Applying BiLSTM and CNN to assist actionable ontology building from FMEA documents

Kamath, Rashmi R.

Award date:
2019

Awarding institution:
Aalto University

Aalto University
School of Science
Master's Programme in ICT Innovation

Rashmi Kamath

Applying BiLSTM and CNN to assist actionable ontology building from FMEA documents

Master's Thesis
Espoo, September 20, 2019

Supervisor: Professor Aristidis gionis
Advisor: Carl Neale

Aalto University
School of Science

Aalto University
School of Science
Master's Programme in ICT Innovation

ABSTRACT OF
MASTER'S THESIS

Author: Rashmi Kamath

Title:
Applying BiLSTM and CNN to assist actionable ontology building from FMEA documents

Date:	September 20, 2019	**Pages:**	55
Major:	Data Science	**Code:**	SCI3095
Supervisor:	Professor Aristidis gionis		
Advisor:	Carl Neale		

Heterogeneous data sources can lead to high efforts for data integration, data access and data analytics. A possible solution to this problem can be the creation of a Semantic Information model also popularly known as a knowledge graph acting as an intermediate data source on which all data access applications are built. Knowledge graph allows huge organization to tap into the connection from various data sources and efficient data storage and data access techniques. A Failure mode and effect analysis(FMEA) document are used as a starting point for creating semantic information models. FMEA is a critical document for providing predictive failure analysis for product and processes in an automotive supplier company. It consists of simple statements which can be used for information extraction tasks such as named entity recognition and relation extraction.
The goal of this thesis is to extract information in the form of entity-relation-entity triples that serves as the basis for knowledge graph creation. As the data is highly company and domain-specific, the entities are labelled manually. Later, Neural network models like BiLSTM and CNN are used for relation classification task from sentences consisting of the entities. Information extraction task typically utilizes an ontology to identify the type of entity and therefore establishing a relation between the entities. However, due to lack of ontology, we make use of an end to end model for entity recognition and relation extraction. Therefore, the results can also serve as the basis for a domain ontology creation for the automotive supplier industry.

Keywords: information extraction, entity recognition, semantic relation extraction, knowledge discovery, recurrent neural network, convolutional neural network, Industry 4.0, Semantic Information model, natural language processing

Language: English

Acknowledgements

This master thesis is a part of a double degree program in Data Science with ICT Innovation at the Eindhoven University of Technology in Netherlands and Aalto University in Finland. I am thankful to all my professors at both universities who helped me gain a solid foundation and understanding of theoretical and practical knowledge in my discipline. Having gained the necessary knowledge while attending the courses, provided me with great confidence to work on this thesis.
I would like to thank my thesis supervisor Aristides Gionis for providing the necessary support and guidance throughout the thesis work.
I would also like to thank my thesis advisor Carl Neale for providing me with the required domain understanding, enthusiasm towards the topic and also for making sure I stick to a routine and schedule which helped a lot to remain focused during implementation and report writing.
I would like to thank Vishwavardhan and Ajay for providing constant support and motivation, and poof reading my thesis with an academic eye.
Lastly, thanks to my parents for inspiring me and instilling the belief that learning does not end with formal education but a life long process.

Espoo, September 20, 2019

Rashmi Kamath

Abbreviations and Acronyms

BILSTM	Bi-directional Long Short-Term Memory
RNN	Recurrent Neural Network
CNN	Convolutional neural Network
POS	Part Of Speech
NER	Named Entity Recognition
NLP	Natural Language Processing
FMEA	Failure Mode and Effect Analysis

Contents

Chapter 1

Introduction

1.1 Motivation

Data and more specifically the value of data is becoming more and more important in industry as organizations search for more deficiencies in their operations. Data analytics and machine learning are coming more in focus and companies are investing in these methods as there has been a significant advancement in these techniques. Multinational companies create a large number of data and documents defining and detailing all aspects of their organization. These can be business processes, quality requirements, manufacturing instruction, cost calculations, etc. These companies also create vast quantities of transnational data usually holding valuable information. The problem for these organizations is that the data landscape is very heterogeneous and most of the time spent during analytics is in preprocessing. A potential solution to some of these issues is semantic data Integration. By defining the specific domain world in semantic models allows the linking and integration of heterogeneous data sources and provides efficient data access for multiple business applications.

1.2 Problem statement in Automotive manufacturing industry

An automotive manufacturing industry typically consists of different kinds of data obtained from various sources such as the bill of material from suppliers, customer sales data, data produced from machines in an automated production line, product returns forms, parts lists and training plans. The data obtained from all these sources are often stored in different types of

databases such as enterprise resource software, manufacturing execution system, text documents in XML format, product return in excel sheets, etc. The data holds valuable information used for various decision-making processes regarding machine maintenance, product return costs, energy consumption, information on suppliers, customers and so on. Currently, different data access tools and application utilizes data directly from these heterogeneous sources which lead to inefficiencies and high prepossessing costs. It is experienced that semantic data access often becomes a huge challenge with heterogeneous data sources. The solution to this problem is to create a semantic information model or knowledge graph using all the data sources. Later, individual applications can be built on the semantic information model instead of assimilating data from heterogeneous data sources every-time to build a business application used to access data. However, the scope of creating the semantic information model is large and extremely complex. Hence, we look for existing documents within the business landscape that already have some structure which could be used as a starting point for defining a domain-specific semantic information model. The Failure Mode and Effect Analysis(FMEA) is a machine-readable text document that follows a clear structure and contains simple statements that define the products and processes. Therefore, the scope of this thesis is to perform information extraction which includes entity recognition and relation extraction from the machine-readable FMEA text document. The information extracted from these text documents, in turn, serves as the basis for creating a domain-specific knowledge base. It also serves as a basis for domain ontology building.

1.3 Structure of the Thesis

This thesis starts by describing the relevant literature about the automotive supplier industry in Chapter 2. It also describes in brief the fundamentals behind the techniques used for implementation. Chapter 3 describes techniques used for data exploration and problem formulation in the context of machine learning. It discusses the possible ways to approach the problem and reasons behind choosing the techniques. Chapter 4 explains the dataset and implementation. It describes the two experimental setups designed for solving the problem statement. Chapter 5 shows the results of the implementation. It defines the performance metrics used for determining the accuracy of the implemented model. It also provides a comparison within the experimental setup for different hyperparameters. Chapter 6 discusses the results, comparison with similar work and limitations of the thesis. Chapter 7 provides a conclusion for the thesis.

Chapter 2

Literature Review

2.1 Background of Automotive Supplier Industry

Automotive supplier companies are involved in the design, development, manufacturing, marketing, and selling of automobile parts which is used in the final assembly of an automobile. They supply automotive parts either directly or indirectly to an Original Equipment Manufacturer (OEM). For this thesis, data has been provided by an automotive supplier company headquartered in Germany. The name of the company cannot be used for privacy reasons and is therefore referred to as *ABCD Group* throughout the scope of this thesis. ABCD group is one such tier one company in the automotive supply chain. It produces different automotive components such as comfort actuators, power actuators, thermal systems, and wiper systems. These actuators and systems are used for various functions inside an automobile such as window lifting, seat adjustment, braking system, climate control and wiper systems.

ABCD group is in the midst of a major transformation to Industry 4.0, a subset of the fourth industrial revolution [19]. Industry 4.0 involves customization of products that cater to every customer of the ABCD group. It involves the use of cyber-physical systems to link real objects and people with information processing via information networks. It intelligently links the entire production process with the use of state of the art technologies such as the Internet of things, 3D printing, artificial intelligence, bio-engineering, and cloud computing. In some cases also makes use of nanotechnologies and new efficient and intelligent materials [4].

One of the design principles and goals of Industry 4.0 is information transparency and data integration. As of now, the company utilizes data from

different machines and enterprise resource software to create different applications and dashboards to enhance appropriate decision making. Figure 2.1 shows the current scenario in the company where data is obtained from different sources ranging from machines used in production processes to enterprise resource software used for the Bill of materials. The figure explains how there are approximately 1000 different sources of information. It also shows the need for approximately 100 applications that utilize different kinds of data. For example, the maintenance support system may utilize data from SAP systems, machines, etc. to create maintenance-related dashboards and report to enhance the decision-making process. However, energy platforms might use data from other resources. Therefore, the effort for data integration is the number of applications multiplied by several data sources i.e. shown as m x n which is at present tremendous.

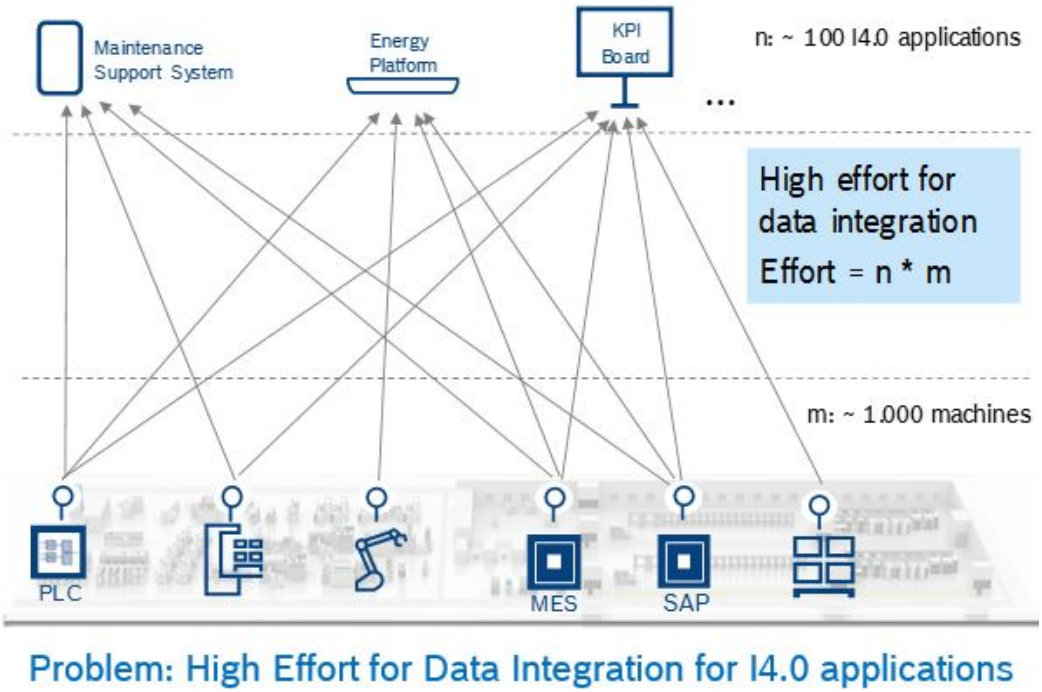


Figure 2.1: Current scenario where different application uses data from different data sources

One of the problems adding to data integration is also the quality of data. For example, there is ambiguity in the nomenclature of data, the part referred to as *ring* in an SAP data source might be referred with another name in another data source such as FMEA due to human error as it is mostly manually written by domain experts. These types of problems add to

the challenges in data integration. Also, the ABCD group is a multinational organization having production plants across different countries. Therefore, it poses additional challenges in product and process standardization. For example, the production plant in a European country tends to have a higher degree of automation which includes the utilization of robotic arms during production. Whereas, one of the plants in an Asian country still performs the same process manually. Thus, forcing the company to create a different application for report generation. The organization proposes a novel approach to resolve the challenges of moving to Industry 4.0 and overcoming data integration issues. They plan to integrate the data from different data sources into an industry and domain-specific knowledge graph.

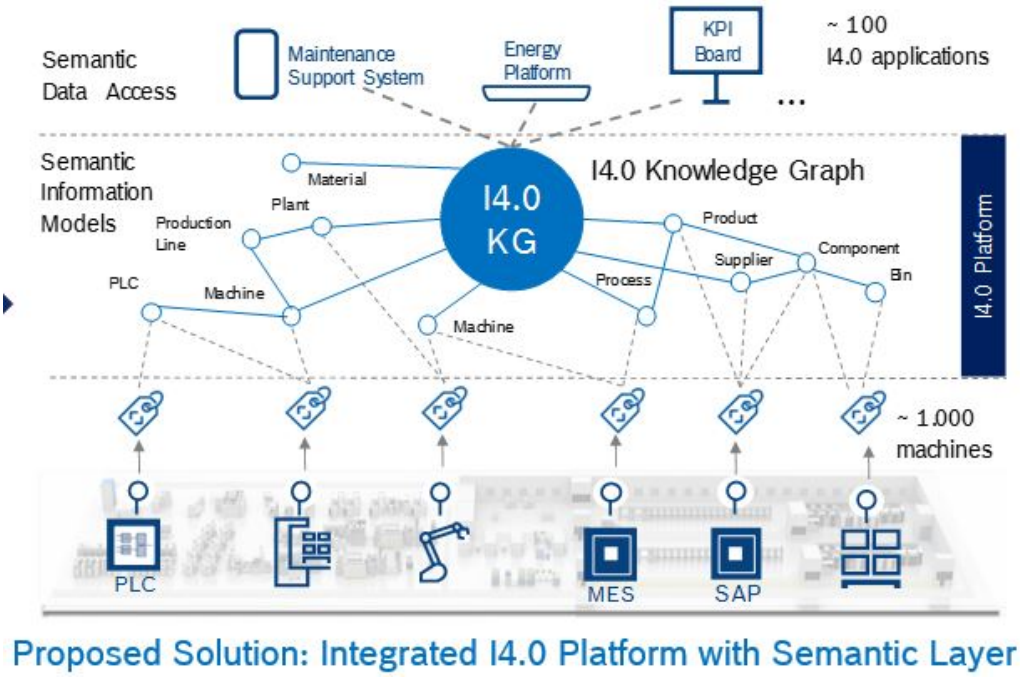


Figure 2.2: Proposed I4.0 knowledge graph platform to integrate data across sources which acts as a common platform for all applications

This project explores the possibility of utilizing one such data source known as FMEA to contribute towards I4.0 knowledge graph generation.

2.1.1 Introduction to Failure Mode and Effect Analysis

Failure Mode and Effect Analysis, or FMEA, provides an analytical method for quality management during the development of product and process. An

unsatisfactory product and/or process design can lead to product recalls inturn causing high costs to the company. Therefore FMEA methodology aims to allow organizations to anticipate failure during the design stage. FMEA provides a structured approach to discover potentials failures and evaluate risks before the production process. It assists in bringing flawless products to the market by identifying, prioritizing and limiting failure modes beforehand. However, it cannot be treated as a substitute for good engineering, rather it is used in enhancing good engineering.

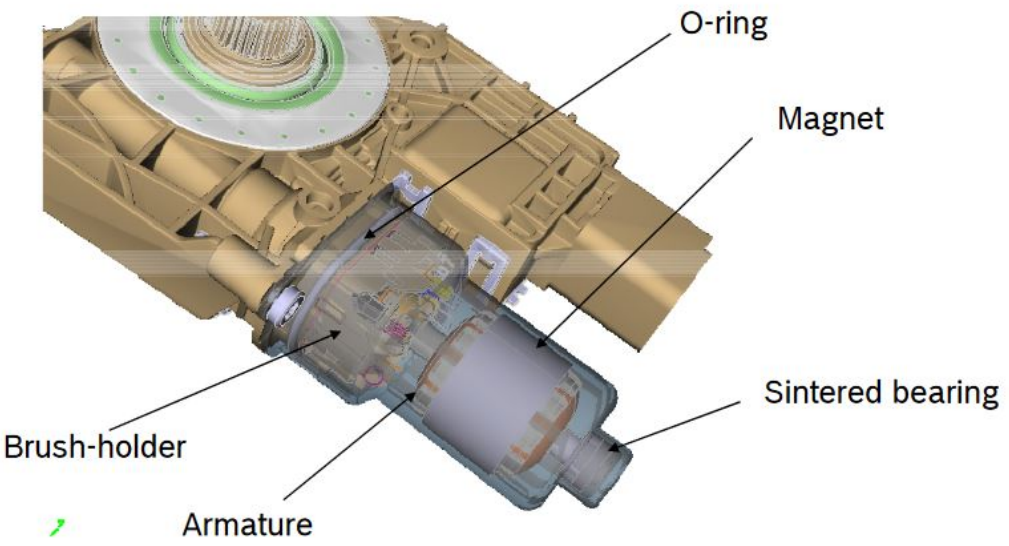


Figure 2.3: 3D model of comfort actuator as an example

There are two types of FMEA, namely, Product FMEA and Process FMEA. Product FMEA explores the functions, possible failures of each subpart that constitute the entire product. It also explores the possible failures of individual component's interaction with each other including but not limited to material properties, tolerance levels and degradation. Process FMEA explores to discover failure that impacts the design of processes in terms of quality, reliability, etc. It also takes into account various factors such as materials, machines, human error and other potential factors that can impact the process. In the Product FMEA, the scope of analysis can be presented in the form of a block diagram or a 3D model. Here, the boundaries of analysis and the interfaces are identified and established as shown in figure 2.3. Process FMEA typically uses a flow chart as shown in figure 2.3.

The FMEA further explains the progression of failure mode and lists the probable causes for each failure. Figure 2.5 shows excerpts of bicycle FMEA

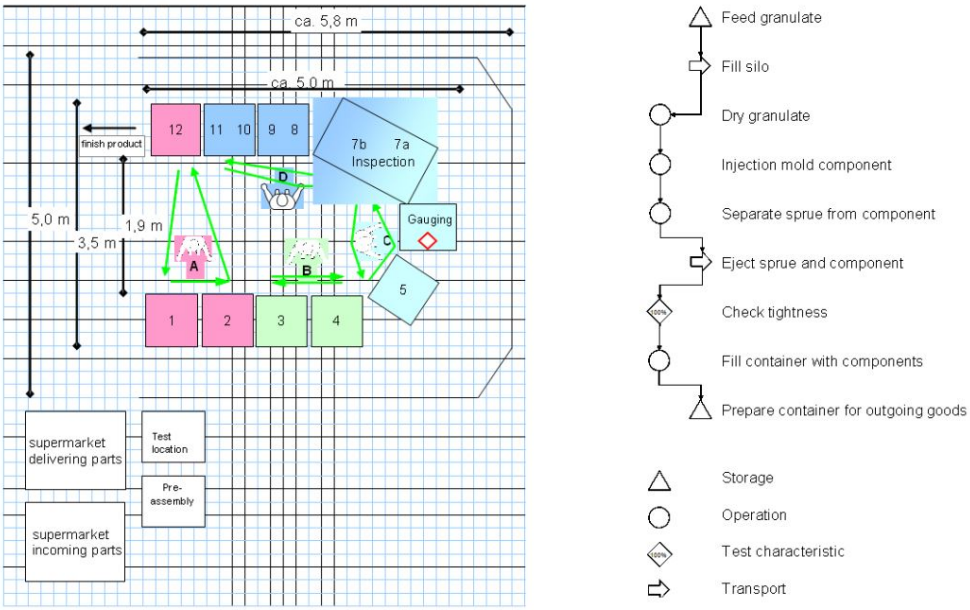
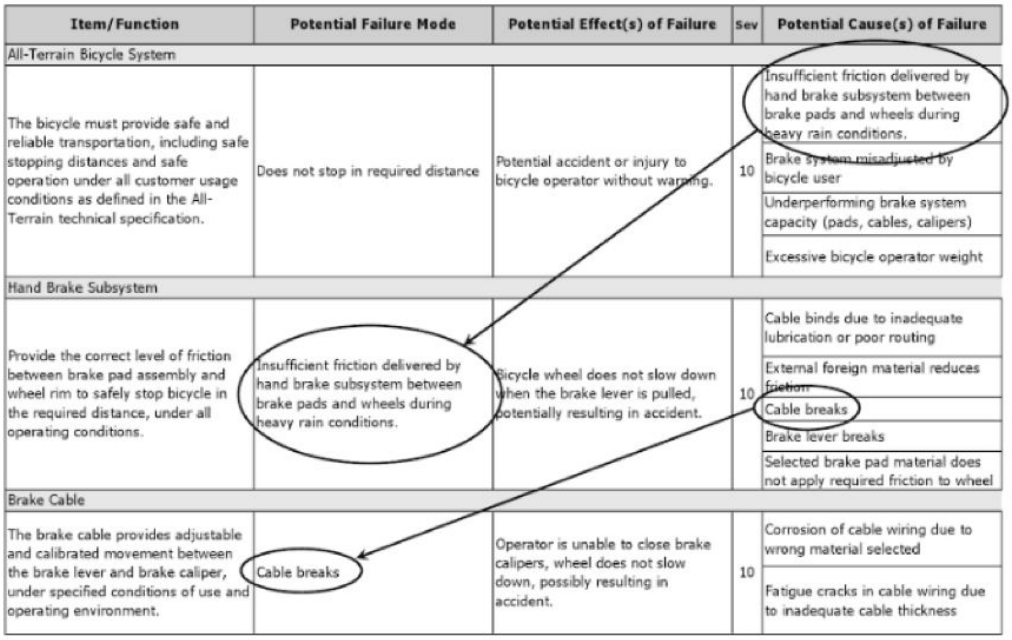


Figure 2.4: Process layout and flow chart as an example

which describes each Item and potential failures and causes.

2.2 Information extraction from Text

Information extraction attempts to automatically extract structured information from unstructured or semi-structured digital documents [24]. Often a lot of experiential knowledge occurs in repositories such as reports, blogs, papers and other such documents that are created and maintained within or across an enterprise. It, therefore, becomes essential to extract, disseminate and reuse this domain knowledge during decision-making and increase the operational efficiency of tasks within an organization. Recently, attempts have been made to extract information in various domains including bio-medical, broadcasting companies, and multinational technology. The task of information extraction can be classified into two main components namely, Named entity recognition and relation classification. For,an example in a sentence *"Hashim Premji is the founder member of Wipro Limited"*, the task of information extraction would be to identify "*Hashim Premji*" as the name of a person and "*Wipro Limited*" as an Organization. The relation between the two entities *"FounderOf"*, also denoted most commonly as *FounderOf(Hashim Premji, Wipro Limited)* is identified. In the above



Item/Function	Potential Failure Mode	Potential Effect(s) of Failure	Sev	Potential Cause(s) of Failure
All-Terrain Bicycle System				
The bicycle must provide safe and reliable transportation, including safe stopping distances and safe operation under all customer usage conditions as defined in the All-Terrain technical specification.	Does not stop in required distance	Potential accident or injury to bicycle operator without warning.	10	Insufficient friction delivered by hand brake subsystem between brake pads and wheels during heavy rain conditions.
				Brake system misadjusted by bicycle user
				Underperforming brake system capacity (pads, cables, calipers)
				Excessive bicycle operator weight
Hand Brake Subsystem				
Provide the correct level of friction between brake pad assembly and wheel rim to safely stop bicycle in the required distance, under all operating conditions.	Insufficient friction delivered by hand brake subsystem between brake pads and wheels during heavy rain conditions.	Bicycle wheel does not slow down when the brake lever is pulled, potentially resulting in accident.	10	Cable binds due to inadequate lubrication or poor routing
				External foreign material reduces friction
				Cable breaks
				Brake lever breaks
				Selected brake pad material does not apply required friction to wheel
Brake Cable				
The brake cable provides adjustable and calibrated movement between the brake lever and brake caliper, under specified conditions of use and operating environment.	Cable breaks	Operator is unable to close brake calipers, wheel does not slow down, possibly resulting in accident.	10	Corrosion of cable wiring due to wrong material selected
				Fatigue cracks in cable wiring due to inadequate cable thickness

Figure 2.5: Excerpts of Bike FMEA

example, the entities provided are named entities which include Person, Organization, Location, Date, Time, etc. However, there is a possibility to identify domain-specific entities for example enzymes, genes, cells for a biological domain.

2.2.1 Named Entity recognition

Named entity recognition is the task identifying most relevant nouns such as people, place or organization from a given text. The definition of an entity may change based on the domain of the textual data. For example, in a pathology report which describes an anomaly, is a certain part of the human body, it may be important to identify words such as heart, lungs, left-lobe. An entity is defined as a thing with distinct and independent existence [6]. An entity recognition algorithm can have various useful applications. It can be used in automatically organizing content, for example, a broadcasting company that produces lots of digital content ranging from sports celebrities to weather forecasting. Therefore, named entity recognition models can be applied to identify entities associated with sports or the type of weather to automatically classify the content into various classes. Another application is to design an efficient search algorithm, therefore the algorithm only searches

based on relevant entities to produce efficient results and also reduce the time of the search. Recommendation systems are powered by entity recognition algorithms, it has applications across entertainment websites such as Netflix, digital news applications such as BBC, etc. Users can be recommended similar content based on their interests recognized by past search patterns. It can have applications in customer support, where products can be automatically identified based on entity recognition algorithms. Thus, eliminating the need to manually go through customer complaints and in turn, speeding customer complaint handling process [20]. For an automotive part manufacturing company it can be used to identify the various sub-parts as entities such as magnet, bearing rings, etc. which may be used in different products. Thus being able to identify and distinguish among different products or processes within an organization. Later, the entities can be used for various applications as mentioned above such as automatic classification of various reports, recommendation system, efficient search algorithms and, or creating a domain-specific knowledge base.

2.2.2 Relation extraction

Relation extraction is another important task within information extraction, it also plays a role in structuring text. It attempts to detect and classify a semantic relation between two recognized entities in a text document. It can have application across various domains, one such domain is biology. For example, in the sentence, *TNF is one of the gene related to breast cancer.*
TNF(Gene) and breast cancer(Disease) are identified as entities in biology domain which possess a gene-disease relation. Automatic extraction of such relations from textual documents can have tremendous benefits in biology domain [18].
One of the ways to approach relation extraction is the use of domain ontologies. Domain ontologies help to detect the meaningful relationship between different types of entities. Let us consider the example of an entity identified as Person, ontologies help identify the most important aspects of relations between entity Person, such as birth-date, birthplace, children. Therefore, in the following sentence:
Elon Musk is a technology entrepreneur born in South Africa.
A semantic relation *birthplace(Elon Musk,South Africa)* can be identified between the two identified entities namely Person and Location.

2.3 Ontology

Ontology is the philosophical study of being, existence or reality in general which deals with the questions concerning what entities exist or can be said to exist. Also, how such entities can be grouped, related within a hierarchy, and subdivided according to similarities and differences [25]. Every domain can create ontologies to limit complexity and organize information into data and knowledge. New ontologies are made to improve and or automate problem-solving within that domain [26]. Ontologies provide the structural schema to hold information. Since Google started an initiative called knowledge graph, a substantial amount of research has used the phrase knowledge graph as a generalized term. Therefore, ontologies provide the necessary structure to hold the data in the knowledge base.

2.3.1 Domain Ontology

A domain ontology represents concepts which specifically belong to certain parts of the world. It provides domain-specific models and the definitions of terms, also hold value only within the domain. An ontology about the domain of *Electrical appliances* would refer to the word "Washer" in terms of washing clothes, mostly by a washing machine. However, the same word possesses a different meaning in *Automotive Industry* domain, where is refers to a ring-shaped "metal" used to distribute the load of a screw evenly. Domain ontology creation is largely manual, time-consuming and expensive process which requires domain expertise. It is called ontology engineering relates to the development of ontologies for a specific domain and is a sub-field of knowledge engineering [26].

2.4 Supervised learning

Supervised learning is a powerful method used for classification and regression tasks. Regression is a technique used for predicting and forecasting purposes, is uses historical data such as various factors for stock prices in the stock market to determine the future performance of stocks. Classification task, analyse the data to recognize patterns to classify data into one of the predefined categories. There are many classification tasks such as sentiment classification, detecting fraudulent credit card transaction and detecting spam emails. Supervised learning considers a labelled dataset with input variable(x), known output variables (Y) for each variable (x) and uses

an algorithm to learn an input to output mapping function given as

$$Y = f(x)$$

The goal is to gather a large number of training samples and approximate the function such that for every new variable (x), a (Y) can be predicted based on the given function.

2.5 Reinforcement learning

Reinforcement learning differs from supervised learning in a way that it learns decision making sequentially. Instead of considering input and output variable to learn a constant function, it learns based on trial and error reward-based environment over a period of time. The object recognition algorithm is a good example of supervised learning algorithms, however, a game of chess is an example of reinforcement learning where an agent learns in each event a particular behaviour to maximize the performance. There are two types of reinforcement, positive and negative reinforcement. It provides a reward-based system where the agent is either penalized for the wrong action or rewarded for correct action with an aim to maximize the total reward. It is very helpful in natural language processing, as natural language is complex since the same words can change its meaning in different contexts. Therefore, a trial and error method provides the basis for an agent to learn the meaning of data in a different context.

2.6 Artificial Neural Network

Deep learning, a sub-field of machine learning techniques takes inspiration for its architecture from the actual working of neurons in a human brain. Deep learning concept in recent time has achieved great success in computer vision and natural language processing tasks. Deep learning has various application from driver-less cars to customised voice recognition and it achieves results considered impossible in the past. The reason is now there is the availability of huge training data and computing power to run complex algorithms. The results obtained from deep learning can be significantly improved with a large amount of training data which is not possible with older machine learning algorithms. There are different techniques of deep learning such as a convolutional neural network(CNN), Recurrent Neural Network (RNN), fully connected neural network etc. CNN and RNN are most popular among the techniques.

2.6.1 Convolution Neural Network

A convolutional neural network (CNN) is a neural network consisting of convolution. A convolution is a mathematical operation or algorithm which can be applied to input data such as image or sentence, assign importance to specific aspects of input to be able to distinguish them from each other. CNN's are popularly used for the application of computer vision. The figure 2.6 shows an edge detection kernel applied on an input image to obtain a feature map of edges in the input image. The image on the left consists of an RGB (red, green, and blue) image. Each pixel in the image consists of red, green and blue (typically in the range of 0 to 255 for each colour) a combination of which provides the final colour to the pixel. A feature detector or edge kernel (for example a 3 x 3 matrix) is slide over the entire image and multiplied element-wise with original image pixel values to obtain the feature map as shown in the figure. Similarly, other kernels can be applied to obtain different features in an input image to obtain appropriate feature maps.

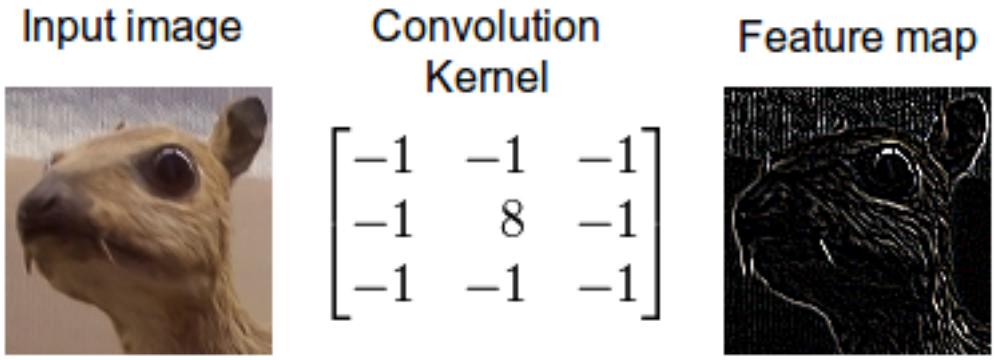


Figure 2.6: Convolving an image with an edge detector kernel

The understanding of a convolutional neural network is more intuitive in computer vision tasks than in natural language processing. However, in recent times, convolutions have also been applied to certain Natural Language Processing (NLP) task to obtain state of the art results. Convolutional neural networks work with numerical values, however, NLP consists of text documents, sentences and words. Therefore, the words that consisted of the documents are initially converted to word embedding. Word-embeddings are numerical matrix representation of a word, with typical dimensions ranging from 50 to 300. For a sentence which consists of 5 words, the vector representation will consist of a 5x100 matrix (when considering the size of the embedding dimension as 100) as the output.

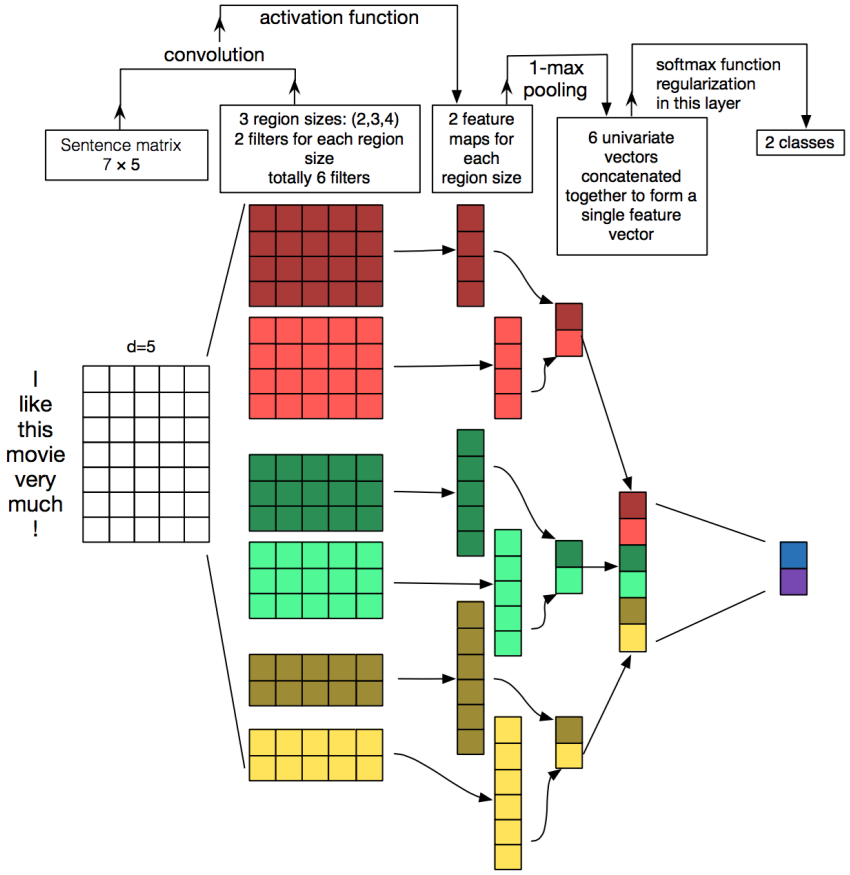


Figure 2.7: Convolutional Neural Network for sentence classification [29]

Then the convolution filter applied only vertically as shown in figure 2.7 with filter sizes typically of 2,3,4 and 5. The convolution is applied vertically so as to not split the word. The filter sizes are similar to considering bigrams, trigrams etc. Later pooling layers are used to obtain a sub-sample of the convolution layer thus providing a fixed-size output matrix helpful for a classification task. Pooling is also a dimensionality reduction technique which keeps the most salient features from convolution.

2.6.2 Recurrent Neural Network

As a human, while reading an article in newspaper it is easy to understand the concept due to memory of word meaning and context understanding. However, when it comes to machine it is not the same. Therefore, RNN

was designed to address this shortcoming, they are a network with loops which allows persistence of information. Figure 2.8 shows the unfolding of an RNN architecture. If we consider a sequence of words in a sentence then the formulas in the RNN used for computation can be given as follows [13]:

- x_t is the word input at time t. It should be noted that the word input in this case is in the form of a vector.
- s_t is hidden state of RNN at time t. s_t takes into consideration the previous state and current input and is computed as $s_t = f(Ux_t + Ws_t - 1)$. A nonlinear function $tanh$ or $ReLU$ is used to compute f.
- The first hidden state, in this case s_{t-1} is initially to zero.
- o_t is the output state at time t. A softmax function consisting of vector probabilities from the vocabulary can be given by $o_t = \text{softmax}(Vs_t)$ in order to predict the output o_t.

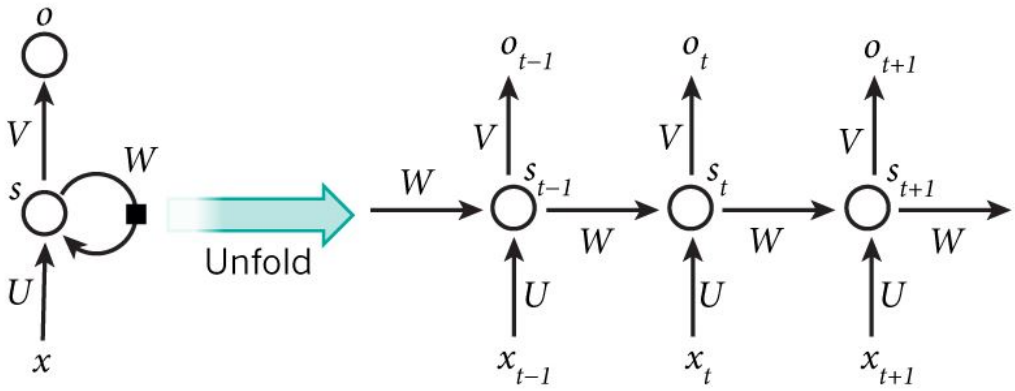


Figure 2.8: RNN model [13]

2.6.2.1 Bidirectional Long Short Term Memory

Long short term memory(LSTM) is the most commonly used RNN. They are better at capturing long term dependencies due to their different way of computing the hidden state. We see in the previous example from RNN that the hidden state consists of simple structure such as $tanh$ or $ReLU$, however, in LSTM hidden layer differs from RNN. As the name suggest BiLSTM is bidirectional LSTM network. Figure 2.9 displays a Bidirectional LSTM architecture, which contain LSTM sequence both forward and backward directions. Also, the hidden layer is contained during both the direction in the

sequence. Figure 2.10 shows the hidden layer of the network consisting of forget gate, input gate and output gate.

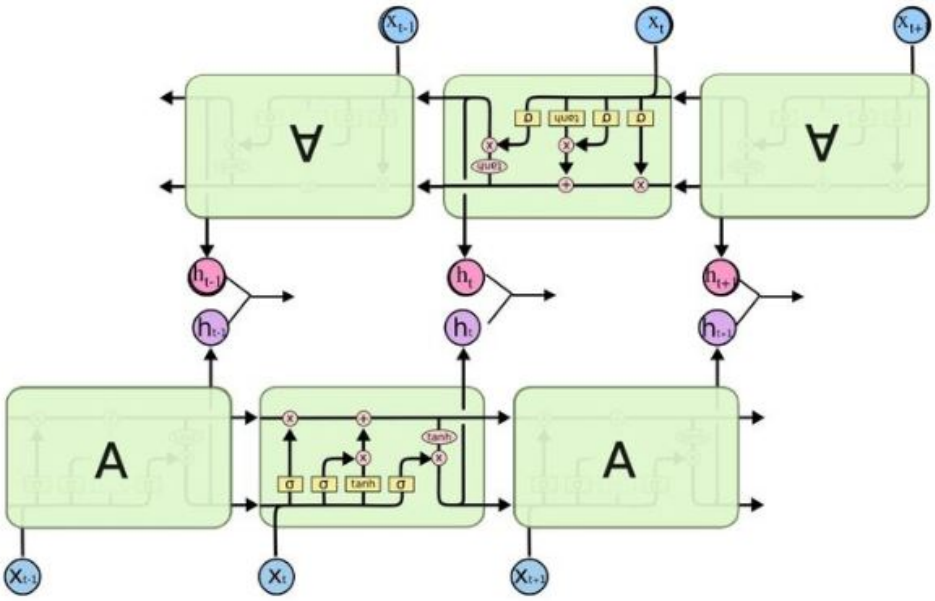


Figure 2.9: BiLSTM Model [21]

A cell state as shown in figure 2.10 allows the information to pass from C_{t-1} to C_t without changes, provided there is no input from other junction of the cell state. The structured gates allow information to be added or removed from the cell state [21]. The forget gate is the first layer of the LSTM, it is a sigmoid function which decides the information retention from previous state. For example, if a previous sentence consist of a proper noun with gender associated with it, then it is valuable to retain the information in order to use the appropriate pronoun in next sentence.

$$f_t = \sigma(x_t U^f + h_{t-1} W^f)$$

The second step is the input gate layer decides the new information that is stored in the cell state. For example, in case of change in proper noun, the new gender information needs to be stored by deleting the old gender information.

$$i_t = \sigma(x_t U^i + h_{t-1} W^i)$$
$$\tilde{C}_t = tanh(x_t U^g + h_{t-1} W^g)$$

Lastly, the output state is a filtered version of the cell state.

$$o_t = \sigma(x_t U^o + h_{t-1} W^o)$$

$$C_t = \sigma(f_t * C_{t-1} + i_t * \tilde{C}_t)$$
$$h_t = tanh(C_t) * o_t$$

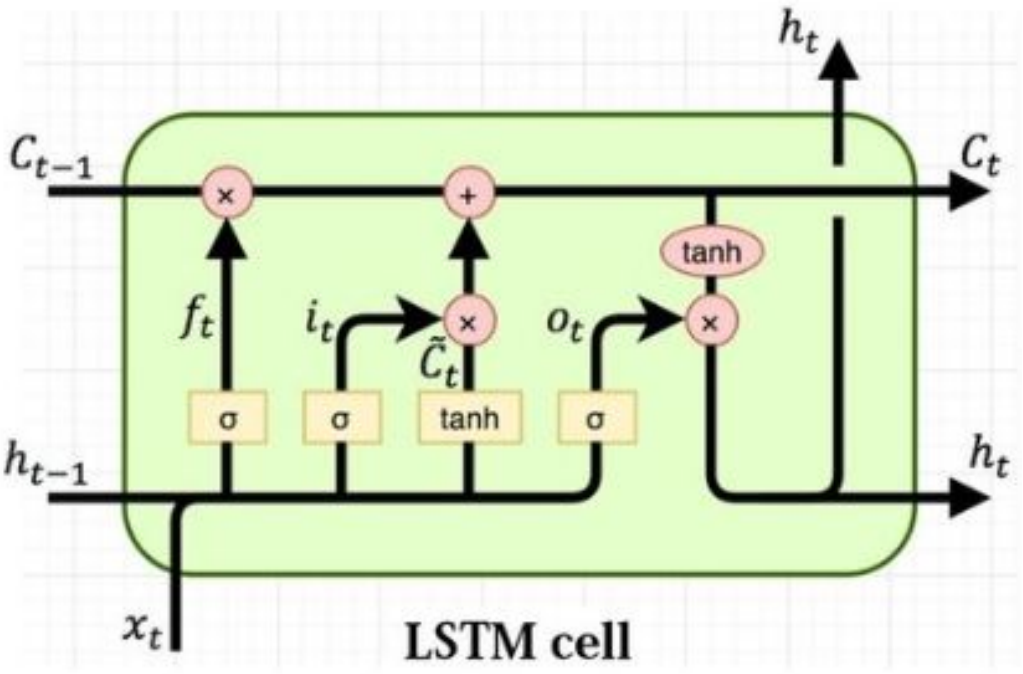


Figure 2.10: LSTM hidden state [21]

We see from figure 2.9 that the same hidden state is used in forward and backward direction, such that dependencies are considered from both sides i.e. before and after the given word. Therefore, BiLSTM is considered more useful in relation classification task as entity-relation can be identified irrespective of its position in the sentence.

2.6.3 Attention Mechanism

Recently, attention mechanisms have proved to be very useful in various natural language processing tasks such as image captioning, machine translation, semantic relation classification etc. Intuitively it can be explained in various contexts as follows:

- During translation task, from one language to another. The attention at each point is required only on one word or context of the entire sentence and not at the entire sentence at once.

- During classification task, a certain word might have greater importance than others to understand the context. For example, in the sentence *Xyz disease can cause blindness among human beings* the attention on the word *cause* can be very helpful for classifying a Cause-Effect relation between *Xyz* disease and blindness.

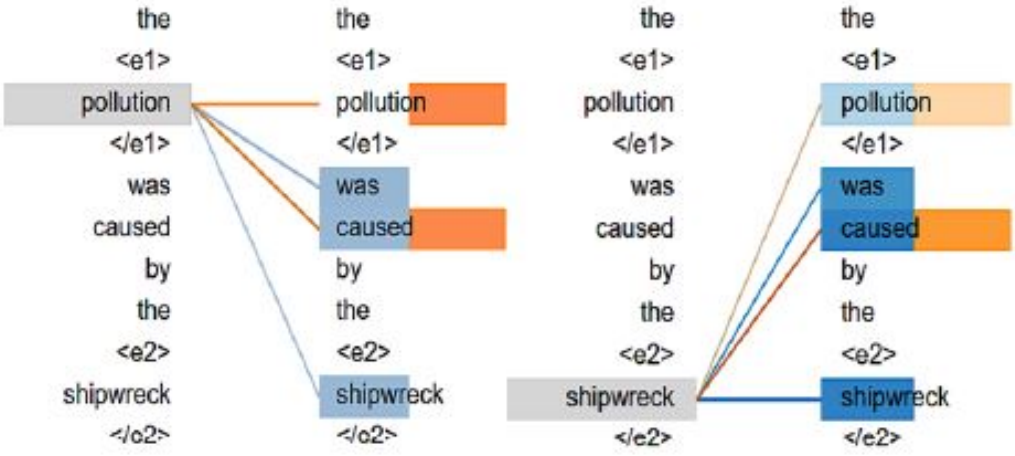


Figure 2.11: Example of self attention used for relation classification task[14]

"An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors"[22]. Certain weights are assigned to each value computed based on a compatibility function, where the output is a weighted sum of all the values. There are different types of attention such as scaled-dot-product attention, multi-head attention, self-attention etc. Figure 2.11 shows an example of self-attention used for relation classification task[14]. It shows entity 1 *pollution* and entity 2 *Shipwreck* focus on *was caused* and each other. The colour intensity determines the attention values for each word, thus assisting the classification.

Chapter 3

System Overview and Problem definition

This chapter focuses on the methods used to extract the textual data. It describes the software used to write the analysis report. The various formats in which the data can be extracted, the most appropriate format chosen for further analysis. It describes the tools used for data preprocessing. Lastly, it defines the problem statement in computer science terminology.

3.1 Data description

The sentences from failure mode and effects analysis(FMEA) are used for information extraction. ABCD group uses the IQ-RM software for writing and maintaining FMEA documents. An FMEA is written for each of the products and processes, produced and functioning within the company. Figure 3.1 shows an example of FMEA containing structure editor, failure editor, and list of nested structure, function and failure net. The FMEA uses a tree-like structure to describe products, sub-products, processes, and sub-processes. The root node displays the name of the final product named cruise control SC 2042, containing sub-components such as control unit, signal cable, and external systems. These sub-components further are formed from cables, plugs and other such sub-parts. When the user clicks on one of the sub-parts such as shown *Sensor 1*, the top right corner displays name, functions and potential failures. The software also has a lot of additional displays and editors such as failure net, function net, etc. Clicking on a failure or function also displays the failure or function net respectively as shown in the below part of the figure 3.1. The software allows exporting the document in many formats such as IQ export file, XML file, an HTML document, etc. For extraction

and data preprocessing, XML format proved to be the most useful format.

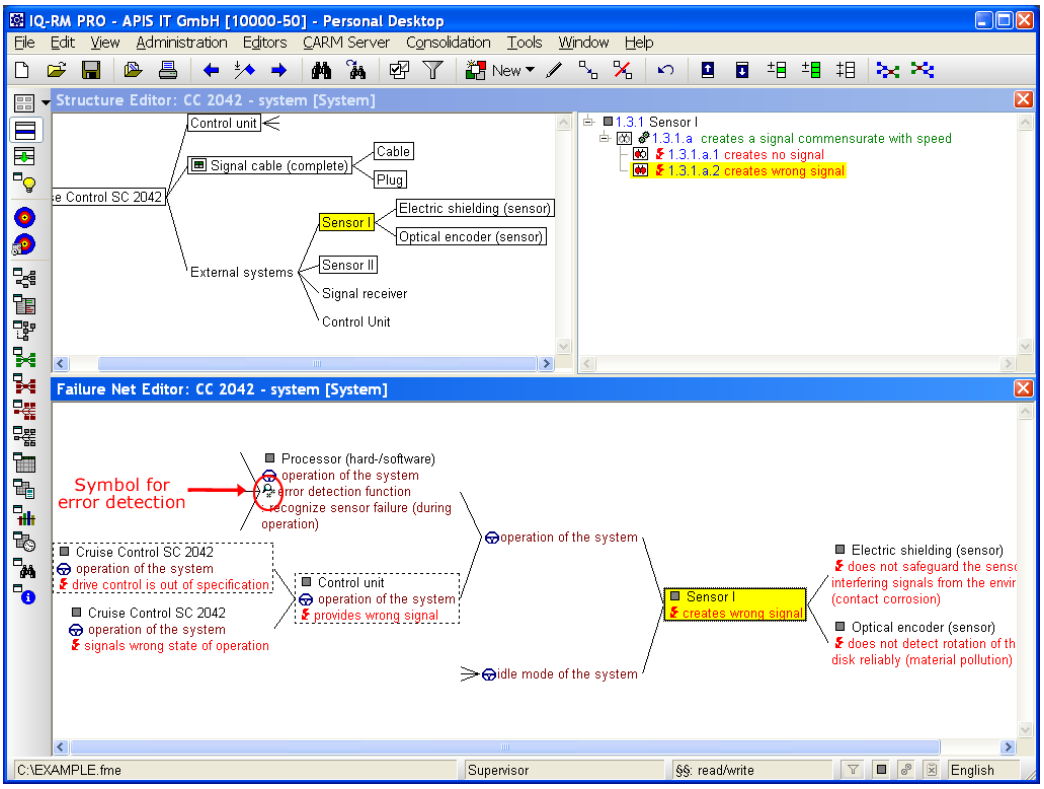


Figure 3.1: Basic structure of an FMEA [2]

3.2 Data exploration

The ABCD group has approximately hundreds of FMEA documents for each product and process across different countries. Also, a new FMEA is written for each variation in the product or process in the same or a different production plant. One of the well written FMEA for a window lift motor was explored for our project. The FMEA exported in the XML format was explored for structure, function and failure related sentences required for the information extraction task. Figure 3.2 shows a snippet of the XML file of a Window Lift motor. Initially, a list of useful elements was obtained by domain experts which was used for extracting only the necessary sentences from the XML file. The entire XML document had to be read manually to create Xpath queries to extract the useful element concerning the list. For example, the Xpath query to extract structure name from figure 3.2 is as

follows: */MSRFMEA/FM-STRUCTURE-ELEMENTS/FM-STRUCTURE-ELEMENT/LONG-NAME* which provides the element *FPG3 fourpole drive*. Thus, Xpath queries were used to extract sentences from the FMEA which was further used for the end-to-end entity and relation extraction model.

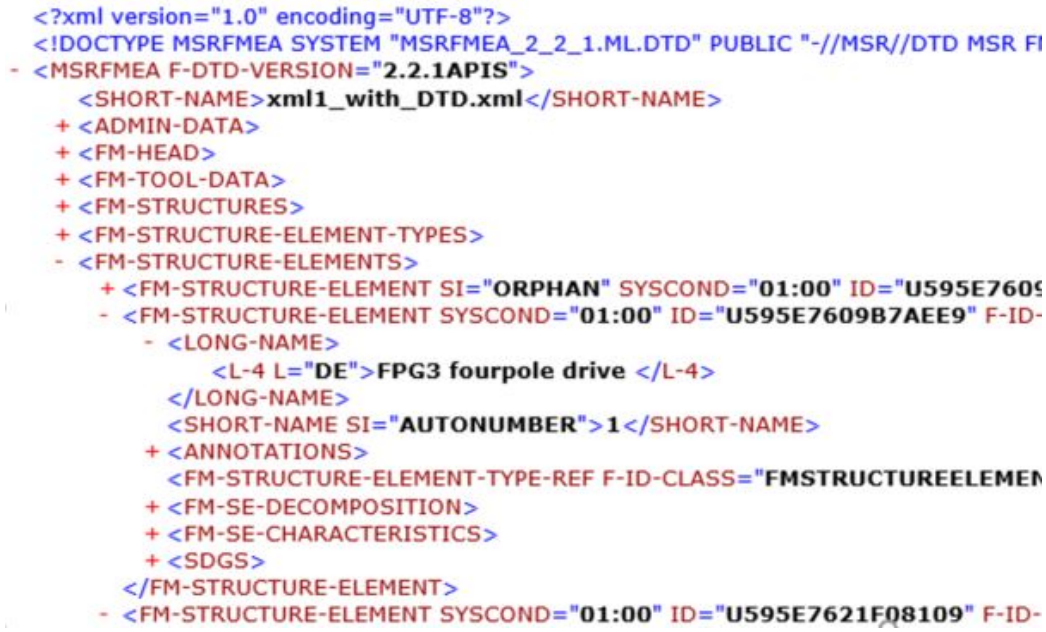


Figure 3.2: XML structure of the exported FMEA

3.3 Tools for Xpath queries

The company uses *KNIME* open source software for data analytics and reporting [12]. Therefore, knime software was used to extract all the sentences from the XML file using the Xpath queries. Figure 3.3 shows the knime workflow for the entire process. The software contains pre-defined nodes to perform a specific function. As shown in the figure 3.3, the XML reader node was used to read-in the FMEA file exported from the IQ-RM software. The path to file is specified in the configuration to read the XML. The second node is used for specifying the Xpath, to only read the elements such as characteristics, functions, faults and actions from the XML file. The column filter node filters out all the unnecessary columns to keep only the required ones, as each Xpath queries add a different column to the document. Lastly, the concatenate node joins all the necessary columns. The CSV Writer node is used to write all the useful sentences extracted during the workflow in a

tsv file.

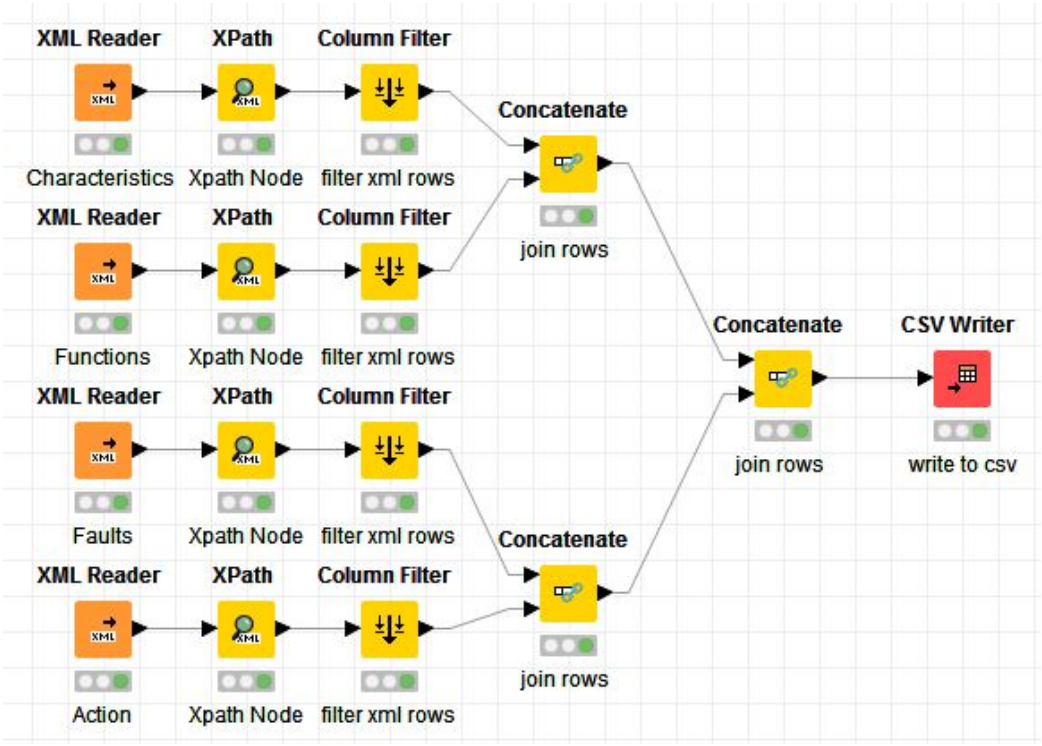
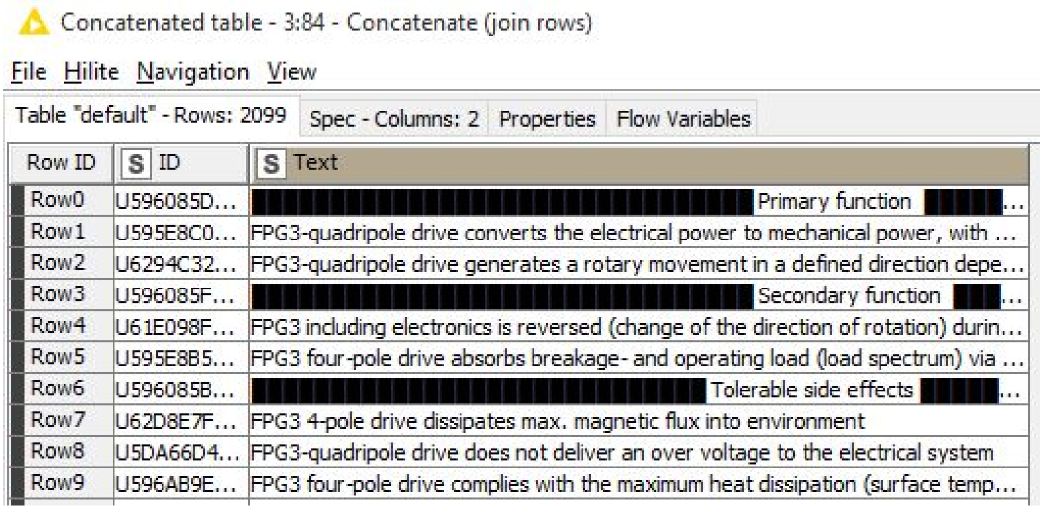


Figure 3.3: KNIME workflow to extract required elements from XML file

The figure 3.4 shows a snippet of the concatenated file containing the entity and relations to be extracted from each sentence. The FMEA contained a total of 2099 sentences. It is important to note that there are certain sentence extracted from the file, such as an entire line consisting of the words "Primary function" as shown in figure 3.4 is contained in the file and our Xpath queries extract all the elements contained for the particular Xpath. Therefore, there may be certain sentences which are not needed for further exploration. The above work is a part of data extraction and not a part of pre-processing. The step for data preprocessing is explained further in the implementation chapter.

3.4 Methods to explore FMEA with Ontology

The failure mode and effect analysis consists of tremendous knowledge in terms of the effectiveness of the manufacturing process, which can act as a



Row ID	S ID	S Text
Row0	U596085D...	[redacted] Primary function [redacted]...
Row1	U595E8C0...	FPG3-quadripole drive converts the electrical power to mechanical power, with ...
Row2	U6294C32...	FPG3-quadripole drive generates a rotary movement in a defined direction depe...
Row3	U596085F...	[redacted] Secondary function [redacted]...
Row4	U61E098F...	FPG3 including electronics is reversed (change of the direction of rotation) durin...
Row5	U595E8B5...	FPG3 four-pole drive absorbs breakage- and operating load (load spectrum) via ...
Row6	U596085B...	[redacted] Tolerable side effects [redacted]...
Row7	U62D8E7F...	FPG3 4-pole drive dissipates max. magnetic flux into environment
Row8	U5DA66D4...	FPG3-quadripole drive does not deliver an over voltage to the electrical system
Row9	U596AB9E...	FPG3 four-pole drive complies with the maximum heat dissipation (surface temp...

Figure 3.4: Extracted sentences from XML file using Xpath queries

preventive measure for potential failure. It defines in advance the function of each sub-part used to create the final product, its behavior when joined or welded to other sub-parts. Therefore, if knowledge from the FMEA can be accumulated reasonably it can assist in timely decision support. However, the integration of FMEA product and process knowledge is a challenging task.

For example, given a sentence *"FPG3-quadripole drive converts the electrical power to mechanical power, with an objective to provide the defined rotation speed."*

The task is to identify various entities and relation between the entities. In the above example, *FPG3-quadripole drive* is a product, also know as a window lift motor which uses another entity *electrical power* and transforms it into *mechanical power*. Initially an ontology based architecture was proposed to extract the entity-relation-entity triple. Figure 3.5 shows a diagram of the proposed architecture. Ontology provided a basis for relation extraction task. Explained further is an example to demonstrate the functioning of the architecture.

Raw Text: "Kalpana Chawla was born on July 1, 1961. She was an Indian by birth and naturalized to United States. She was the first woman of Indian origin to go on a space mission as an astronaut."

Following are the triples expected to be extracted from the sample raw text data:

Kalpana Chawla(PERSON), date_of_birth,July 1, 1961(DATE)

Kalpana Chawla(PERSON), nationality_at_birth, India(LOCATION)

Kalpana Chawla(PERSON), nationality_at_death, United States(LOCATION)
Kalpana Chawla(PERSON), profession, Astronaut(JOB TITLE)

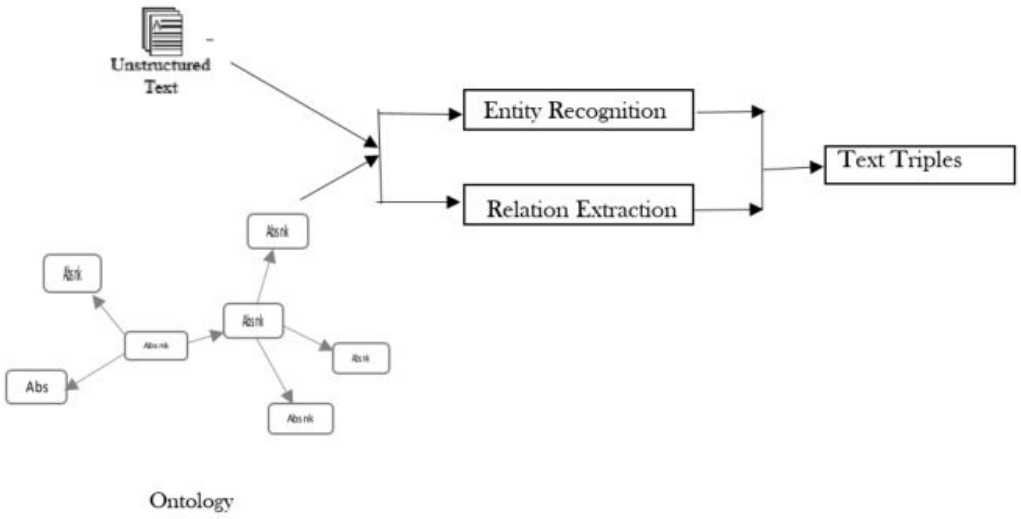


Figure 3.5: Proposed architecture for entity-relation-entity extraction

There are software and libraries to identify certain named entities belonging to classes such as *Person, Organization, Location, etc.* Also, based on the two entities recognized in a sentence, only a certain number of relations can be established. For example, it is not possible to have a *Nationality* relationship between two entity types belonging to entity type Person and Organization. These types of attributes assist in providing or denying possible entity-relation-entity triple thus simplifying the process of triple extraction. However, in the case of the data obtained from FMEA, the classes of entities and their relations are not known beforehand, since the data is extremely domain and company-specific. Initially, attempts were made to utilize existing ontologies to identify classes of products, part of products and establishing relations such as *part_of* relation between the recognized entities. However, they failed when it comes to domain and company-specific parts and products that are not recognized universally and not defined under any existing ontology classes. Also, creating ontologies from scratch is an extremely time-consuming, expensive process that requires a tremendous amount of domain knowledge.

3.5 Methods to explore FMEA without Ontology

Due to the challenges faced in utilizing existing ontology or creating ontology, an end-to-end model was used for the triple extraction without the use of ontology. The new approach attempts to jointly learn to extract the entities which contain semantic relations. To perform the task we take help from SemEval(Semantic Evaluation) 2010 Task 8 dataset, which is a series of evaluations of computational semantic analysis system in the field of Natural Language Processing and Computational Linguistics. The dataset attempts to identify word senses and investigate interrelationship among entities within a sentence [8].
The SemEval-2010 task 8 dataset provides a Multi-way classification of semantic relation between pairs of nominals [8]. It provides a set of 9 relations that exists between pair of entities. For example, in a sentence *Smoking causes cancer* *Smoking* and *cancer* are in a *Cause-Effect* relation with each other, Smoking being the cause and cancer being the effect. The SemEval-2010 task 8 dataset provides a testbed for automatic classification of relations. It was observed that most of the entities present in FMEA dataset belonged to one of the relations specified in this testbed. Therefore, we attempt to use this as training dataset to extract semantic relations from FMEA.
At the start, entities have to be labelled manually, however, relation such as *Component-Whole* helps establish a *Entity1* $\rightarrow$ *is part of* $\rightarrow$ *Entity2* relation for knowledge graph creation. The identified entities and relations can also serve as a basis for domain ontology creation.

3.6 Problem formulation

As discussed, for this thesis, we use the method without the ontology to explore FMEA. The main task of this thesis is information extraction from FMEA sentences. Within each sentence, an entity-relation-entity triple needs to be identified and extracted. The open-source SemEval-2010 task 8 dataset is used as a training dataset for relation classification, which contains sentences in entity-relation-entity triple format. It is tested on sentences from FMEA which also consist of entity-relation-entity triple. The test dataset is created by domain experts from the ABCD group with their knowledge in identifying entities and relationships among those entities. The information extraction from FMEA serves as the basis for creating a domain-specific semantic knowledge graph used for Industry 4.0 data access applications. As

the data is domain and company-specific, therefore the entity names used in the FMEA had to be labeled manually. We try to search for a lexicon or an entity term index, however, there was no such available document and therefore the labelling had to be done manually at least for the initially phase of the process. Later, A neural network such as convolutional neural network and BiLSTM with attention mechanism was used for the relation classification task. Each sentence containing two entities were classified based on the type of relationship that existed between the two entities. A supervised learning approach was used to determine the performance of neural networks on the domain-specific test data. As the training dataset is a generic open-source dataset and the test dataset is highly domain and company-specific, a supervised approach was chosen for this thesis. An alternative would have been a weakly supervised method however due to lack of domain expertise and limitations in access to domain experts it would have been difficult to identify the model performance.
Chapter 4 explains in detail the datasets description and machine learning techniques used to extract domain specific relation extraction.

Chapter 4

Implementation

4.1 Training Dataset Description

As discussed in the previous chapter, SemEval-2010 task 8 dataset has been used for training data. It contains nine relations that are broad enough to cover general and practical interest without overlaps. The following are the relations and descriptions as proposed by Hendrickx et al., 2010:

Cause-Effect (CE) An event or object leads to an effect. Example: those cancers were caused by radiation exposures.

Instrument-Agency (IA) An agent uses an instrument. Example: phone operator.

Product-Producer (PP) A producer causes a product to exist. Example: a factory manufactures suits.

Content-Container (CC) An object is physically stored in a delineated area of space. Example: a bottle full of honey.

Entity-Origin (EO) An entity is coming or is derived from an origin. Example: letters from foreign countries.

Entity-Destination (ED) An entity is moving towards a destination. Example: the boy went to bed.

Component-Whole (CW) An object is a component of a larger whole. Example: my apartment has a large kitchen.

Member-Collection (MC) A member forms a nonfunctional part of a collection. Example: there are many trees in the forest.

Message-Topic (MT) A message, written or spoken, is about a topic. Example: the lecture was about semantics.

The training dataset is a tab-separated file (.tsv) containing 8000 sentences classified into 9 relations as mentioned above. However, on comparing the

FMEA sentences, it consisted of mainly 4 relations namely, Cause-Effect, Product-Producer, Component-Whole, and Entity-Destination. Therefore, the training dataset was filtered to contain only the four relations. The filtered training data set consists of a total of 3391 sentences. The relation distribution of the training data is shown in figure 4.1. The implementation was done using CNN and BiLSTM neural network models on training data and tested on the FMEA sentences.

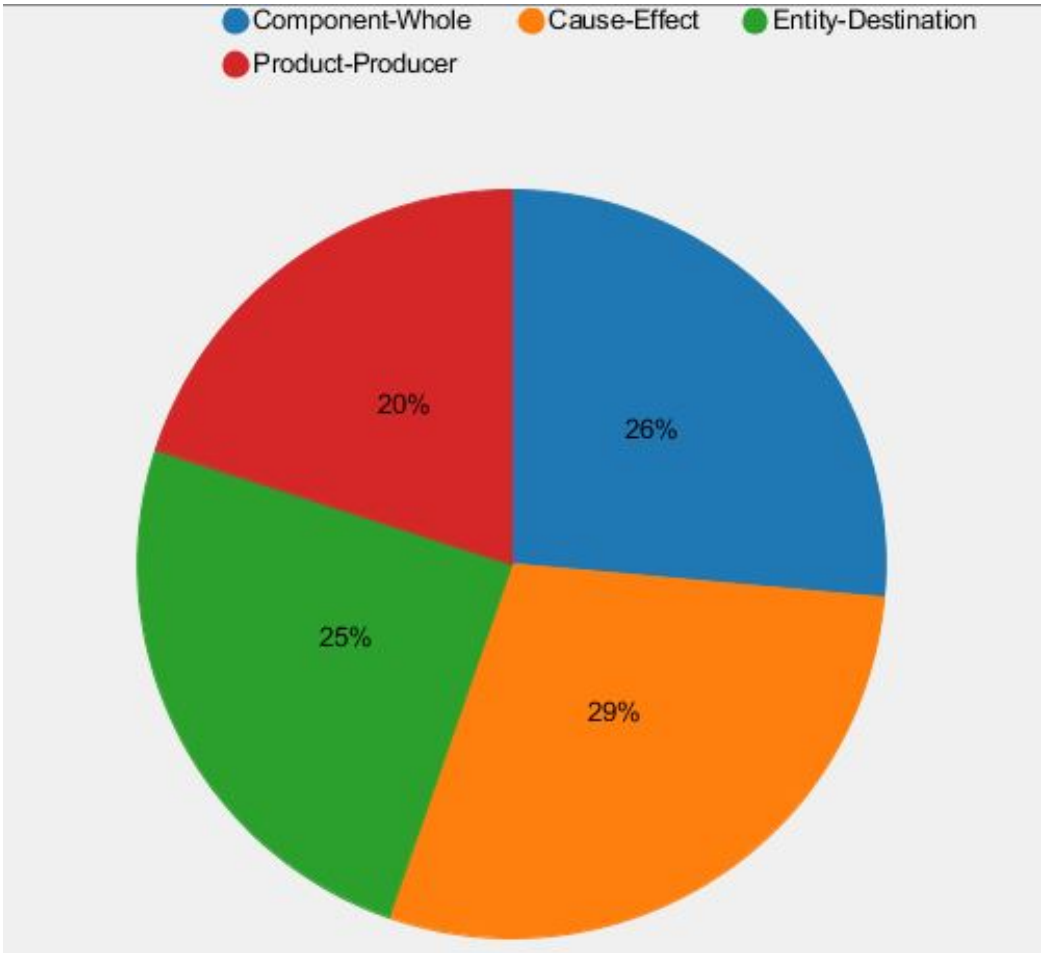


Figure 4.1: Training dataset distribution

4.2 Test Dataset Description

The test data was created with the help of domain experts from the ABCD group, as mentioned, the sentences can be classified only in the four rela-

tion categories. A test dataset was created with 292 sentences distributed equally among the 4 relation class labels. As the entities are company and domain-specific it was manually labeled. An effort was made to find a company specific lexicon or an term index which provides entities, however the decision for manual labelling was taken when no such lexicon was found. Given below are some examples of sentences from test dataset:

Cause-Effect (CE) An event or object leads to an effect. Example: Lubricant reduces friction coefficient.

Product-Producer (PP) A producer causes a product to exist. Example: Quadripole drive converts the electrical power to mechanical power.

Component-Whole (CW) An object is a component of a larger whole. Example: Wrong material number on Polehousing.

Entity-Destination (ED) An entity is moving towards a destination. Example: Quadripole drive delivers an over voltage to the electrical system.

4.3 Data preprocessing

Data processing involved row filtering to exclude the rows not needed for the experimental setup. It involved removing punctuation marks from the sentences. However, stop-word removal was not performed on the data as it is important in the context of this setup. For example, stop-word such as “in” can be important in the context of a Component-Whole relationship. It is a strong indicator that “entity1” is “in” entity2“ thus helping to establish the relation between the two entities.

4.4 Experimental Setup 1

This thesis uses mainly 2 methods to perform the relation classification task and provides a comparison between the two methods. Relevant technical papers and literature review suggested better results from deep learning methods than older machine learning methods such as Support vector machines, random forest etc. Therefore only deep learning techniques were selected for this thesis. The first setup uses BiLSTM with a unique attention mechanism to perform the specified task. As mentioned, the filtered dataset from SemEval consisting of only 4 relation types, is considered for training. Relation classification task differs from sentiment classification and other language models in terms of the features it consists of such as entities positions, lexical resources, part of speech tags, etc. Traditional classification methods pro-

pose to utilize these features for the classification task and have also proven to achieve high performances. The downside of these techniques is increased computational cost and additional propagation errors [30]. Besides, because we are using domain-specific test data might lead to poor performance. For example, part of speech tag for "Drive" in the English language is "Verb", but in the context of domain data it means an entity such as "four-pole drive" which is a physical component and not a verb. We, therefore, use attention-based BiLSTM to perform the classification(Zhou et al., 2016). Figure 4.2 shows the model using BiLSTM and unique attention mechanism for relation classification.

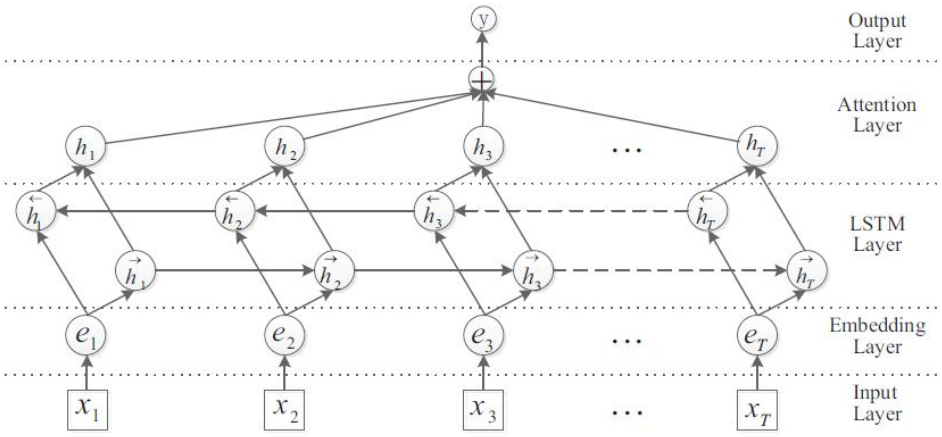


Figure 4.2: Bidirectional LSTM model with Attention [30]

The input layer consists of the sentences in the form of raw text. It is later transformed into an embedding layer as required by the neural networks. Given a sentence S consisting of x_t words, every word is converted to a real-value vector for each word. Also, due to irregular sentence length, all sentences are padded such that they have equal lengths. The model is also tested with pre-trained vectors $GloVe$[1] to compare performances. The third layer of the model consists of a bi-directional LSTM layer. As shown in figure 4.2 the network contains two-sub networks to cover context from direction left-to-right and later right-to-left, namely forward and backward passes respectively [30]. The forth layer is a unique attention mechanism as proposed by (Zhou et al., 2016). Consider H as an output matrix from LSTM layers. Therefore, there are $[h_1, h_2, h_3, ...h_T]$ output vectors from the previous BILSTM layer considering T is the length of the sentence. The

[1]https://nlp.stanford.edu/projects/glove/

attention mechanism proposed is as follows:

$$M = tanh(H)$$

$$\alpha = softmax(w^T M)$$

$$r = H\alpha^T$$

$$h^* = tanh(r)$$

where d^w is the dimension of word vector, w^T is the transpose of the trained parameter vector and the dimensions of w, α, r is d^w, T, d^w respectively. Lastly, a softmax classifier is used to predict class, as the probability of the highest class needs to be considered for the output. A combination of L2 regularization and dropout is used to avoid overfitting. A comparison of performances of various regularization parameters, word vector dimension, and other hyperparameters are discussed in detail in the evaluation section of the thesis.

4.5 Experimental Setup 2

In the second experimental setup, CNN is used with a combination of pre-trained word-embedding for relation classification. This method is selected as one of the experimental setups since deep learning models have achieved remarkable results in natural language processing and classification tasks. The model architecture [11] as explained performs extremely well on sentiment analysis dataset such as movie reviews (Pang and Lee, 2005) [17], subjectivity dataset (Pang and Lee, 2004) [16], Stanford sentiment treebank (Socher et al. (2013) [11], TREC question dataset (Li and Roth, 2002) [15] and customer reviews(Hu and Liu, 2004) [9]. The model architecture is as shown in the figure 4.3. Each sentence from the training dataset is transformed into word embedding where pre-trained word vectors are used to prepare the embedding matrix as shown in the first part of figure 4.3. Two types of variations are used during embedding, a static model uses the pre-trained word embedding as is, without training it based on the context of the sentence. The other is a non-static embedding where we train the embedding with the context in each sentence. The vector dimensions are ranging from 50 to 300, where publicly available dataset *word2vec*[2] vectors which are trained on Google news dataset having a dimension of 300 and *GloVe*[3] embedding with dimension 50,100,200 and 300 are used an input to the model architecture.

[2]https://code.google.com/archive/p/word2vec/

[3]https://nlp.stanford.edu/projects/glove/

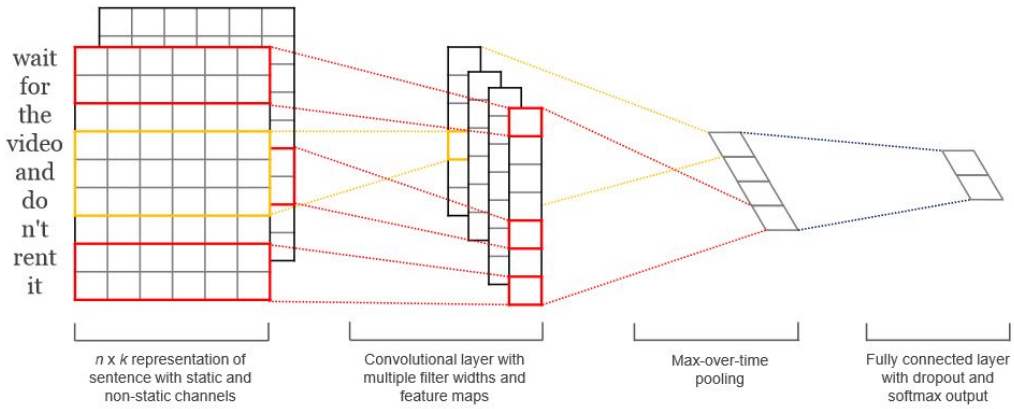


Figure 4.3: Model architecture used for relation classification task as an example sentence [11]

Once the pre-trained embeddings are loaded, an index mapping is created to the known embeddings by parsing the loaded pre-trained embeddings. An embedding layer is created using the embedding matrix and input text sequence, here the embeddings can be kept trainable or non-trainable. The values of embeddings change on the sentence input based on the setting used during the embedding layer. In our setup, we use both the settings to compare the results. Also, the input should be padded because sentences cannot have variable length and the embedding layer accepts only uniform length input shape. A combination of 1D convolution layer and Max pooling is used for the classification problem. Standard filter sizes of 2,3,4,5 are used is the convolution layer to consider bi-grams, tri-grams and context from around the word. Since the convolution layer does not consist of many parameters, the additional gain on using regularization techniques such as dropout may not add performance gain or add a low-performance gain [7]. Finally, the softmax activation is used to obtain the probabilities of classification for each relation label value. Pre-trained word vectors have proved to provide excellent accuracy up to 91% on the mentioned datasets. However, it is also important to note that although trained dataset used in an open-source dataset covering a wide range of use cases, the test dataset is highly domain-specific. The domain-specific dataset might consist of words such as *drive* which refers to a physical part of a component. However, the pre-trained words might be trained on the same words having different meanings or context affecting performances. The results and model evaluation are discussed further in Chapter 5 and Chapter 6.

Chapter 5

Evaluation

In this chapter, we evaluate the results obtained during the two experimental setups as discussed in Chapter 4. Initially, quality metrics and formulas used for quality metrics are discussed. Later, a comparison is provided within the experimental setup for various hyperparameters, then the best performing hyperparameters within each setup are compared against each other to determine which method performs better for the task of relation classification.

5.1 Quality Metrics

The most commonly known quality metrics for a neural network is accuracy. However, accuracy only considers the factors which are truly classified ignoring the false rates. In a real-life application, a wrongly classified example can have serious consequences for example if the model is trying to classify if a person is carrying a virus that can spread quickly then false positive can have a serious impact. Therefore, precision, recall, and f1 scores are important performance indicators of a model rather than just the accuracy.
The formulas for Accuracy, Precision, Recall, and F1 score are calculated as per the formulae given in 5.1, 5.2, 5.3 and 5.4:

$$accuracy = true\_positive/(true\_positive + true\_negative) \tag{5.1}$$

$$precision = true\_positive/(true\_positive + false\_positive) \tag{5.2}$$

$$recall = true\_positive/(true\_positive + false\_negative) \tag{5.3}$$

$$f1\_score = 2 * (precision * recall)/(precision + recall) \tag{5.4}$$

5.1.1 Accuracy

The accuracy defines the number of instances which are truly classified as positive, only among the true negative and true positive instances.

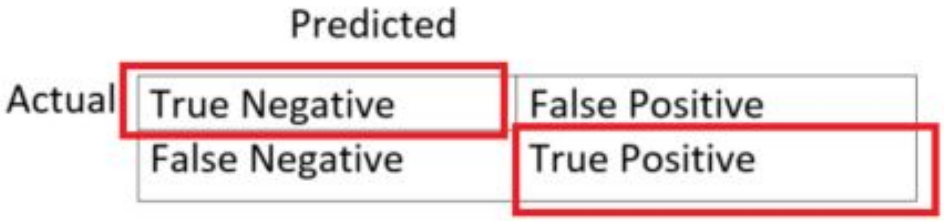


Figure 5.1: Accuracy in terms of Confusion matrix

5.1.2 Precision

Precision is used to identify how precise the model performs taking into consideration the truly classified example compared to the total predicted positive. It gives a good way to measure if many examples are falsely classified, then the precision will be low and vice-versa.

Figure 5.2: Precision in terms of Confusion matrix

5.1.3 Recall

Recall considers the truly classified among the true positive and false negative instances. This is helpful when the penalty for false negative is high such as fraud detection cases for a bank. Here, it can be very critical if a fraudulent transaction is overlooked, the company can be at risk.

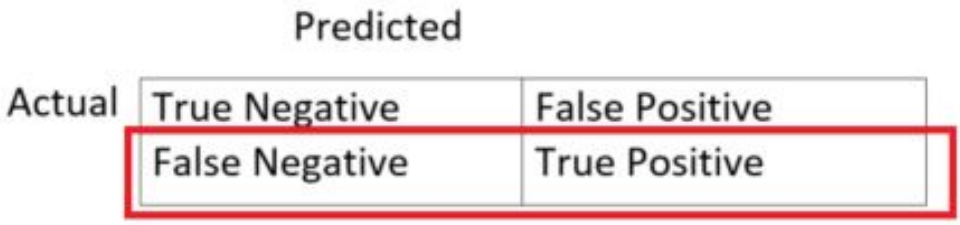


Figure 5.3: Recall in terms of Confusion matrix

5.1.4 F1-Score

F1-Score provides a balance between precision and recall as it takes both the measures into account. Often, false positive and false negative both have a tangible or intangible business cost associated with it and F1-score, as shown in equation 5.4, is a good measure for a performance indicator.

5.1.5 Training and Validation loss

During each epoch, the training and validation loss is calculated. This is a good way to determine if the model over-fits or under-fits the data. In our experiment it is an important factor that the model should not over-fit the data as a generic training dataset is used for training purpose and highly domain-specific data is used for testing. Therefore, overfitting can lead to bad accuracy and f1-score for test data. When training loss is greater than validation loss, it is an indicator of under-fit. When training loss is much greater than validation loss it means the model fits nicely on the training data but it is an indicator than it may not generalize on test data and can perform poorly on test data. The ideal scenario is when values of training loss and validation loss are close to each other means the data is not only performing well on training data but also is very relevant for test data.

5.2 Results

As two types of the experimental setup are used for the relation classification task, this section concentrates on the best possible outcome within each setup. Different hyperparameters such as the number of epochs, regularization techniques, word embedding techniques, dropouts, attention mechanism, etc. are compared to find the best possible results.

5.2.1 Results of Experimental Setup 1

The first setup uses BiLSTM model with unique attention mechanism for the classification task. Hyperparameter selection is often a tough task, various parameters can be taken into consideration while training models such as learning rate, optimizer, word embedding dimension, and dropouts. Each parameter affects the performance of the model and finding the right fit can often be a very challenging task. Therefore, parameters are chosen from relevant papers (Zhou et al., 2016) and fine-tuned to observe its performance on the test dataset. The first hyperparameter chosen was the different dimensions of word embedding, 50 and 100 dimensions of word embedding are selected for the experiment. Pre-trained word embeddings such as *GloVe* is available only in a few dimension and therefore there are not many options to experiment. However, different dimensions can be used for non-pre-trained embeddings but for purposes of result comparison, the same dimensions of 50 and 100 were chosen. The higher dimension word embedding of 100 is seen to give better results as compared to dimension 50 word embedding. The word embedding trained on the data(non-pretrained embedding) gave better results on test data than pre-trained *GloVe* word embeddings. Also, even among pre-trained and non-pre-trained embeddings, the higher dimension embedding had better results.

Table 5.1: Hyperparameter Optimization Results for optimum learning rate

Type of Embedding	**Word Embedding Dimension**	**Learning Rate**	**(Precision, recall, F1)**	**Test Accuracy**	**Train Accuracy**
Non-pretrained	50	0.001	(0.60, 0.63, 0.61)	0.59	0.74
Non-pretrained	**100**	**0.001**	**(0.68,0.66,0.66)**	**0.61**	**0.92**
Glove	50	0.001	(0.57,0.54,0.52)	0.49	0.69
Glove	100	0.001	(0.58,0.56,0.53)	0.53	0.71

The learning rate at the start is kept as 0.001 for both the word embedding dimension to obtain comparable results. Later, the learning rates of 0.01 and 0.05 were also used for both the embedding dimension. The experiment was run for 20 epochs to analyze the loss and accuracy after each epoch. The loss for lower learning rate is gradual but the accuracy improves significantly when using higher epochs. The loss reduces faster with higher learning rates but a lot of fluctuations are observed during every epoch. The optimizer was kept standard throughout the setup. Going ahead, different dropout rates of zero, 0.3, 0.5 were checked to compare results. Dropouts were used in two positions of the model architecture, one dropout was placed for the embedding layer and others after the bidirectional LSTM layer. Lower dropout rates provided better results in our experiment compared to higher dropouts. An activation

map shows the words on which attention layer focuses, the word weights learned during training is used for classification on the test data. Figure 5.4 shows the attention map of truly classified and falsely classified sample and the words on which higher weights are focuses to determine the classification. The falsely classified example consists of a high emphasis on the word weight *in* due to which it has been classified as Component-whole relation between the two entities but the true relation is that of Product-Producer.

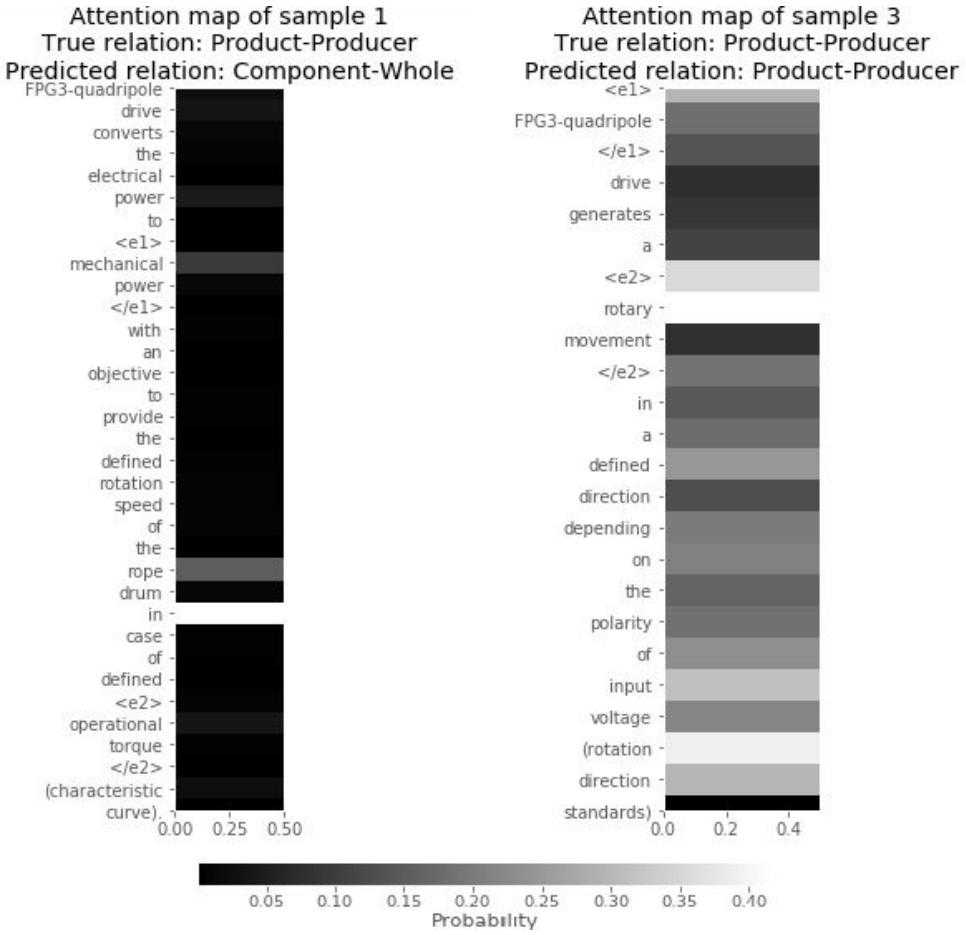


Figure 5.4: Attention map for BILSTM model

The training accuracy, test accuracy, f1 score, precision, and recall provide the overall results of the model. However, there is a possibility that the scores differ for each relation class label. Therefore, a classification report is also printed for each variation in the hyperparameter setting to determine the scores for each class. Figure 5.5 provides individual scores for each class label. We see that f1 scores for Entity-Destination relation for test data are

very high however, the average reduces due to low scores for Cause-Effect relation.

```
In [37]: print(classification_report(rclf.encoder.transform(test_rel), y_test_pred,
                                    target_names=label_to_use,
                                    labels=label_idx_to_use))
```

	precision	recall	f1-score	support
Cause-Effect	0.57	0.38	0.45	79
Component-Whole	0.47	0.68	0.56	75
Entity-Destination	0.89	0.91	0.90	34
Product-Producer	0.81	0.67	0.73	45
accuracy			0.61	233
macro avg	0.68	0.66	0.66	233
weighted avg	0.63	0.61	0.61	233

Figure 5.5: Classification report

5.2.2 Results of Experimental Setup 2

The second experimental setup uses simple CNN architecture to perform the relation classification task. For this setup, only pre-trained word embeddings were used of different dimensions namely 50, 100, 200 and 300. The learning rate was kept at a value of 0.001 to compare the various word embedding dimension. Once the optimum word embedding dimension was found, different optimizers were used to compare the performance of the training and test dataset. The test data gave the best results for a word dimension of 100. The training accuracy improved significantly with a higher dimension of word embeddings such as 200 and 300 and the training loss was very low. However, the model overfit the training data and resulted in lower test accuracy.

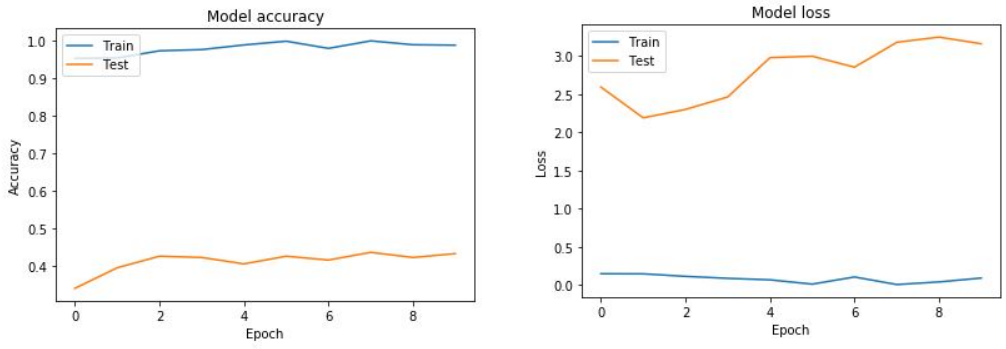


Figure 5.6: Model accuracy and loss for 10 epochs for RMSprop

For further hyperparameter tuning, the word embedding dimension was kept constant at 100. Various optimizer was tested with a constant learning rate of 0.001 and word embedding dimension of 100. Optimizers such as RMSprop, Stochastic Gradient Descent(SGD) and Adam were using for the model to check performances. RMSprop provided better results among others. However, the training and validation loss diverged for all the optimizers where ideally the loss should converge. RMSprop and Adam optimizers reached high accuracy with a few epochs, but the test accuracy remained constant. SGD slowly fit the data, however, training and validation loss also gradually showed a downward trend. Figure 5.7 shows the model accuracy and the model training and validation loss with SGD optimizer.

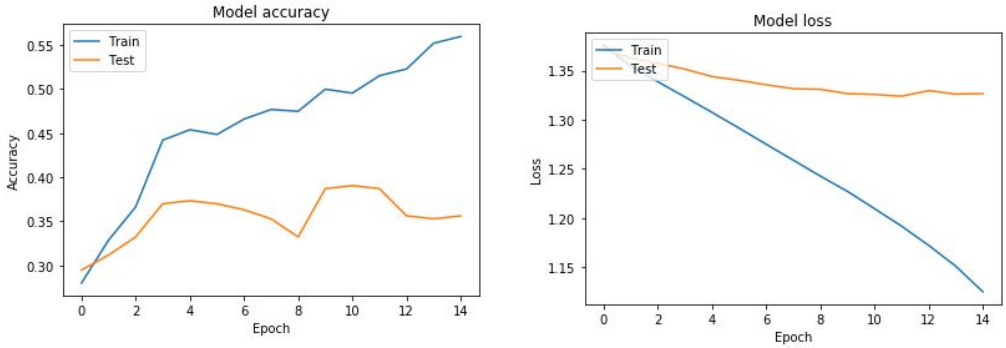


Figure 5.7: Model accuracy and loss for 10 epochs for SGD

A dropout of 0.5 and 0.3 was also used in the model to find out if it prevents overfitting of data on the training dataset. However, no significant improvement was seen in the test accuracy with the dropouts. The best overall f1-score achieved with the hyperparameter tuning of CNN model architecture was 42%. In a multi-class classification problem, it is also useful to know the individual class, therefore a classification report is also printed for the CNN model as shown in figure 5.8.

Table 5.2 provides a training accuracy, test accuracy, f1-score, precision, recall and other hyperparameters used for CNN model.

5.3 Comparison between the two experimental setup

On observing the results from both the experimental setup, it is seen that in both the cases the model caused overfitting on training data. However, out

```
In [37]: print(classification_report(predict_class1, predict_class, target_names=target))
```

	precision	recall	f1-score	support
Product-Producer	0.35	0.41	0.38	79
Entity-Destination	0.43	0.41	0.42	85
Cause-Effect	0.67	0.58	0.62	64
Component-Whole	0.28	0.28	0.28	64
accuracy			0.42	292
macro avg	0.43	0.42	0.42	292
weighted avg	0.43	0.42	0.42	292

Figure 5.8: Model accuracy and loss for 10 epochs for SGD

Table 5.2: Hyperparameter Optimization Results for optimum learning rate

Type of Embedding	**Word Embedding Dimension**	**Optimizer**	**(Precision, recall, F1)**	**Test Accuracy**	**Train Accuracy**
Glove	50	RMSprop	(0.44,0.41,0.42)	0.40	0.98
Glove	**100**	**RMSprop**	**(0.43,0.42,0.42)**	**0.42**	**0.98**
Glove	200	RMSprop	(0.40,0.36,0.37)	0.35	0.99
Glove	300	RMSprop	(0.41,0.41,0.41)	0.40	0.99
Glove	100	Adam	(0.45,0.40,0.41)	0.40	0.99
Glove	100	SGD	(0.30,0.36,0.30)	0.35	0.55

of both the setups BiLSTM model with an attention mechanism provided better results.

Chapter 6

Discussion

6.1 Results Interpretations

Results obtained for relation classification using BiLSTM (test accuracy = 61% and training accuracy = 92%) were better compared to CNN(test accuracy = 42% and training accuracy = 98%). It is seen that BiLSTM is trained on task-specific word embedding and also pre-trained word vectors such as GloVe. Whereas CNN is only trained on pre-trained word embeddings. Therefore, we see that although CNN achieves a higher training accuracy, it does not perform better on test dataset. However, even though BiLSTM provides a comparatively lower training accuracy it performs better on test data than CNN. Even in the experimental setup for BiLSTM when compared within the two types of embedding, an embedding layer trained on the data is seen to perform better. The training loss reduces faster with pre-trained embedding on the training data in both the models however, it overfits the training data leading to lower performance on test data. Learning rate is a hyperparameter which controls the rate of change of the model with respect to estimated error rate. Usually, lower learning rates leads to lower convergence rates and higher learning rate leads to higher convergence rates. However, choosing lower learning rate with high training data leads to long training time and also the model can get stuck. Since our experiment consisted of small training dataset, a lower learning rate of 0.001 was possible and it was observed that lower learning rate performed best among the learning rate such as 0.01, 0.05 etc. Higher learning rate such as 0.01 provided better training accuracies within fewer epochs but it did not perform well on test data. Dropout is a good technique for regularization, however it did not help improve accuracy to a great extent. Lower dropout rates seemed to perform well on both training and test data. Optimizers such as

RMSprop, SGD and Adam were used, and RMSprop gave the best results. However, it was seen that SGD converged slowly and with higher epochs it might have been a better fit. Attention mechanism used during the BiLSTM experimental setup helped improve the accuracy. Relation classification differs from simple sentence and document classification in a many ways. For example, document classification for news dataset with classes such as sports and political news tries to focus on words which might occur in sports but not in politics and visa Versa. However, with relation classification task, preposition words such as "in", "on" have higher significance than in document classification. In document classification these words can easily be treated as stop-words. Figure 5.4 shows how the attention on word "in" leads to a wrong classification to a Component-Whole relation whereas the actual relation is that of "Product-Producer". Also, one of the reasons why embeddings trained on dataset performed better than pre-trained embedding such as GloVe, since words holding importance in relation classification tasks are less conceptually trained.

6.2 Comparison with Related Work

SemEval is an ongoing series of evaluations for semantic analysis of the relation between two entities. Many research papers have been written using the same dataset for evaluation, therefore the results can be compared with each other. Many papers have been written to evaluate the SemEval-2010 Task 8 dataset and obtained state-of-the-art results. Authors Shanchan Wu and Yifan He[27], recently wrote a paper for Relation classification based on Google's Bidirectional Encoder Representations from Transformers (BERT) model [5]. R-BERT (Wu et al. 2019) attains an f1-score of 89.25 on the semantic relation evaluation test dataset which is the highest. There are other models based on the CNN architecture such as Multi-Attention CNN (Wang et al. 2016) [23] achieves f1-score of 88.0, Attention CNN (Huang and Y Shen, 2016) [10] achieves f1-score of 85.9 with use of features like WordNet, part-of-speech tags etc. Also, RNN based models such as Entity Attention Bi-LSTM (Lee et al., 2019) [14] and Hierarchical Attention Bi-LSTM (Xiao and C Liu, 2016) [28] achieve f1-scores of 85.2 and 84.3 respectively on the semantic evaluation test dataset. However, the difference is that these models are tested on generic semantic relations test dataset much similar to training dataset but our experiment uses a high domain-specific test dataset based on the automotive industry. Therefore, the certain words used in the generic data set such as *drive* consists of a generic meaning of controlling and operating a motor vehicle whereas in domain-specific data it refers to a component

or a part.

6.3 Limitations

- The experimental setup consists of only four relation types, however, the data consists of several other relations that were not able to consider for this thesis due to limited domain expertise and lack of pre-defined domain ontology.
- Some sentences considered in the test dataset consisted of multiple entities, however for the purpose of the experiment only sentences consisting of two of the most important entities were chosen for the relation classification task.
- There is a possibility for the same entity pair to have more than one relation. For example, some sentences consisted of same entity pair having a Component-Whole and Entity-Destination relation as the entity1 was a part of entity2 and also entity1 was the last part in the process, thus making it a destination of entity2. The type of sentences consisting of multiple relations among the same entities was not considered in the process of test dataset creation as it is lacking in training dataset. However, these type of relationships is useful for future work of creating knowledge graph from extracted relations.
- Due to limited time and domain expertise only 3391 and 292 sentences were considered for training and testing purposes. It is known that Neural networks perform well with huge datasets.
- Due to lack of lexicons, an entity recognition model was not possible with this project. The project only attempts to establish the relation classification for the sentences.

6.4 Possible extension and future Work

- The experiment only considered word embeddings as feature engineering. However, related works in Semantic relation classification uses other features such as position embeddings, part-of-speech tag embeddings, lexical features, wordnets and entity attention models which can be explored to improve the results of the experiment.

- The experimental setup only considers CNN and RNN based neural network, however, a recent paper discusses an R-BERT model(Wu et al. 2019) based on a Google's BERT model achieves state-of-the-art results for relation classification which can be explored.

- Information extraction from FMEA is an ongoing process, also it is a complex document with sentences consisting of multiple entities and relations. Therefore, a CNN and RNN model with reinforcement learning can be explored for better results.

- Lastly, the information extraction task is only the first step towards Knowledge graph creation. A domain ontology needs to be created before the knowledge graph creation, which specifies different kinds of entities contained in the domain and the semantic relation each type of entity possess. For example in a sentence, *Lubricant reduces friction coefficient*, Lubricant is identified as an entity however the type of entity is not defined. Once an ontology is created it provides a base for knowledge base population task namely, slot filling and entity linking. Slot filling completes all the information related to a given entity. For example, entity Lubricant is classified as type Component then the system's goal is to collect all the semantic relations such as type, attribute, relation, function etc [1]. The second aspect is entity linking which attempts to resolve ambiguities related to entities described in the text data. It is similar to coreference resolution across multiple FMEA documents consisting of the same products and its variants[3].

Chapter 7

Conclusions

The work of this thesis performs semantic relation extraction between entities from FMEA document. This aims to serve as the basis for company-specific I4.0 knowledge graph creation. FMEA is an analytical method of preventive quality management conducted during the development of product and process in the manufacturing industry. FMEA is a critical document, allowing organizations to anticipate failures during the design stage of a product and process. A poorly designed FMEA can result in product recalls from customers and high costs to the organization. Since this document can be of tremendous help to the organization it was used as a starting point for the creation of semantic data integration. Entities and semantic relation between the entities need to be extracted from the document to create the semantic information model. Semantic data integration is a potential solution to reduce prepossessing time spend in data analytics due to a large number of heterogeneous data produced within the organization. The data contained within the FMEA is highly company and domain-specific, therefore generic libraries used for information extraction tasks such as named entity recognition and relation extraction cannot be used directly to identify entities and relations. Therefore, initially, an attempt was made to create a domain-specific ontology which specifies entities and relations within the company domain. However, domain ontology requires huge amount of domain expertise. Therefore, SemEval-2010 Task 8 was used as a training dataset to extract relations between domain-specific entities. A small test dataset was created by domain experts from the industry to test the accuracy on the FMEA documents. Since the entities are company-specific and also lacking domain lexicons to identify the entities, it was labelled manually during the creation of test dataset. The thesis uses an attention-based BiLSTM, and CNN model to extract semantic relations between the selected entities in the test dataset. The attention BiLSTM model achieves an f1-score of 0.66 on

the relation extraction task. Total accuracy achieved on training dataset is 94% and test dataset is 61%. The CNN model achieves an f1-score of 0.42. The total accuracy achieved for the CNN model on training data is 96% and on the test dataset of 42%. The high domain-specific data caused certain limitations to the results however, several methods are proposed in the future work that can be explored to improve the results significantly in the system.

Bibliography

[1] Angeli, G., Gupta, S., Premkumar, M. J., Manning, C. D., Ré, C., Tibshirani, J., Wu, J. Y., Wu, S., and Zhang, C. Stanford's distantly supervised slot filling systems for kbp 2014. In *Proceedings of the Seventh Text Analysis Conference* (2014).

[2] APIS Informationstechnologien GmbH. FMEA Solutions— APIS IQ-Software — CARM-CARM CN Server. `https://www.apis-iq.com/software/solutions/`, 2019. [Online; accessed 26-August-2019].

[3] Clark, K., and Manning, C. D. Improving coreference resolution by learning entity-level distributed representations. *arXiv preprint arXiv:1606.01323* (2016).

[4] Devezas, T., and Sarygulov, A. *Industry 4.0*. Springer, 2017.

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

[6] English, L. D. . entity — definition of entity in english by lexico dictionaries. webpage, 2019. `https://www.lexico.com/en/definition/entity`.

[7] Gal, Y., and Ghahramani, Z. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158* (2015).

[8] Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Séaghdha, D. O., Padó, S., Pennacchiotti, M., Romano, L., and Szpakowicz, S. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of SemEval-2* (Uppsala, Sweden, 2010).

[9] Hu, M., and Liu, B. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (2004), ACM, pp. 168–177.

[10] Huang, X., et al. Attention-based convolutional neural network for semantic relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (2016), pp. 2526–2536.

[11] Kim, Y. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[12] KNIME AG. KNIME Open for Innovation. `https://www.knime.com/`, 2019. [Online; accessed 27-August-2019].

[13] LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature 521*, 7553 (2015), 436.

[14] Lee, J., Seo, S., and Choi, Y. S. Semantic relation classification via bidirectional lstm networks with entity-aware attention using latent entity typing. *Symmetry 11*, 6 (2019), 785.

[15] Li, X., and Roth, D. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1* (2002), Association for Computational Linguistics, pp. 1–7.

[16] Pang, B., and Lee, L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics* (2004), Association for Computational Linguistics, p. 271.

[17] Pang, B., and Lee, L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd annual meeting on association for computational linguistics* (2005), Association for Computational Linguistics, pp. 115–124.

[18] Quan, C., and Ren, F. Gene–disease association extraction by text mining and network analysis. In *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi)* (Gothenburg, Sweden, Apr. 2014), Association for Computational Linguistics, pp. 54–63.

[19] Schwab, K. *The fourth industrial revolution.* Currency, 2017.

[20] Shashank Gupta. Named Entity Recognition: Applications and Use Cases. https://towardsdatascience.com/named-entity-recognition-applications-and-use-cases-acdbf57d595e, 2019.

[21] Varsamopoulos, S., Bertels, K., and Almudever, C. G. Designing neural network based decoders for surface codes. *arXiv preprint arXiv:1811.12456* (2018).

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

[23] Wang, L., Cao, Z., de Melo, G., and Liu, Z. Relation classification via multi-level attention CNNs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (Berlin, Germany, Aug. 2016), Association for Computational Linguistics, pp. 1298–1307.

[24] Wikipedia contributors. Information extraction — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Information_extraction, 2019. [Online; accessed 26 July 2019].

[25] Wikipedia contributors. Ontology — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Ontology&oldid=909848221, 2019. [Online; accessed 23-August-2019].

[26] Wikipedia contributors. Ontology (information science) — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Ontology_(information_science)&oldid=911914334, 2019. [Online; accessed 23-August-2019].

[27] Wu, S., and He, Y. Enriching pre-trained language model with entity information for relation classification. *arXiv preprint arXiv:1905.08284* (2019).

[28] Xiao, M., and Liu, C. Semantic relation classification via hierarchical recurrent neural network with attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers* (2016), pp. 1254–1263.

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

[30] Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., and Xu, B. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (2016), pp. 207–212.