing options in terms of performance, e.g., the probability of delay and/or damage.

Figure 1 shows a general schema of how our proposed methodology can be applied in the context of special cargo transportation. In particular, the proposed relation representation models are part of the Information Extraction (IE) engine that helps to populate an ontology and employ it in semantic reasoning. In general, IE is the task of automatically extracting useful information from unstructured and/or semi-structured enormous data and is an essential step in the ontology population. It is used for a variety of tasks in Natural Language Processing (NLP), such as machine translation [2] and speech recognition [3]. Other advanced related tasks to IE include handling repetitive processes and disturbances [4, 5] and Building Temperature Control [6] that apply Iterative Learning Control (ILC) approach.

ILC uses the previous experiment data to handle the repetitive control processes and improves the tracking performance for repetitive processes by learning from the historical information.

In the special cargo domain, the extraction of the available information for populating the ontology has also a critical role. Relation extraction is a fundamental subtask in IE that helps to identify relationships between instances of ontology concepts and to acquire and classify ontology instances. In this paper, we extract and structure the special cargo information from a number of resources to instantiate the special cargo knowledge resource that is built using knowledge extraction and UPON (Unified Process for Ontology) [7]. The results obtained from ontological reasoning can aid the managers of airfreight forwarders in their routing decisions.

The main information on the special cargo domain is available on the web pages of the freight forwarders and airlines. Various data sources use different templates, structures and languages to present the data. These sources are nonstandard and contain a large diversity in terminology. It is difficult to match the same information that is expressed in different ways. Furthermore, the absence of a resource terminology is a big challenge in practice. The massive volume of data makes manual information extraction costly. This research aims to develop deep neural network models with the least available samples in order to elicit the information from the special cargo domain. In particular, this study addresses the challenge of extracting special cargo domain knowledge in order to instantiate the ontology with the least human involvement and the lack of sufficient amount of training samples for developing a robust model. The experimental findings represent the effectiveness of the special cargo relation representation models in two tasks: relation extraction and few-shot relation matching. Few-shot learning is a machine learning approach that enables models to learn from a small amount of labeled data, typically with very few examples per class. It aims to address the challenge of limited labeled data availability by leveraging prior knowledge learned from related tasks or domains to quickly adapt and generalize to new tasks or domains with minimal supervision. Few-shot has received much attention in recent years in

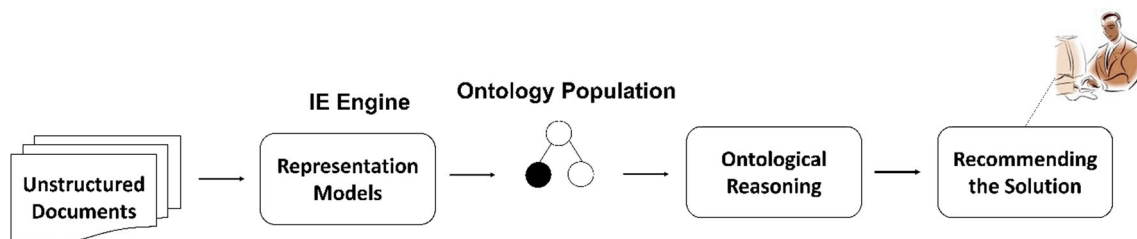


Fig. 1 A general schema for using of ontology in application. IE Engine assists to populate an ontology and utilize it in semantic reasoning

different domains such as fault diagnosis. MAMN [8] is an intelligent fault diagnosis model for addressing the problems of sparse fault samples and data cross-domain in data-driven fault diagnosis. The core of the method is to combine the relative similarity information of sample groups with the meta-information between data domains.

The main contributions of this paper are as follows:

- We propose a novel hierarchical attention-based multi-task architecture for relation representation learning in the special cargo domain and apply it in the multi-class relation classification task. The model is simple, robust, and domain-independent.
- Using transfer learning, we investigate leveraging a Bidirectional Encoder Representations from Transformers (BERT)-based relation extraction model as a feature extractor for multi-class relation extraction in the special cargo shipment domain.
- This study is one of the first in Information Extraction for transportation, particularly the special cargo shipment domain, that has utilized the deep neural models for multi-class relation extraction.
- We develop a new collection of datasets called Special Cargo Relations (SCR). It is made publicly available as evaluation datasets for multi-task relation extraction tasks in the special cargo shipment domain.
- The data set contains various examples for each relation type to train a relation classifier using the suggested relation representation models.

This paper is organized as follows. In the next section, we review the literature. Section 3 develops the proposed architecture in detail; Sects. 4 and 5 depict the steps of the datasets and results of the experiments, respectively; finally, Sect. 6 describes the practical implications of this work and Sect. 7 concludes the paper.

2 Related work

An explicit and formal specification of the underlying concepts of a domain is referred to as an ontology. The components of an ontology are concepts, relations, rules, and terms [9, 10]. Ontologies can be built manually by combining available ontologies, semiautomatically or automatically using the so-called ontology learning cake [11]. In an ontology, two types of elements exist: TBox and ABox.

TBox contains the concepts and relations of the ontology. ABox contains the realizations of concepts and the relations between them. They are real instances of the concepts specified in TBox. The process of updating an ontology based on the instances of input information source is called ontology population. When the extracted

information conforms to TBox, it is added to the ABox. Therefore, the ontology population does not affect the ontology structure.

Various ontologies are developed in some aspects of the transportation domain. The Transport Disruption Ontology¹ collects and models the travel data and helps in identifying events that have a disruptive impact on travel. Some ontologies [12] are modeled to support decision-making for drivers by specifying efficient routes for emergency vehicles. Different ontologies such as GenCLOn [13] and iCity Ontology² are designed to obtain the domain of city logistics and urban systems. Although various ontologies are designed in the context, populating ontology is a complicated task.

Many ontology population methods have recently been presented, ranging from basic rule-based and statistical approaches to complicated machine learning and hybrid architectures. Some of the main techniques are reviewed in this section.

Rule-based methods: These methods are the most common type of approach in the ontology population task. They rely on a number of prespecified rules to determine the structure of the target data. Hearst [14] presented the so-called scalable rule-based technique using the lexico-syntactic patterns produced with the bootstrapping of some seed samples. Since the approach extracts the hyponymy relations, it can be utilized to generate taxonomy relations in the ontology learning task.

Inspired by Hearst, Finkelstein and Yangarber [15] also proposed a semiautomatic rule-based method by using lexico-syntactic patterns to extract concept and relation instances from text. These systems utilize expert intervention for generating the rules based on some candidate patterns [15]. Ibrahim [16] is a semiautomatic ontology population method for extracting instances from biomedical text. Building the biomedical text syntactic parse tree of the sentences, lexico-syntactic patterns are generated by domain experts and used for identifying the concepts.

SOBA [17] is another example of automatic rule-based ontology population systems. The rule-based approaches require a thorough insight of the domain. The fundamental disadvantage of these approaches is the high necessity for human interaction [18].

Statistical-based methods: There are some statistical ontology population systems that use similarity measures and fitness functions to estimate the similarity degree of the extracted information and the ontology instances. Maynard et al. [19] proposed statistical domain-independent systems that extract the concept instances from unstructured text.

¹ <https://transportdisruption.github.io/>.

² <https://enterpriseintegrationlab.github.io/icity/UrbanSystem/doc/index-en.html>.

Another domain-independent statistical system is presented in [20] that uses statistical similarity techniques in order to determine the correct class for each extracted concept instance.

Machine learning-based methods: These models are commonly applied in populating ontologies and categorized into three major classes: supervised, weakly supervised, and unsupervised approaches [21].

Adaptiva [22] is a domain-independent semiautomatic ontology population system that uses seed samples of concepts and relations among them as input to recognize the sentences containing the examples of relations using the bootstrapping approach. Annie is a GATE component used for extracting concepts. Ontosophie [23] is a semi-automatic method that uses annotated data for relation types in a specific domain. By applying NLP tools, the main constituents of the sentences are used for inducing the extraction rules. The extractions are assigned a confidence value that shows the correctness degree of them. Web → KB [24] is a web-centric machine learning method that populates ontology using tools managing HTML pages. As the existing webpages are the rich source of information in different domains, this approach finds the webpages identified as the instances of the seed ontology classes. These webpages are linked by ontology relation with examining the hyperlink paths among the webpages. Three different Naive Bayes classifiers are used in order to classify the web pages to the most relevant class. The ontology relations are learned using a supervised first order inductive learning (FOIL) algorithm. Mintz [25] applies a distant supervision algorithm in order to find various patterns for different relation types.

Open information extraction (OIE) systems are not constrained to a predefined set of relations and can extract any type of relations from a massive and unstructured corpus automatically. There are various types of OIE extractors from shallow such as SONEX [26] to deep such as SDE-OIE [27]. The main difference between these two categories is related to the depth of the NLP tools they exploit. Deep extractors deal with sophisticated structures such as long-distance relations; therefore, they have high performance in terms of recall. OIE systems are designed based on various types of methods, including generic patterns [28], bootstrapping [29], supervised [30], and unsupervised learning [31]. These systems suffer from incoherent extractions and uninformative extractions.

Deep neural network-based methods: Deep neural networks are powerful approaches that have drawn much attention in recent studies of ontology population since the feature engineering procedure is done automatically. This is the key advantage of deep neural network models over other machine learning methods. In order to extract information at the lexical and contextual levels from sentences,

several approaches use relation classification and various deep neural models, such as convolutional deep neural networks [32]. The extracted features are then sent into a softmax classifier that predicts the target relation. All of the approaches require an annotated corpus with concepts and the relations between them.

MTB [33] is a task-independent approach for learning the representation from entity-linked English Wikipedia. The approach was derived from Harris' distributional theory and is based on BERT [34]. Recently, systems that are built with pre-trained models have attained high efficiency [35].

Some ontology population systems are a combination of machine learning and rule-based methods [36]. TR-DOE [10] and RV-DOE [10] are two hybrid methods that rely on the trade-off between shallow and deep OIE by using certain rules. BOEMIE [37] is a hybrid ontology population system that extracts information from different resources and applies reasoning to them. Figure 2 shows a taxonomy of ontology population methods.

Transfer-based deep neural networks have achieved remarkable success in many NLP tasks with limited data and are being widely used in different domains.

This research is an extension of our previous work [38], where we have proposed a framework for ontology population framework for the special cargo and examined an effective method for building an information extraction engine. In this paper, we present a very detailed description of the hierarchical attention-based multi-task architecture and the generated datasets. Moreover, we propose a new model for extracting information in the special cargo transportation domain using Matching the Blanks [33] method and investigate the performance of the proposed models in a new task namely, few-shot relation matching. The special cargo transportation domain suffers from the lack of data and can highly benefit from the strength of the novel transfer-based neural networks models. To our knowledge, this work is one of the first studies addressing IE in the special cargo domain that is highly important in the design and development of various intelligent applications.

3 Methodology

Special cargo ontology has many applications in the transportation and global freight forwarding of special cargo, route planning, traffic scheduling, risk assessment, and decision-making systems. Developing and populating an ontology for shipping special cargoes is challenging. It needs a description of concepts, properties, and relations among the concepts. The UPON [7] technique is used to produce a precise conceptualization of special cargo

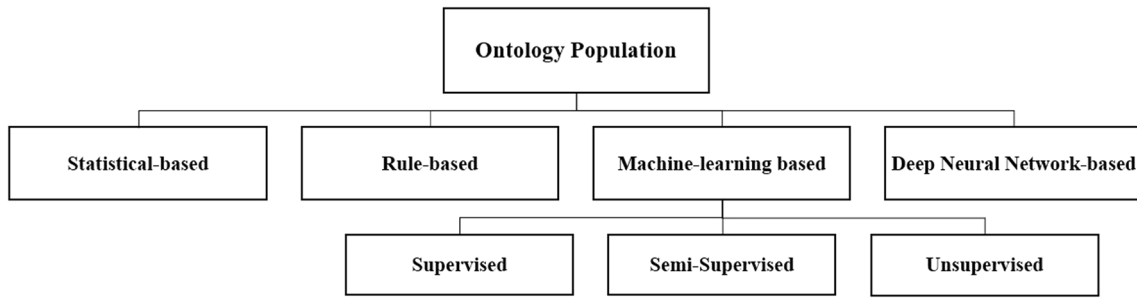


Fig. 2 A taxonomy of relation extraction methods in ontology population approaches. Deep neural network-based methods are being widely used in information extraction

transport in [39]. This effort uses elicitation approaches to organize the available domain information and obtain it from domain experts. Concepts should be instantiated using domain information to populate the cargo ontology. Webpages, online texts, and databases dedicated to cargo transportation include a wealth of important information. The required knowledge needs to be gathered from many sources and related coherently to populate the cargo ontology. As the available relation extractors heavily relied on the knowledge graph or ontology utilized in their design, and there is a great variation among them, their application in the special domain is challenging. Because of the lack of annotated data for the cargo shipment domain, constructing an effective relation extractor is challenging and time-consuming. The main goal of this paper is the relation representation task that is the core component in populating the special cargo in Fig. 3. We described the pipeline with more details in [38].

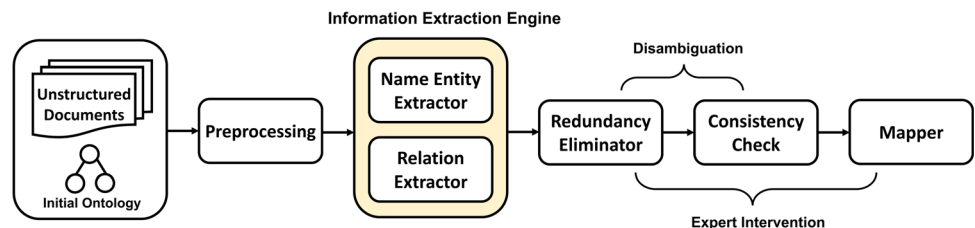
We suggest a multi-task representation learning approach to overcome this issue that is based on light tasks trained from open domain literature, with automated annotating and the least human intervention. The model is compared with a BERT-based relation representation model adapted with the special cargo domain. We leverage these two specific relation representation models to initialize a supervised multi-task relation classification model and tune them on a small special cargo domain dataset.

3.1 Hierarchical attention-based multi-task model for learning relation representation from special cargo domain

Existing pre-trained deep learning-based language models can extract different features and are employed in multiple downstream tasks [40]. The lack of training data in the special cargo domain is the most significant obstacle to developing an effective representation model. Manually generating train data for relation classification with multiple classes is too costly. Following [41, 42], we proposed a hierarchical-based approach built on some underlying tasks that encode features from shallow tasks in the bottom layers of the model and deep tasks in the top layers of the model. We can automatically build sufficient datasets for the underlying tasks with minimum human interventions and apply them in the multi-task representation model setting. Thus, building the model is simple because underlying tasks are trained using domain-specific data created automatically, that is not costly. Moreover, as multi-task models take advantage of inference transmission across tasks, resulting in complementing features in the generated embeddings and thus, improved generalization performance [42].

This multi-task learning model can be employed as a rich representation of special cargo domains in different tasks consisting of multi-class relation classification.

Fig. 3 The pipeline for populating the special cargo ontology [38]. Information extraction engine is the core of the special cargo ontology population task



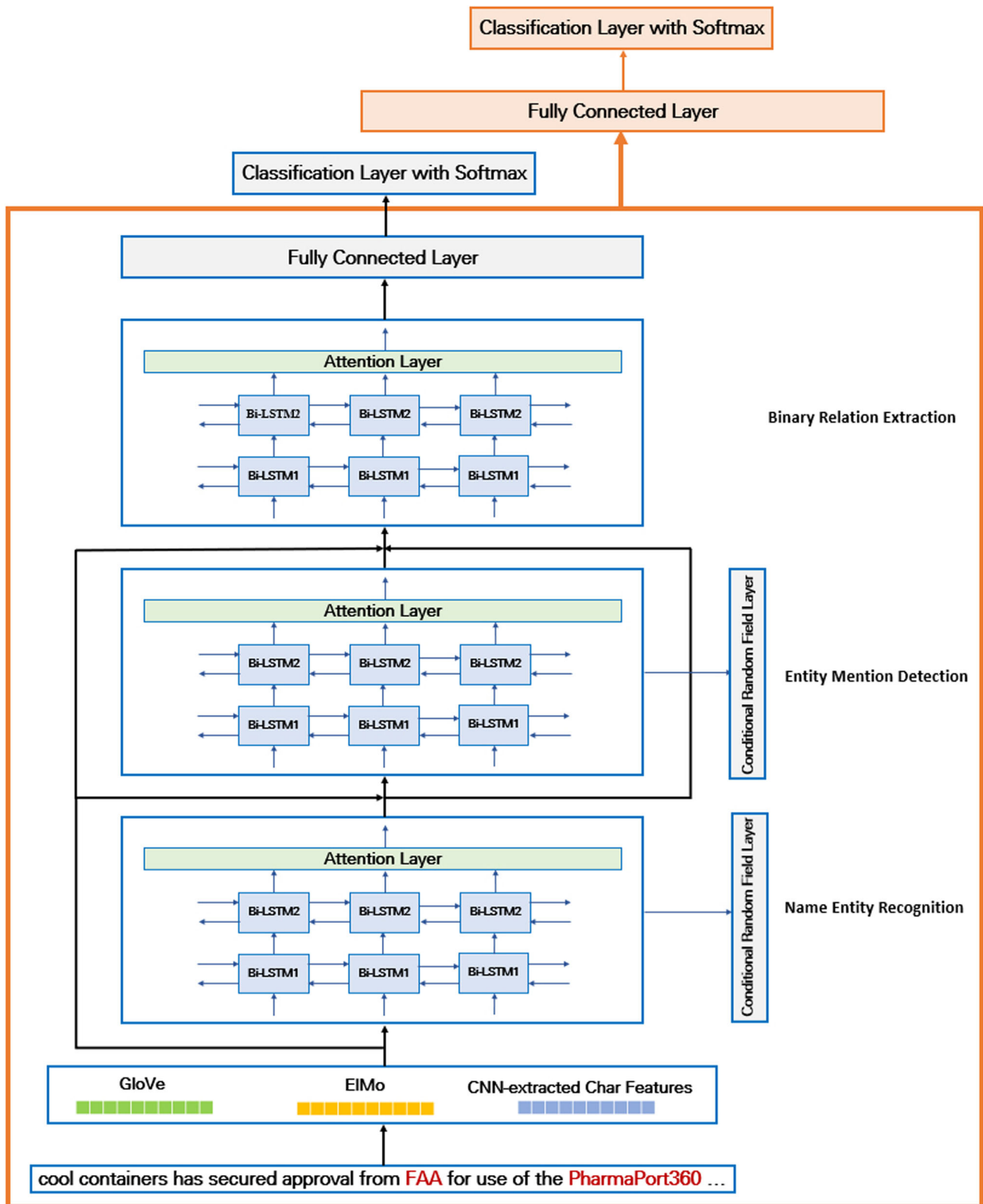


Fig. 4 Illustration of the proposed hierarchical model for learning relation representation in classifying relation types. The hierarchical-based approach is built upon some underlying tasks that encode

features from superficial tasks in the bottom layers of the model and complex tasks in the top layers of the model

Figure 4 shows the architecture of the multi-task representation learning model.

Word embedding: The input is a representation of the input sentence consisting of three various embedding approaches, namely GloVe [43] for word embedding, ELMo [44] for contextual embedding, and a model based on Convolutional Neural Network (CNN) [45] for character-level embedding. This yields a comprehensive representation of the input sentence that covers features from character to context level. Hence, each word w_t in the input sentence $s = (w_1, w_2, \dots, w_n)$ is encoded as g_e , that is the result of concatenating three different word embeddings. CNN-based character-level embedding is computationally efficient and facilitates the training procedure. We fine-tune the GloVe model with the domain data during the training of the proposed model. ELMo generates customized representation for the same input in various situations. Many successful results have been reported using ELMo in different NLP downstream applications before emerging BERT. To support a reasonable comparison with recent models of natural language (e.g., BERT), we do not apply such embedding models as the input of the proposed approach.

Name entity recognition (NER): The hierarchical model consists of three different shallow tasks. NER is the first underlying task with a conditional random field (CRF) for identifying NER tags. Name entity mentions are recognized from the embedding vector and classified into predetermined classes. The concatenated embedding vector is fed into a 2-layer BiLSTM network with an attention layer. The network encodes the input by taking the word embedding g_e and generating sequence embeddings g_{ner} . Then it is fed into a CRF-based sequence tagging layer.

The attention layer enables the encoding of the sequence data by allocating a significance value to every component. It generates a weighted vector multiplied by the vector of features obtained from LSTM in each time step [35]. Let H be a matrix of LSTM output vectors $[h_1, h_2, \dots, h_n]$, where n is the sentence length. A weighted sum of LSTM output vectors composes the representation r of the sentence [35].

$$M = \tanh(H) \quad (1)$$

$$\alpha = \text{softmax}(w^T M) \quad (2)$$

$$r = H\alpha^T \quad (3)$$

where $H \in R^{d^w \times T}$, d^w is word embedding size, w is a vector of trainable weights, and w^T is a transpose. The dimension of w , α , r is d^w , n , d^w , respectively. The representation for classification is attained by:

$$h^* = \tanh(r) \quad (4)$$

Entity detection (ED): It is another fundamental task in the proposed architecture that is similar to NER. In

comparison with NER, which focuses just on the name entities, ED is a wider solution and entails identifying all relevant references to a real-life entity. ED is defined as a sequence tagging problem that utilizes a 2-layer BiLSTM with an attention layer and a CRF layer. The representation vectors from the bottom layer are concatenated $[g_e, g_{ner}]$ and given to the encoder that generates representations g_{ed} .

Binary relation extraction: It is the process of extracting semantic relationships between name entities from textual data (RE). The relationship generally involves two or more entities and is traditionally considered as a classification problem. A binary relation classifier can determine if a pair of entities have a specific relationship. This task requires mention detection and classification, and we used a joint model developed in [46] that develops the subtasks together. Similar to NER and ED, binary relation extraction has a 2-layer BiLSTM with a layer of attention. The binary relation classification is trained with a fully connected layer and a classification layer with softmax.

The outputs of the preceding tasks are the inputs of each current task and the input embedding of the entire model. The encoder takes $[g_e, g_{ed}]$ as input and generates a representation indicated as g_{re} . A feed forward network receives these representations as input.

There appears to be no consensus on how to train a hierarchical multi-task model. We performed the successful training approach presented in [42]. For training the model, we randomly select a batch of training data for the target task in each iteration and the parameters of the task are updated. Sampling of tasks are done equally, and the procedure is repeated till convergence is achieved.

Multi-class relation extraction: The hierarchical multi-task architecture is trained using domain-specific data and used as a based model and as a means for extracting features for the other model, namely the multi-class relation classifier. The final layer of the hierarchical model was eliminated, and the rest of the layers were used to extract features for $SCRE_{HMTL}$.

We can leverage pre-trained models to transfer common features to our domain since there is not enough data in the target domain to train an efficient multi-class relation extractor. As a result, the performance of the target model that is trained using the base model and with only a few instances is effectively benefited from transferred features.

In a nutshell, the binary classifier utilizes the rich relation representation generated by a hierarchical attention-based multi-task model with adequate data samples in the special cargo domain. The ultimate configuration of the model is specified by applying the relation embedding obtained from the hierarchical architecture that operates as a way for extracting features for the multi-class extractor.

This paper is an extension of our previous work [38] and investigates two different transforming models as relation representations for extracting multi-class relation types.

3.2 Matching the blanks for learning relation representation from special cargo domain

The airfreight industry of shipping special cargo is a low-resource domain without annotated data. In order to decrease the need for human effort in generating datasets and also for comparison purposes, we build another relation representation model that relies on entity-linked text data.

We propose a relation representation method for special cargo domain based on the Matching The Blanks (MTB) [33] approach. MTB is Google’s state-of-the-art relation representation that significantly outperforms previous works in general web text. We model it with our relations and tune it on the special cargo domain for building $SCRE_{BERT(EM)+MTB}$ that is a multi-class relation extractor based on BERT.

We apply the BERT [34] model to encode the relations between entity pairs in the special cargo shipment domain. BERT is a language model built based on multiple transformer layers and self-supervised learning that utilizes a huge volume of corpus data to learn better feature representation of words. It has obtained state-of-the-art performance on a variety of NLP tasks.

The relation representation is trained without human annotation data using a plain text corpus with name entities linked to unique identifiers. A *relation statement* is defined as a segment of text containing two marked name entities and shown as (\tilde{r}, s_1, s_2) where \tilde{r} is a sequence of tokens and s_1 and s_2 are the spans of indices that determine two entities in the relation statement.

Because of the low-resource nature of the domain and availability of automatic entity resolution annotation systems for generating training samples, MTB approach is adapted to the special cargo domain by generating training data based on the relation statements with marked name entities replaced with a special symbol [BLANK], as shown in Fig. 5.

In the training process, MTB exploits a pair of relation representations for each pair of relation statements with the aim of learning an encoder f_θ that specifies if two

statements express the similar relation using the following binary classifier.

$$p(l = 1|r, r') = \frac{1}{1 + \exp f_\theta(r)^T f_\theta(r')} \tag{5}$$

The classifier allocates a probability when r and r' embed the same relation ($l = 1$) or not ($l = 0$).

Therefore, given a pair of relation statements, MTB tries to learn an embedding model that their inner product is high when both contain the same entity pair and low when entity pairs are different. In this way, the encoder is learned from distant supervision in the form of entity-linked text using the MTB method [33]. Figure 6 depicts the training process.

The parameters of the relation encoder f_θ are learned by minimizing the following loss. $\delta_{e,e'}$ is the Kronecker delta that assigns the vale 1 when $e = e'$, and otherwise 0.

$$L(D) = -\frac{1}{|D|^2} \sum_{(r, e_1, e_2) \in D} \sum_{(r', e'_1, e'_2) \in D} \delta_{e_1, e'_1} \delta_{e_2, e'_2} \cdot \log p(l = 1|r, r') + (1 - \delta_{e_1, e'_1} \delta_{e_2, e'_2}) \cdot \log(1 - p(l = 1|r, r')) \tag{6}$$

In the training setup, the loss of masked language models of BERT and MTB is minimized concurrently. Training corpus generation is described in Sect. 4. We test various training data and BERT models for training the relation statement encoder in this architecture.

Using the MTB pre-train model on domain data, the relation extraction model is built. The architecture of relation statement classification using MTB is illustrated in Fig. 7. As shown in the figure, the target entities are marked with special entity markers in the input. Then, BERT takes the marked sentence as input and the states related to the beginning of the two entity markers are joined and the relation representation retrieved.

The generated relation representation from the BERT transformer is fed into a fully connected layer. This layer is either the normalization of the relation representation or linear activation function. The layer type is selected as a hyper-parameter. The last layer is a classification layer with softmax activation that produces the probability of each class. These layers are trainable using a few instances for each type of relation.

Fig. 5 Two relation statements that share the same pair of name entities marked with [BLANK]

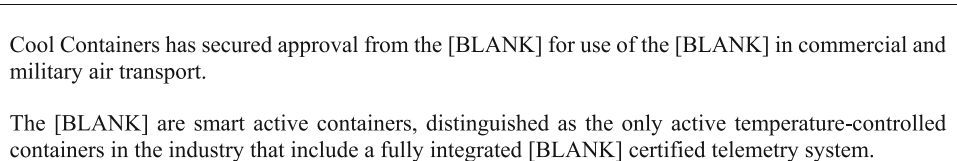


Fig. 6 An overview of the training process in MTB. The figure depicts that two different relation statements are fed into BERT. The classifier is defined to learn a relation encoder that is utilized to specify if two relation representations embed the same relation. Parameters of the encoder is learned by minimizing the loss function

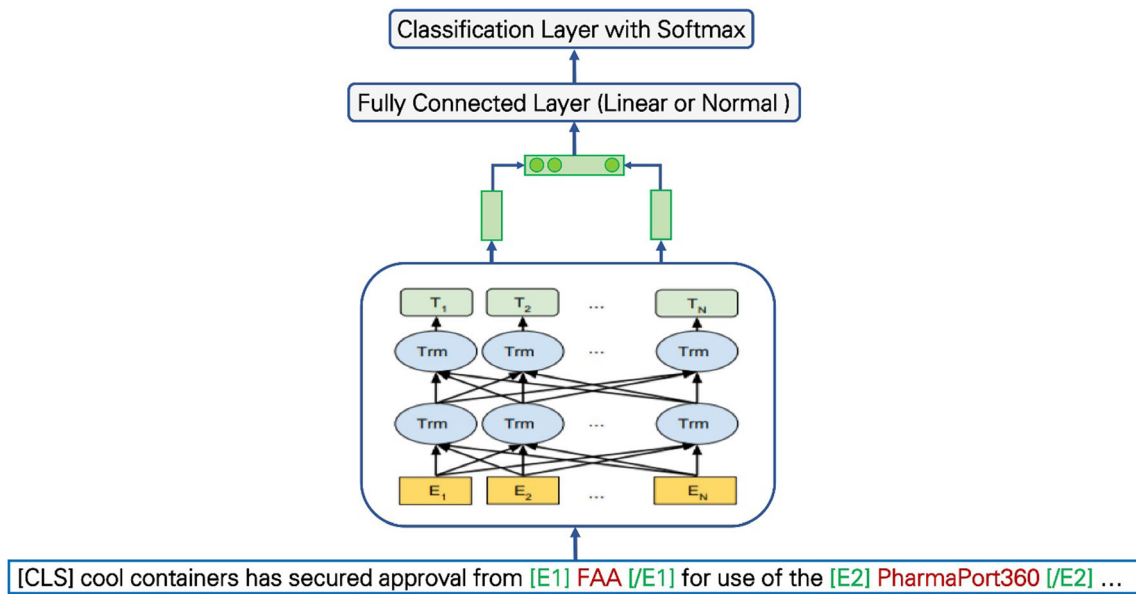
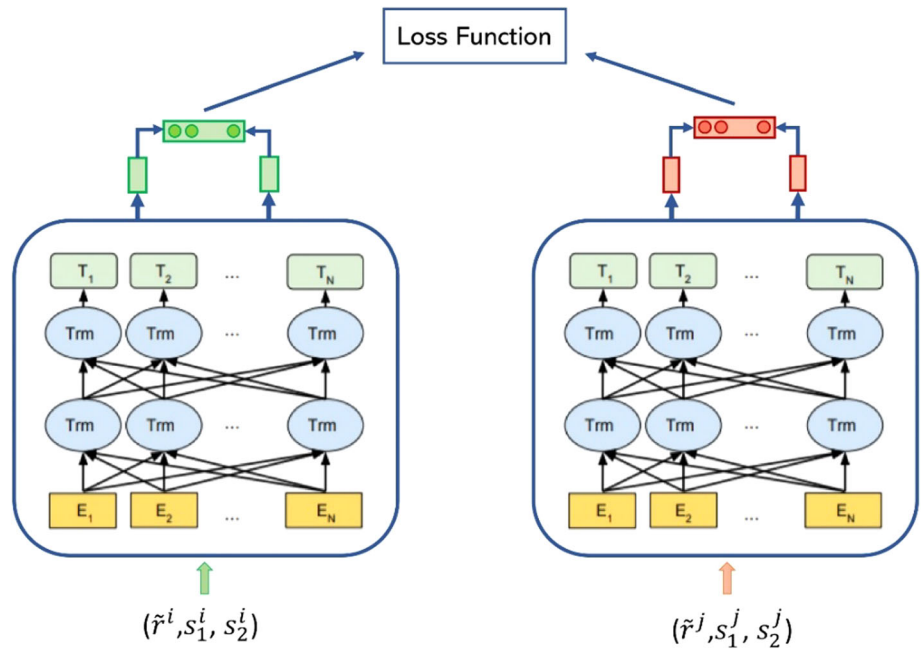


Fig. 7 Architecture of the special cargo relation classifier. Target entities in the input (FAA and PharmaPort360) are represented using special markers showing the start ([E1] and [E2]) and the end ([/E1]

and [/E2]) of each entity. The relation representation is returned after combining the states associated with the beginning of the two entity markers

4 Datasets

We utilize different resources for evaluation of the proposed models. These sources are discussed in greater detail in this section.

4.1 Train data for MTB

The training data for learning BERT-base relation are created by extracting the cargo domain relevant cargo news text pages and HTML paragraphs from cargo websites and eliminating tables and lists. In total, around 600,000 words were collected. We use Google API³ for annotating the

³ https://cloud.google.com/natural-language/docs/basics#entity_analysis.

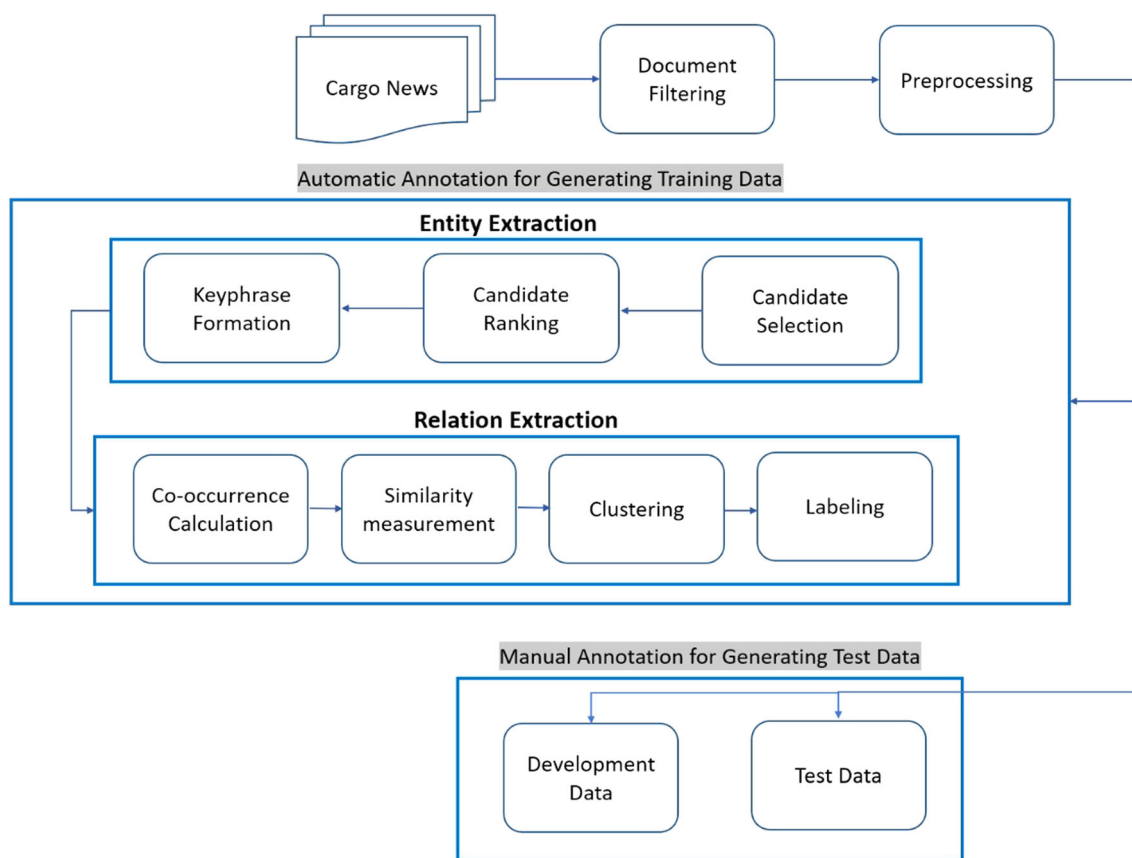


Fig. 8 An overview of the dataset generating for training the hierarchical attention-based multi-task model. Entity Extraction and Relation Extraction are the two main components of the automated annotating procedure

corpus text spans with a unique identifier. Relation statements containing at least two entities within a window of 40 tokens are extracted. These domain relation statements are utilized in the training procedure in which the entities are replaced with a [BLANK] notation.

4.2 Train/test data for the attention-based hierarchical multi-task model

The significant challenge is the lack of annotated resources in the special cargo domain. In this section, we explain how the automatic annotation is built to generate labeling for the underlying tasks in the attention-based hierarchical multi-task model. Figure 8 depicts the procedures involved in creating the training data.

We collected⁴ 28,809 cargo documents from news web pages that consist of formal texts to train the binary relation classifier. We employed an automatic filtering method to find all candidate documents associated with the special cargo domain. After applying the Latent Dirichlet Allocation (LDA) [47] topic model, we generated ten clusters. Unlike keyword-based methods, LDA does not need a

predetermined list of keywords for document filtering. The cluster with the more relevant terms is selected as the most domain representative. A threshold of 0.9 is used to guarantee that only documents with high relevancy are captured. In the end, 775 filtered texts were gathered.

Different NLP tasks, including sentence splitting, tokenization, NER, and part-of-speech (POS) tagging, are carried out in the preprocessing component. We adopted a fully automatic annotating approach for labeling data because deep learning models demand a large volume of training data, and manually producing this data is costly. Entity Extraction and Relation Extraction are the two main components of the automated annotating procedure. The entities are elicited from the texts using an unsupervised domain-specific entity extraction framework [48] and a NER tool [49].

The entity extraction task consists of three major modules. The Candidate Selection component uses heuristics to select candidate keywords from a list of approved POS. The candidate lexical terms are then prioritized depending on the text representation in the Candidate Ranking component. In the Keyphrase Formation, the rated candidate keywords are refined, and the final keyphrases are

⁴ We use the *requests* package and *Beautiful Soup* library.

generated. We utilized pke [50], an open-source library in python for keyphrase extraction that includes a variety of statistical and graph-based techniques. According to our experiments, KPM [51], a statistical-based technique, and PositionRank [52], a graph-based approach, outperform the other algorithms for extracting specific cargo keyphrases.

Figure 8 depicts the four primary phases of a general clustering-based relation extraction technique. The input of the Relation Extraction task is all sentences from the Entity Extraction component, as well as labeled entities. The co-occurrence calculation step extracts a pair of name entities and their context that happens within a defined window size. The contexts of the name entities pair that occur simultaneously are considered in terms of similarity. This is necessary for the clustering task. We use the so-called fast Levenshtein [53] as a similarity measurement calculation of the contexts based on the minimum edit distance of the two strings. We employ the DBSCAN [54] clustering algorithm that is not dependent on the predefined number of clusters. Based on our evaluation, since KPM obtains the highest performance, we apply it as the key extraction method. The Labeling component labels the relation clusters relative to the special cargo domain. The clusters are labeled as relevant or irrelevant using patterns from the annotated development set.

We randomly selected 223 documents from the filtered collection in order to annotate them for the binary classifier as relevant or irrelevant and generate test data. 118 documents are deleted due to ineligibility, and the rest are utilized in the Dev and Test collection. Besides, we randomly selected a dataset of 10 online domain documents. Table 1 displays the data statistics. Although all of the documents are news articles, in comparison to the development set and test set 1, the training set has more sentences per document. Test set 2 (online documents), on the other hand, includes more sentences and entities per document than the training set, yet the total number of words in each sentence is small. It shows in comparison to the news documents, online documents have a larger density of related information.

4.3 Train/test data for multi-class relation classification model

In the cargo ontology, there are 43 distinct relation types. There are two possible argument ordering for each of the relation types.

The relation types are chosen based on the wide coverage of the relations in the special cargo ontology [39]. Table 2 shows an instance of the relation type and an example sentence for it. The dataset is in the standard

Table 1 Statistics of the datasets generated based on the proposed pipeline for training the attention-based hierarchical multi-task model [38]

	Train set	Dev set	Test Set1	Test Set2
Total documents	552	53	52	10
Total sentences	8361	556	548	221
Sentences per document	15.15	10.49	10.54	22.10

format of SemEval-2 Task8 [55] and publicly available on GitHub.⁵

Existing datasets usually comprise hundreds of samples for their target relation types. In specific domains such as special cargo transportation, generating a dataset is costly. We selected and annotated only a few sample sentences for each relation type. Two domain experts independently annotate each sentence. The experts agree on 87% of the instances. We apply the agreed subset of the data in our experiment for assessing both multi-class classifiers. Table 3 shows the statistics of the datasets.

In the SCR_S dataset, each relation type has only a few instances. SCR_L is generated by increasing amount of relation type instances in SCR_S. We build two datasets, namely SCR_{SM} and SCR_{LM} by combining similar classes in SCR_S and SCR_L datasets. The datasets have the least semantic overlap across the classes and serve as coarse-grained representations of the domain classes.

5 Experiments

In this section, we evaluate the effect of proposed relation representation models developed on the hierarchical multi-task model and BERT in the performance of the special cargo multi-class relation classifier. We present the results of the experiments using *F1* score on the datasets described in Sect. 4.3.

5.1 Performance measures

F1 and Accuracy are widely used performance metrics in the evaluation of Relation Extraction and FewRel tasks [33, 56, 57]. Accuracy measures the overall correctness of the extracted relations in the special cargo domain which is important in evaluating the quantitative measure of the model performance. Accuracy is calculated as the ratio of

⁵ <https://github.com/VahidehReshadat/CargoRelationExtraction>.

correct predictions to the total number of predictions made across all classes. We calculate the accuracy of the model across different few-shot settings, such as 1-shot or 5-shot classification or with different ways of generating support sets.

$$\text{Accuracy} = \frac{\text{Total number of the correctly classified positive and negative relation instances}}{\text{Total number of relation instances}} \tag{7}$$

$$\text{Recall} = \frac{\text{Number of correctly extracted relation instances}}{\text{Total number of extracted relation instances}} \tag{9}$$

Accuracy alone is not a sufficient indicator in the special cargo information extraction task as it does not take into account the incorrectly identified relations and non-identified relations by the special cargo relation extractors. *F1* score, on the other hand, considers the harmonic mean of precision and recall which makes it a better evaluation metric for information extraction in our domain. Precision measures the fraction of the number of correctly extracted positive relation instances to the total number of extracted positive relation instances. Recall measures the fraction of the number of correctly extracted relation instances to the total number of relation instances in the dataset. *F1* gives equal weight to both metrics. Therefore, it provides a balanced evaluation of the classifier’s performance with respect to both precision and recall.

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

5.2 Evaluation of the BERT-based multi-class relation classifier for the special cargo domain

The performance of the relation classifier over the relation representation obtained from BERT and BERT + MTB is measured by applying SCR datasets. The hyper-parameters of the model that are used in the experiments are shown in Table 4. This setting is employed for the experiments to perform a fair comparison.

The result of training the classifier with/without MTB is displayed in Table 5. We examine the classifier based on the BERT-Base and BERT-Large architectures. Since the concatenation of the corresponding representation of the beginning entity markers is used, we notate these versions as $\text{SCR}_{\text{BERT(EM)}}$. Then, the results are compared with the variants of BERT trained with domain data using the MTB

Precision

$$= \frac{\text{Number of correctly extracted positive relation instances}}{\text{Total number of extracted positive relation instances}} \tag{8}$$

Table 2 Examples of relation types and sample sentences that are used as input of the proposed models

< <i>e</i> ₁ > Brussels Airport < / <i>e</i> ₁ > signs MoU for the seamless transportation of < <i>e</i> ₂ > COVID-19 vaccine < / <i>e</i> ₂ >	Ships (<i>e</i> ₁ , <i>e</i> ₂)
< <i>e</i> ₁ > Active Pharma Ingredients < / <i>e</i> ₁ > is required to be pre-cooled within the desired temperature between < <i>e</i> ₂ > −25 and −15 °C < / <i>e</i> ₂ > before loading by the shipper	HasTemperatureRange (<i>e</i> ₁ , <i>e</i> ₂)
The < <i>e</i> ₁ > Va-Q-Tec containers < / <i>e</i> ₁ > keep the < <i>e</i> ₂ > Pfizer/BioNTech vaccines < / <i>e</i> ₂ > cool enough	isPackedIn (<i>e</i> ₂ , <i>e</i> ₁)

Table 3 Statistics of the dataset generated for training relation classifier with multiple classes

	SCR _S		SCR _L		SCR _{SM}		SCR _{LM}	
	Train set	Test set	Train set	Test set	Train set	Test set	Train set	Test set
#Annotated samples	230	104	420	104	180	80	310	80
#Relation types	43	43	43	43	19	19	19	19

The table shows the number of annotated samples and relation types for each train and test set

Table 4 The hyper-parameter configurations used in BERT-based multi-class relation classifier

Epochs	55
Learning rate	0.0005
Batch Size	32
BERT-base architecture	12-layer, 768-hidden, 12-heads, 110 M parameters
BERT-large architecture	24-layer, 1024-hidden, 16-heads, 340 M parameters

Table 5 Evaluation results (*F1*) on special cargo test data for multi-class relation classification built on different BERT configuration

	SCR _S		SCR _L		SCR _{SM}		SCR _{LM}	
	BERT-base	BERT-large	BERT-base	BERT-large	BERT-base	BERT-large	BERT-base	BERT-large
SCRE _{BERT(EM)}	49.5	51.73	52.88	53.92	62.5	63.82	67.09	68.54
SCRE _{BERT(EM)+MTB}	49.97	52.3	53.32	54.68	63.4	65.21	69.14	69.88

According to the assessments, SCRE_{BERT(EM)+MTB} outperforms SCRE_{BERT(EM)} in terms of *F1*-score

Table 6 Evaluation results (*F1*) of the multi-class relation classification built on BERT with non-direction datasets

	SCR _S		SCR _L		SCR _{SM}		SCR _{LM}	
	BERT-base	BERT-lLarge	BERT-base	BERT-large	BERT-base	BERT-large	BERT-base	BERT-large
SCRE _{BERT(EM)}	53.39	54.87	55.07	56.42	65.38	67.23	71.05	72.68
SCRE _{BERT(EM)+MTB}	54.98	55.39	56.13	57.74	66.26	69.12	72.94	74.14

The performance of both models is improved in comparison with the case where the direction of the relations is taken into account

approach. These versions are notated as SCRE_{BERT(EM)+MTB}.

Based on the evaluations, SCRE_{BERT(EM)+MTB} achieves a higher *F1*-score when compared with SCRE_{BERT(EM)}. This improvement is not significantly different since the generated corpus for training the model is not large enough to train the MTB model. The performance of the model on the merged relation instances is high since unifying the similar relation instances strengthens the classifier’s distinguishing ability. Moreover, utilizing BERT-Large leads to an improvement in performance due to it being trained on a massive dataset. Therefore, applying a large dataset with coarse-grain relations using BERT-Large transformer yields the highest performance.

The relation samples in the dataset are in bidirectional setup. For instance there are two different variations for Ships relation, one that shows e_1 entity ships e_2 entity (Ships(e_1, e_2)), the other illustrates e_2 entity ships e_1 entity (Ships (e_2, e_1)). We measure whether the model is able to detect relations correctly, regardless of the direction. Table 6 shows the assessment of the model without directions. The performance of the model is improved when we compare it with the case where the direction of the relations is taken into account. In this situation,

Table 7 The configuration of the hyper-parameters in the hierarchical multi-task-based multi-class relation classifier

Epochs	20
Learning rate	0.0005
Dropout rate (Embedding)	0.5
Dropout rate (LSTM)	0.2
Batch size	32
Word embedding dimension	100
Char embedding dimension	16

performance is calculated using instances classified in the proper class but in a different direction. In both experiments over the directional and non-directional datasets, we see that there is not a large gap between the performance of the model trained on the large dataset that has almost two times more samples than the small dataset. Furthermore, even applying BERT that solely relies on the general domain texts yields reasonable performance. This shows that using representation learning is effective for low-resource domains.

5.3 Evaluation of the special cargo multi-class relation classifier developed on hierarchical multi-task model

In this section, the efficiency of the multi-class relation classifier is examined using the representation developed from the hierarchical multi-task model on the SCR datasets. Table 7 lists the hyper-parameters of the model utilized in the evaluations. The same setting is used in all evaluations to ensure a fair comparison.

In our implementation, besides ELMO⁶ [44], GloVe⁷ [43], and CNN⁸ [45] for embedding the input, the proposed hierarchical attention-based multi-task architecture consists of a CRF-based sequence tagging layer and a BiLSTM with an attention layer. We adapted a PyTorch implementation Matching The Blanks⁹ (MTB) [33] for implementing the Bert-base relation extractor for the special cargo domain. We applied Python (3.6 +), PyTorch (1.2.0 +), and Spacy (2.1.8 +) in our implementation. Regenerating the models by other researchers is also easy as it is clear how the data is prepared, and classifiers are trained and implemented. The code and datasets are also available in GitHub¹⁰ platform for verification and reusability. It is also possible to easily customize and implement the model for similar problems in other domains and generate the data automatically.

The hierarchical model is employed for feature extraction in the special cargo domain and generates a rich feature representation for the multi-class relation classifier. Table 8 shows the experimental results of the classifier trained with features elicited from the base model. The model performance in classifying both coarse-grain and fine-grain relations is promising.

The performance of the multi-class relation classifier is promising since it is trained with the inductive transfer; however, the evaluation result is not as high as that of the BERT-base model as BERT has used a very large corpus for pre-training. The dataset used for training the hierarchical multi-task representation model is quite smaller than that of BERT; therefore, increasing the amount of domain data will produce higher performance.

In order to assess the efficiency of the model over undirected relations, a new set of experiments are performed. Table 9 illustrates the results. The results demonstrated that in comparison with the prior scenario, the

⁶ <https://github.com/allenai/allennlp-models>.

⁷ <https://nlp.stanford.edu/projects/glove/>.

⁸ <https://github.com/kamalkraj/Named-Entity-Recognition-with-Bidirectional-LSTM-CNNs>.

⁹ <https://github.com/jpablou/Matching-The-Blanks-Ths>.

¹⁰ <https://github.com/VahidehReshadat/SCRE> and <https://github.com/VahidehReshadat/CargoRelationExtraction>.

Table 8 Evaluation results (*F1*) for multi-class relation classification developed on hierarchical multi-task model

	SCR _S	SCR _L	SCR _{SM}	SCR _{LM}
SCRE _{HMTL}	43.31	46.25	55.73	56.48

The multi-class relation classifier has demonstrated promising performance

Table 9 Evaluation results (*F1*) for multi-class relation classification developed on hierarchical multi-task model over the non-direction datasets

	SCR _S	SCR _L	SCR _{SM}	SCR _{LM}
SCRE _{HMTL}	46.72	47.18	59.26	60.88

The results demonstrated that in comparison with original dataset, the model achieves a higher *F1*-score across different datasets

Table 10 Accuracy for FewRel few-shot relation classification

	5 way 1 shot		10 way 1 shot	
	HMTL	BERT	HMTL	BERT
SCR _S	57.82	66.17	51.42	56.07
SCR _L	74.68	82.05	64.52	73.07
SCR _{SM}	53.84	61.94	46.75	52.66
SCR _{LM}	59.01	76.53	57.2	65.36

The BERT-base classifier outperforms the Hierarchical Multi-task learning-based classifier

Table 11 Accuracy for FewRel few-shot relation classification over the non-directional dataset

	5 way 1 shot		10 way 1 shot	
	HMTL	BERT	HMTL	BERT
SCR _S	40.1	46.44	35.4	41.2
SCR _L	54.73	63.88	52.68	58.3
SCR _{SM}	35.48	42.44	32.14	38.93
SCR _{LM}	48.21	55.71	41.2	49.77

The BERT-base classifier outperforms the Hierarchical Multi-task learning-based classifier

model obtains a higher *F1*-score across various datasets. One probable explanation is that determining the proper class regardless of name entities order solely improves performance.

The experiment results also support this argument that training by hierarchical multi-task model can significantly reduce the human effort for generating domain training data and consequently relation extractors creation and knowledge base population.

5.4 Few-shot relation matching

In few-shot relation matching, given a query relation statement, the candidate relations are ranked and matched. We evaluate the few-shot relation matching task on the SCR dataset. In this case, the dot product is applied as a similarity score between the relation representation of the query and each of the candidate statements.

Few-shot learning is usually studied using N-way-K-shot classification. Here, we aim to discriminate between N classes with K examples of each. Tables 10 and 11 show the evaluation results of the few-shot relation classification task on the two different variants of the datasets.

The BERT-base classifier outperforms the Hierarchical Multi-task learning-based classifier. This is probably because BERT employs a large dataset in the training phase. The evaluation results show the performance increases either when the diversity of relation types decreases or the number of training samples for each relation type increases.

6 Discussion and practical implications

The experimental findings represent the effectiveness of the special cargo relation representation models in two tasks: relation extraction and few-shot relation matching. The special cargo relation extractors provided comparable performance over the different datasets. Learning relation representation through pre-trained models solely has a great effect on the efficiency in the special cargo domain, and the influence of increasing the quantity of the training data on performance is not impressive. This noticeably reduces the human effort in domain-specific tasks.

Since the multi-task hierarchical architecture benefits from the inductive transmission within tasks, it provides a complementing representation from the lowest level to the highest level of the model. This results in an easy feature flow from the bottom to the top of the architecture that accelerates the training process.

We can automatically build sufficient datasets for the underlying tasks from domain-specific texts with minimum human interventions and apply them in the multi-task representation model setting. Because underlying tasks are trained using only automatically generated domain-specific data, building the model is effective, simple, and inexpensive. This architecture can be easily adapted to a low-

resource target domain. The whole model can be trained end-to-end without any external linguistic tools or hand-engineered features.

Natural Language Processing (NLP) tasks can be broadly categorized into syntactic and semantic tasks based on the linguistic analysis used in them, which involve different aspects of language analysis and understanding [58, 59]. Semantic tasks refer to a range of tasks that involve understanding the meaning of human languages such as Named Entity Recognition (NER) and Relation Extraction. Syntactic tasks, on the other hand, focus on analyzing the grammatical structure and the syntax of sentences, such as Part-Of-Speech tagging and Parsing [60]. Hierarchical architectures are most effective when their constituents' tasks are related [61]. Since the final target task is a semantic task, the proposed multi-task hierarchical architecture is designed on a set of semantic tasks. The model aims at integrating a set of semantic tasks ranging from NER to Binary Relation Extraction into a single architecture.

The number of layers in the multi-task hierarchical model depends on the complexity of the target classification task. We design the layers based on the complexity of our target task, namely the multi-class relation extraction. The linguistic hierarchies between NLP tasks are discussed in [41, 60]. The “Low-level” tasks are simple and require a limited amount of modification to the input of the model while other “higher-level” tasks require a deeper processing of the inputs and likely a more complex architecture. It is argued in [61] that low-level tasks that are assumed to require less knowledge and language understanding are better kept at the lower layers, enabling the higher-level tasks to make use of the shared representation of the lower-level tasks. In principle, the linguistic levels of semantics tasks would benefit each other by being trained in a single model. We incorporate this knowledge in the hierarchical architecture and design 3-levels of multiple tasks using the concept of the linguistic hierarchies of NLP tasks in [41, 60]. Thus, the lower-level semantic tasks (e.g., NER) affect the representation of the higher-levels (e.g., Binary Relation Extraction) ones in a seamless cascaded way. The shared embeddings and stacking hierarchical encoders allow us to share the supervision from each task along the full structure of our model and achieve state-of-the-art performance.

The attention layer in a BiLSTM model is used to selectively focus on certain parts of the input sequence that are most relevant to the task at hand. The attention layer uses a mechanism that assigns attention weights to each time step of the input sequence, based on how important they are for predicting the output. These weights are then used to compute a weighted sum of the input sequence, which captures the most important information. The

required attention layers in our model are selected based on the number of tasks in the designed architecture. They are put on top of each task to convey important features in a hierarchical way and function as a connector to the other tasks. This attention mechanism helps the model to better understand the relationships between different parts of the input sequence, which can improve the accuracy of the model's predictions. Attention connections are key in the performance of the model, as they feed important parts of the sequences to the higher-level tasks and help them to encode the rich representation.

The proposed approach is improved and operated at high performance using BERT. Relation extractors built over BERT-based relation representation models outperform the hierarchical multi-task representation model. One potential reason is the size and domain of the training data used in BERT training. BERT is trained on an enormous corpus with both general and domain-specific texts, whereas the hierarchical multi-task model is based primarily on a limited dataset of domain-specific texts. While the supervised multi-task learning model can encode a rich complementary representation of the linguistic features for input sentence through leveraging different underlying tasks in the hierarchical setting, it seems still the size of training data makes it difficult for SCRE_{HMTL} to achieve substantial results. Hence, it is more likely that increasing the size of training data or exploiting a rich encoding resource (e.g., BERT and GPT-3 [62]) in the model architecture leads to an efficient hierarchical multi-task model over BERT.

In terms of computational point of view, the proposed models are computationally efficient and cost-effective. The training time of the best model on 4 × Titan V, 12 GB HBM2 with 12 cores took less than 20 min. Due to the simple structure and the small training dataset, training the hierarchical multi-task model is easier than BERT-based models and needs fewer data and human effort. Thus, they are efficient in any domain with limited training data such as special cargo shipment. In hierarchical attention-based multi-task architecture, we employ Convolutional Neural Network (CNN) for extracting the character-level features because CNNs perform similarly to Recurrent Neural Networks (RNNs) in terms of performance, but RNNs are computationally more expensive to train, as reported by [63].

Training is sped up in the multi-task setting [42]. This means that training time for each task in the hierarchical model is reduced in comparison to the single-task model training framework. Thus, the processing time required to execute the task is shortened. Namely, the count of updates required to achieve convergence is reduced. This supports the insight that the features obtained from one task are conveyed to the other tasks within the hierarchical model.

For obtaining a domain-specific relation embedding, both relation representation models are trained using domain-specific training corpora generated with least human intervention. Increasing the amount of training data does not provide a significant boost in the performance of the models. Thus, because of the fact that little human participation is involved in training these relation representation models, they are efficient in any domain with small training data such as special cargo shipment.

This research sheds light on the development and design of logistic knowledge bases as well as the methods for extracting domain-specific data. As a structured resource, populating special cargo ontology provides valuable insights into the scope of the application, the different components of the system, and the interaction between them. Furthermore, the ontology can be used during the actual operation of the system.

As an example, the fact that Pfizer vaccines need to be shipped at specific temperature can be reasoned from the special cargo ontology, which means that, when the freight forwarder processes a request for shipping vaccines, the system can determine that the cargo service needs to contain temperature requirements for vaccines to be shipped.

Figure 9 illustrates an example of how the cargo ontology can be included in the context of a logistics intelligence system and helps the planning of special cargo shipments. When a shipping request arrives, the freight forwarder sends a query to the special cargo intelligent system in order to find the shipment requirements. The proposed relation representation models can be part of the knowledge base and applied as information extractors for extracting the instances (Pfizer, Va-Q-Tec, −80 °C) for the concepts of the special cargo ontology (Container, Pharmaceutical, Temperature) to populate the ontology and employ it in semantic reasoning. Our approach processes the input domain information as explained in Sect. 3.2 and output the triples in the form of entities and the relation between them (Pfizer, IsPackedIn, Va-Q-Tec), (Va-Q-Tec, HasTemperature, −80 °C) based the relations defined in the special cargo ontology *IsPackedIn*, *HasTemperature*.

The special cargo ontology is used to infer that vaccines must be transported at a specified temperature. Unlike a database, ontology is not a schema that only stores the desired relations and then retrieves them. It can be used for asking questions that are not explicitly expressed in the ontology. The fact that *Pfizer is stored at −80 °C* is not explicitly expressed in the ontology and is acquired by inferencing knowledge from existing facts based on certain rules and constraints namely: *Pfizer is packed in Va-Q-Tec*, *Va-Q-Tec has temperature −80 °C*, *Covid19 vaccines are stored in a special temperature*. The result of reasoning helps freight forwarders in decision-making in planning.

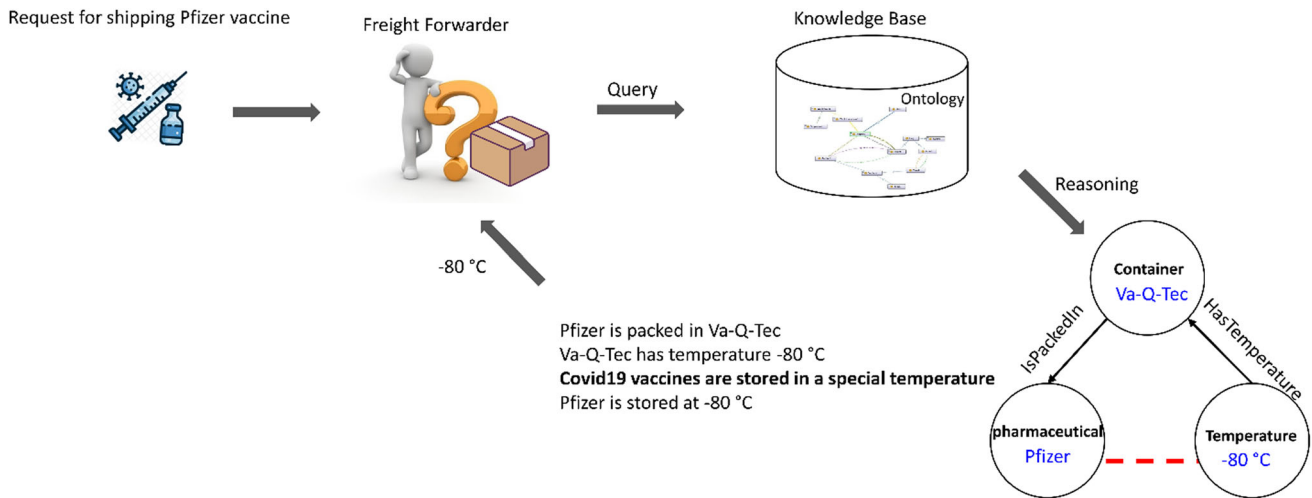


Fig. 9 An instance of applying the special cargo ontology in planning cargo shipments. The proposed relation representation models can be applied as information extractors for extracting the instances for the concepts of the special cargo ontology

As a result, this novel data analytics model can play a major role in the freight forwarding industry by providing a set of solutions for many intelligent applications in logistics. This is currently quite challenging for shippers and forwarders, due to the intricate characteristics of special products (e.g., different types of pharmaceutical products such as COVID-19 vaccines) and the absence of capability and service standardization supplied by airfreight companies. Applying the models similar to the proposed ones is essential in acquiring and modeling of logistics and cargo knowledge due to addressing these limitations and optimizing the solutions for the organizational issues.

7 Conclusion

The special cargo airfreight, which transports commodities that require special handling, is in desperate need of digital innovation. With this paper, we aim to bring innovation in order to find a robust approach to elicit and model relevant information.

Training an efficient model with limited data is challenging. Lack of specific domain data makes the problem worse. In this paper, we proposed two different relation extraction approaches for the special cargo domain based on two different representation learning models namely, BERT and hierarchical multi-task architecture. BERT-based relation representation solely relies on entity-linked text, and applying MTB training setting yields enriched domain-specific relation representation. The multi-task relation representation model is built on different tasks from deep at the top layers to shallow at the bottom layers in a hierarchical setting. The generated representations are then integrated into a new classifier, and

the final model is trained. The proposed models can be applied for classifying the relations in populating the special cargo domain ontology.

In order to evaluate the proposed models, we developed some datasets in the special cargo domain. The experiment results demonstrate that when dealing with small training data in a specific domain, applying proposed models is even more effective than increasing data size. Thus, the proposed models are particularly useful in low-resource environments and minimize human intervention for generating information extraction datasets in specific domains.

This work is one of the first studies that investigate multi-class relation extraction in the special cargo domain. Since the architectures are independent of the domain, they could be used in other domains. In the future, we intend to use external knowledge resources as well as our relation embeddings. In addition, we will consider how well the attention-based multi-task model performs in domain tasks like co-reference resolution.

Acknowledgements TKI Dinalog funded this research as part of the “LARA: Lane Analysis & Route Advisor” project (1002655LARA). Validaide provided domain expertise and essential data in special cargo, which the authors gratefully acknowledge. With the help of SURF Cooperative, this work was done on the Dutch national e-infrastructure.

Data availability The datasets generated during the current study are available in the <https://github.com/VahidehReshadat/CargoRelationExtraction/tree/master/dataset-SCR>.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Kersten W, Blecker T, Ringle CM (2017) Digitalization in supply chain management and logistics: smart and digital solutions for an industry 4.0 Environment: Berlin: epubli GmbH.
- Shavarani HS, Sarkar A (2021) Better neural machine translation by extracting linguistic information from bert. arXiv preprint [arXiv:2104.02831](https://arxiv.org/abs/2104.02831).
- Vryzas N, Vrysis L, Kotsakis R, Dimoulas C (2021) A web crowdsourcing framework for transfer learning and personalized Speech Emotion Recognition. *Mach Learn Appl* 6:100132
- Mishra S, Coaplen J, Tomizuka M (2007) Precision positioning of wafer scanners segmented iterative learning control for non-repetitive disturbances [applications of control]. *IEEE Control Syst Mag* 27:20–25
- Tao H, Qiu J, Chen Y, Stojanovic V, Cheng L (2023) Unsupervised cross-domain rolling bearing fault diagnosis based on time-frequency information fusion. *J Franklin Inst* 360:1454–1477
- Minakais M, Mishra S, Wen JT (2014) Groundhog day: Iterative learning for building temperature control. In: IEEE international conference on automation science and engineering (CASE) 2014:948–953
- De Nicola A, Missikoff M, Navigli R (2005) A proposal for a unified process for ontology building: UPON. In: International conference on database and expert systems applications, pp 655–664.
- Tao H, Cheng L, Qiu J, Stojanovic V (2022) Few shot cross equipment fault diagnosis method based on parameter optimization and feature metric. *Meas Sci Technol* 33:115005
- Buitelaar P, Cimiano P, Magnini B (2005) Ontology learning from text: An overview. *Ontology learning from text: Methods, evaluation and applications*. Front Artif Intell Appl Series, p. 123.
- Reshadat V, Hoorali M, Faily H (2016) A hybrid method for open information extraction based on shallow and deep linguistic analysis. *Interdiscip Inf Sci* 22:87–100
- Gómez-Pérez A, Manzano-Macho D (2003) A survey of ontology learning methods and techniques. *OntoWeb Deliverable D*, vol. 1.
- Bermejo A, Villadangos J, Astrain JJ, Córdoba A (2013) Ontology based road traffic management. In: *Intelligent distributed computing VI*. Springer, Cham, pp 103–108.
- Anand N, Yang M, van Duin JR, Tavasszy L (2012) GenCLON: an ontology for city logistics. *Expert Syst Appl* 39:11944–11960
- Hearst MA (1992) Automatic acquisition of hyponyms from large text corpora. In: *Coling 1992 volume 2: The 15th international conference on computational linguistics*.
- Finkelstein-Landau M, Morin E (1999) Extracting semantic relationships between terms: Supervised vs. unsupervised method. In: *International Workshop on Ontological Engineering on the Global Information Infrastructure*, pp. 71–80.
- Ibrahim Z, Noah SA, Noor MM (2010) Rules for ontology population from text of Malaysia medicinal herbs domain. In: *International conference on rough sets and knowledge technology*, pp 386–394.
- Buitelaar P, Cimiano P, Racioppa S, Siegel M (2006) Ontology-based information extraction with soba. In: *Proceedings of the international conference on language resources and evaluation (LREC)*.
- Cui M, Li L, Wang Z, You M (2017) A survey on relation extraction. In: *China Conference on Knowledge Graph and Semantic Computing*, pp. 50–58.
- Maynard D, Li Y, Peters W (2008) NLP techniques for term extraction and ontology population, pp. 107–127.
- Yoon HG, Han YJ, Park S-B, Park S-Y (2007) Ontology population from unstructured and semi-structured texts. In: *Sixth international conference on advanced language processing and Web Information Technology (ALPIT 2007)*, pp. 135–139.
- Nakhaei Z, Ahmadi A, Sharifi A, Badie K (2021) Conflict resolution using relation classification: high-level data fusion in data integration. *Comp Sci Inform Syst*, pp 14–14.
- Brewster C, Ciravegna F, Wilks Y (2002) User-centred ontology learning for knowledge management. In: *International conference on application of natural language to information systems*, pp. 203–207.
- Celjuska D, Vargas-Vera M (2004) Ontosophie: a semi-automatic system for ontology population from text. In: *International conference on natural language processing (ICON)*.
- Craven M, DiPasquo D, Freitag D, McCallum A, Mitchell T, Nigam K, Slattery S (2000) Learning to construct knowledge bases from the World Wide Web. *Artif Intell* 118:69–113
- Mintz M, Bills S, Snow R, Jurafsky D (2009) Distant supervision for relation extraction without labeled data. In: *Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNLP, 2009*, pp. 1003–1011.
- Merhav Y, Mesquita F, Barbosa D, Yee WG, Frieder O (2012) Extracting information networks from the blogosphere. *ACM Trans Web (TWEB)* 6:1–33
- Vahideh Reshadat HF (2019) A new open information extraction system using sentence difficulty estimation. *Comput Inform* 38:986–1008.
- Lin H, Wang Y, Zhang P, Wang W, Yue Y, Lin Z (2016) A rule based open information extraction method using cascaded finite-state transducer. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 325–337.
- Saha S, Pal H (2017) Bootstrapping for numerical open ie. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 317–323.
- Etzioni O, Banko M, Soderland S, Weld DS (2008) Open information extraction from the web. *Commun ACM* 51:68–74
- Bollegala DT, Matsuo Y, Ishizuka M (2010) Relational duality: Unsupervised extraction of semantic relations between entities on the web. In: *Proceedings of the 19th international conference on World wide web*, pp 151–160.
- Gao T, Han X, Zhu H, Liu Z, Li P, Sun M, Zhou J (2019) FewRel 2.0: Towards more challenging few-shot relation classification. In: *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing*, pp. 6250–6255.
- Soares LB, FitzGerald N, Ling J, Kwiatkowski T (2019) Matching the blanks: distributional similarity for relation learning. In: *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 2895–2905.
- Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language

- understanding. In: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, vol. 1, pp. 4171–4186.
35. Nourani E, Reshadat V (2020) Association extraction from biomedical literature based on representation and transfer learning. *J Theor Biol* 488:110112
 36. Alt C, Hübner M, Hennig L (2019) Improving relation extraction by pre-trained language representations. In: Proceedings of AKBC 2019.
 37. Castano S, Peraldi ISE, Ferrara A, Karkaletsis V, Kaya A, Möller R, Montanelli S, Petasis G, Wessel M (2009) Multimedia interpretation for dynamic ontology evolution. *J Log Comput* 19:859–897
 38. Reshadat V, Akçay A, Zervanou K, Zhang Y, De Jong E (2021) Relation representation learning for special cargo ontology. In: IEEE Symposium Series on Computational Intelligence (SSCI), pp 1–8
 39. Kolkman T (2020) An ontology for special cargo. Master's thesis, Eindhoven University of Technology, The Netherlands, Retrieved from https://pure.tue.nl/ws/portalfiles/portal/148941906/Master_Thesis_Tess_Kolkman.pdf.
 40. Conneau A, Kiela D, Schwenk H, Barrault L, Bordes A (2017) Supervised learning of universal sentence representations from natural language inference data. In: Proceedings of the 2017 conference on empirical methods in natural language processing, vol. Association for Computational Linguistics, pp. 670–680.
 41. Hashimoto K, Xiong C, Tsuruoka Y, Socher R (2016) A joint many-task model: Growing a neural network for multiple nlp tasks. arXiv preprint [arXiv:1611.01587](https://arxiv.org/abs/1611.01587).
 42. Sanh V, Wolf T, Ruder S (2019) A hierarchical multi-task approach for learning embeddings from semantic tasks. In: Proceedings of the AAAI conference on artificial intelligence, pp. 6949–6956.
 43. Pennington J, Socher R, Manning C (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543.
 44. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer, L (2018) Deep contextualized word representations. In: Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, pp. 2227–2237.
 45. Chiu JP, Nichols E (2016) Named entity recognition with bidirectional LSTM-CNNs. *Trans Assoc Comput Linguistics* 4:357–370
 46. Bekoulis G, Deleu J, Demeester T, Develder C (2018) Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Syst Appl* 114:34–45
 47. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. *J Mach Learn Res* 3:993–1022
 48. Hasan KS, Ng V (2010) Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art. In: Proceedings of the 23rd international conference on computational linguistics: Posters, pp. 365–373.
 49. Qi P, Zhang Y, Zhang Y, Bolton J, Manning CD (2020) Stanza: a python natural language processing toolkit for many human languages. In: Proceedings of the 58th Annual meeting of the association for computational linguistics: system demonstrations, pp. 101–108.
 50. Boudin F (2016) PKE: an open source python-based keyphrase extraction toolkit. In: Proceedings of COLING 2016, the 26th international conference on computational linguistics: system demonstrations, 2016, pp 69–73.
 51. El-Beltagy SR, Rafea A (2009) KP-Miner: a keyphrase extraction system for English and Arabic documents. *Inf Syst* 34:132–144
 52. Florescu C, Caragea C (2017) Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In: Proceedings of the 55th annual meeting of the association for computational linguistics (Volume 1: Long Papers), pp. 1105–1115.
 53. Levenshtein VI (1966) Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady, pp. 707–710.
 54. Ester M, Kriegel H-P, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd, pp. 226–231.
 55. I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. O. Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz, “Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals,” *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 33–38, 2019.
 56. Han X, Zhu H, Yu P, Wang Z, Yao Y, Liu Z, Sun M (2018) Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. In: Proceedings of the 2018 conference on empirical methods in natural language processing, vol. 4803–4809.
 57. Reshadat V, Faili H (2019) A new open information extraction system using sentence difficulty estimation. *Comput Informatics* 38:986–1008
 58. Daniel J, James MH (2019) Speech and language processing: an introduction to natural language processing, Computational Linguistics, and Speech Recognition (3rd edn). Pearson, UK.
 59. Manning C, Schütze H (1999) Foundations of statistical natural language processing. MIT Press, Cambridge
 60. Daniel J, James MH (2007) Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition: Prentice Hall, New Jersey.
 61. Søgaard A, Goldberg Y (2016) Deep multi-task learning with low level tasks supervised at lower layers. In: Proceedings of the 54th annual meeting of the association for computational linguistics (Volume 2: Short Papers), pp. 231–235.
 62. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A (2020) Language models are few-shot learners. arXiv preprint [arXiv:2005.14165](https://arxiv.org/abs/2005.14165).
 63. Ling W, Luís T, Marujo L, Astudillo RF, Amir S, Dyer C, Black AW, Trancoso I (2015) Finding function in form: Compositional character models for open vocabulary word representation. arXiv preprint [arXiv:1508.02096](https://arxiv.org/abs/1508.02096).