

2022-11-28

Gaining more from Tweets: Knowledge, Actions, and Requirements Elicitation by a Hybrid Method of Natural Language Processing

Masahati, Mohammad Navid

Masahati, M. N. (2022). Gaining more from tweets: knowledge, actions, and requirements elicitation by a hybrid method of natural language processing (Master's thesis, University of Calgary, Calgary, Canada). Retrieved from <https://prism.ucalgary.ca>.

<http://hdl.handle.net/1880/115561>

Downloaded from PRISM Repository, University of Calgary

UNIVERSITY OF CALGARY

Gaining more from Tweets: Knowledge, Actions, and Requirements
Elicitation by a Hybrid Method of Natural Language Processing

by

Mohammad Navid Masahati

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

GRADUATE PROGRAM IN COMPUTER SCIENCE

CALGARY, ALBERTA

NOVEMBER, 2022

© Mohammad Navid Masahati 2022

Abstract

In Decision Support Systems (DSS) one of the most important types of input data for supporting the final decisions is textual data. However, in this era, there are significantly more volumes of textual content generated from various sources than could ever be processed, analyzed, and further used for decision-making. Moreover, most of the textual data sources produce unstructured content with no unified scheme, thus, making it even more difficult to automatically mine them for gold nuggets of information supporting pivotal data-supported decisions. One of the most widely used sources of textual content for mining public opinions and extracting subject-specific requirements is Twitter in which people publish over 500 million tweets on a daily basis. While this makes Twitter a great source of knowledge for public requirements and trends, Tweets are very difficult to be processed for requirements and knowledge elicitation since they are unstructured, written in conversational and imperfect grammar, and often non-informative for supporting decisions without being properly processed and analyzed.

Here in the course of this thesis, a semi-automatic methodology pipeline named DeKoReMi (Deep Knowledge and Requirements Miner) is proposed that employs state-of-the-art Deep Learning and Natural Language Process-

ing techniques. The goal is to elicit the hidden and integral requirements, and more importantly, the necessary related knowledge and description to explain the extracted requirements from extremely large corpses of textual unstructured content (specifically Tweets). The retrieved information will further be used as the basis of pivotal data-supported decisions in a wide variety of Decision Support Systems. In this research, DeKoReMi has been developed and proved to be effective using the "Action Research" methodology over the course of three real-life industrial-academic projects conducted in collaboration with the City of Calgary, and Suncor Energy having processed over 10 million tweets combined.

Acknowledgments

I want to express my deepest appreciation and gratitude to Dr. Guenther Ruhe who had trusted me to become his mentee and provided me with wonderful opportunities to work with the leaders in the industry and excellent guidance to flourish in my graduate studies.

I am honored to have Dr. Raymond Patterson and Dr. Usman Alim as my thesis committee and I want to thank them for taking the time and effort to review my work.

Though the past two years, I cherished every second I've talked with my parents giving me hope and reason to continue. I cannot state how much I've missed them in my heart, especially in this difficult time and situation in my home country.

Lastly, I am thankful for all my friends, here and around the globe who kept me company and made me good times through all the ups and downs of my life.

Table of Contents

Abstract	ii
Acknowledgments	iv
Table of Contents	iv
List of Tables	vi
List of Figures	viii
Glossary	xii
1 Introduction	1
1.1 Introduction and Motivation	1
1.2 Research Questions	6
1.3 Thesis Outline	9
2 Background	16
2.1 Transformers	16
2.2 Text Vectorization	23
2.3 Text Summarization	28
2.4 Summary	30
3 Literature Review	31
3.1 Requirements Engineering and Extraction from Text	32
3.2 Topic Modeling from Social Media	35
3.3 Knowledge Extraction	38
3.4 Conclusions	39
4 Methodology	42

4.1	Overview	42
4.2	Workflow	42
4.3	Data Collection	46
4.4	Eliminating Non-informative Tweets	52
4.5	Text Normalization and Processing	58
4.6	Text Vectorization	59
4.7	Tweet Clustering	62
4.8	Cluster Analysis and Cluster Theme Detection	67
4.9	Analysis and Requirements Extraction	75
5	Empirical Studies	83
5.1	Study Design	85
5.2	DeKoReMi - Proof of Concept Analysis	92
5.3	Association Between Emotional Analysis and Actionability	111
5.4	Comparative Analysis of Auto-assigning Themes to Tweet Clusters	118
6	Discussion and Threats to Validity	134
6.1	Discussion	134
6.2	Threats to Validity	140
7	Conclusions and Future Work	145
7.1	Summary	145
7.2	Future Work	147
	Bibliography	150

List of Tables

3.1	The list of features and values offered by the reviewed studies . . .	40
5.1	Validation of the selected clusters by three domain experts from the city of Calgary.	91
5.2	Initial keywords for gathering the tweets related to the Suncor project.	95
5.3	Search queries generated for fetching related tweets to the sub- ject under study of "digital workplace" for the Suncor project. . .	95
5.4	Validation results for five classification models over training data.	98
5.5	1-word keywords generated by KeyBERT along with their scores for the 11 clusters for the Suncor project	102
5.6	3-word keywords generated by KeyBERT along with their scores for the 11 clusters for the Suncor project	103
5.7	2-word diverse keywords generated by KeyBERT along with their scores for the 11 clusters for the Suncor project	104
5.8	2-word diverse keywords generated by KeyBERT along with their scores for the 11 clusters for the Suncor project	106
5.9	Example of two emotionally opposing tweets within the same topic.	129
5.10	Layer-1 clusters sorted by anger	129
5.11	Layer-1 clusters sorted by fear	130
5.12	Layer-1 clusters sorted by sadness	130
5.13	Top-10 closest tweets to cluster 9 centroid labeled as 'fear'. . . .	133

6.1	The list of features and values offered by the reviewed studies along with DeKoReMi	137
-----	---	-----

List of Figures

1.1	The outline of the experiments and steps done for this thesis and how they connect to each other.	11
1.2	The development process of DeKoReMi using action research over the industrial-academic projects and empirical investigations.	13
2.1	The transformer - model architecture (Vaswani et al. (2017)) . . .	17
2.2	Multi-Head Attention Vaswani et al. (2017)	19
2.3	Pre-Training and Fine-Tuning in BERT Devlin et al. (2018c)	21
3.1	The main components of this research and the overlapping area highlighting the focus of this thesis.	32
4.1	The outline of the experiments and steps done for this thesis and how they connect to each other.	43
4.2	Diagram for the first part (Data Collection) from Figure 4.1	46
4.3	An example of searching the query " <i>(collaboration OR workplace) OR ((remote OR hybrid) AND work) -(software development)</i> " in Twitter advanced search.	53
4.4	Diagram for the second part (Noise Elimination) from Figure 4.1	54
4.5	Example tweet for text normalization	59
4.6	Diagram for the third part (Tweet Clustering) from Figure 4.1 . .	64
4.7	Diagram for the fourth part (Cluster Theme Detection) from Figure 4.1	68
4.8	KeyBERT example taken from https://maartengr.github.io/KeyBERT/#usage	72

4.9	KeyBERT diverse keywords example taken from https://github.com/MaartenGr/KeyBERT	73
4.10	Diagram for the fifth and the last part (Analysis and Requirements Extraction) from Figure 4.1	76
4.11	Sample highlighted keywords by Suncor field experts.	77
4.12	Sample tweets containing different variations of the keywords "collaboration" and "hybrid OR remote work" in the Suncor project 4.11	79
4.13	Sample summary of the tweets containing different variations of the keywords "collaboration" and "hybrid OR remote work" in the Suncor project 4.11	80
5.1	The iterative cycle of action research (Hur et al. (2013))	84
5.2	The development process of DeKoReMi using action research over the industrial-academic projects and empirical investigations.	86
5.3	The COVID-19 new cases rate in Alberta (source: https://health-infobase.canada.ca/covid-19/).	88
5.4	The inertia plot of the sentence embeddings representing the Suncor tweets used for determining the optimal number of clusters for the elbow method.	100
5.5	Sample clustering result for the tweets of the Suncor project.	101
5.6	The tweets in cluster 5 that match the searching the important keywords mentioned in Table 5.8	107
5.7	The summary of the tweets in cluster 5 that match the searching the important keywords mentioned in Table 5.8	108

5.8	The result of applying the Equation 5.1 on the 2300 classified tweets to calculate the probability of tweets being actionable if expressing certain emotions.	115
5.9	The result of applying the Equation 5.2 on the 2300 classified tweets to calculate the Pearson Correlation Coefficient between the tweet emotions and actinability	116
5.10	Final relevance scores for the four proposed auto theme assignment methods representing how close each result to human perception.	122
5.11	Distribution of emotion classes over the training dataset	125
5.12	Distribution of emotion classes over the Parks and Recreation related dataset 5.1	127

Glossary

BERT	BERT stands for Bidirectional Encoder Representations from Transformers which is a transformer based deep learning model for NLP developed by Google.
Fine-tuning an NLP model	Since pre-trained models are trained on general data, they should be trained again on more focused textual data to work efficiently on specific tasks. The post-trained model is called a fine-tuned model and the post-training process is called fine-tuning.
LDA	Latent Dirichlet allocation is the most famous topic modeling statistical model that use word use frequencies to group parts of a text corpus in different topics
Masked Language Models (MLM)	Language Models like BERT that are trained by masking words and predicting the mask replacements
Multi-class Classification	A classification which involves more than two classes (or set of labels) to assign to each unit of data.
Multi-layer Clustering	Performing text clustering multiple times on a text corpus. Treating the resulting clusters as text corpuse and re-clustering them into sub-clusters.

Named Entity Recognition	The task of identifying named entities from text and classifying them into pre-existing named entity categories, such as organizations, person names, locations, number entities like currencies, etc.
POS tagging	Point of Speech tagging is the task of mapping each element in text to a part of speech such as noun, verb, adjective, etc.
Pre-training an NLP model	A pre-trained NLP model is normally a heavy deep model which is initially highly expensive to train to work as expected that is already trained on a huge amount of general textual document by someone with enough computation power.
Random Forest	RF is an ensemble learning classification method that builds multiple decision trees and trains them on a labeled dataset for classification purposes
Short-Text Classification	The act of assigning labels called classes to a series of short texts to distinguish them between each other in a predefined criterion.
Short-Text Clustering	Breaking a series of texts into groups of texts with similar meaning and contexts.
Text Vectorization	Representing text corpus as a series of numbers called vectors.

Transformers	A transformer is a deep learning model used in Natural Language Processing and Computer Vision which focuses on paying attention on specifying a weighing differential for specific parts of the input data (Vaswani et al. (2017)).
--------------	--

Chapter 1

Introduction

1.1 Introduction and Motivation

A Decision Support System (DSS) uses the information retrieved from different types and data sources to support the reasoning behind the decisions made. Textual data is one of the most critical data types explored for information retrieval to support decisions. However, in this era, there is significantly more textual content generated from various sources than could ever be processed, analyzed, and further used for decision-making. Moreover, most of the textual data sources produce unstructured content with no unified scheme, thus, making it even more difficult to automatically mine them for gold nuggets of information supporting pivotal data-supported decisions.

Twitter is one of the most widely used sources of textual content for mining public opinions and extracting subject-specific requirements which people publish over 500 million tweets on a daily basis. This makes Twitter a great source of people's old and recent challenges, demands, and requirements. People tweet freely about all aspects of their daily lives every day,

especially true in rapidly changing situations like COVID-19 that have drastic effects on the public's everyday living needs. Even though what we discussed are the reasons for Twitter being a great source of public requirements about any subject, extracting knowledge from tweets is a difficult task. Because tweets are large volumes of multi-media and textual content that are "public opinions" published by anyone by their nature written in unstructured, conversational, mostly grammatically incorrect, and abbreviated language. Which means processing them for supporting pivotal business and political decisions is extremely challenging and should be done with caution (Qi et al. (2020)).

There have been many studies done on information retrieval on different subjects from social media or other types of textual content which will be further discussed in Chapter 3. Most studies such as Gupta and Gupta (2019), Kengphanphanit and Muenchaisri (2020), and Henriksson and Zdravkovic (2020) use topic modeling (LDA), different classification methods, and sentiment analysis for extracting merely requirements or events. They mostly provide no additional knowledge beneficial to support the decision-making process.

Here we introduce DeKoReMi (Deep Knowledge and Requirements Miner) that deeply mines any given textual content, segregates the dataset into different themes of discussion, extracts the requirements from the segregated themes, and highlights the important parts of the data for each requirement. Moreover, it provides summaries and knowledge explaining the extracted requirements to be able to provide the important gold nuggets of information to support pivotal decisions in any context, such as industry or politics.

The DeKoReMi methodology pipeline is designed based on state-of-the-art Deep Learning and Natural Language Processing techniques. The pipeline first starts with systematically gathering the most related tweets. Then it eliminates the non-informative ones using text classification. The remaining informative tweets will further be clustered into tweet groups with semantically similar tweets. Each group will be used to infer the theme of discussion using auto-text summarization and keyword extraction. The valuable extracted keywords and summaries will be used to be mapped to problem areas and elicit the requirements. Lastly, two objective and subjective analyses will be provided for each requirement providing sufficient knowledge about them to support the decisions addressing the requirements. Additionally, DeKoReMi goes beyond extracting themes of discussion, corresponding requirements, and supporting knowledge by leveraging emotional analysis for prioritizing the extracted requirements. Moreover, it suggests different parts of the text with high potential for triggering action, which are the results of two additional empirical experiments to the pipeline.

DeKoReMi has been developed using the "Action Research" methodology [Reason and Bradbury \(2001\)](#). Action research is a collaborative interaction between performing action and doing research in a feedback loop that uses real-life projects to develop, test, and improve the research methodology interactively. This action research has been developed over the course of three industrial projects with different contexts collaborating with the City of Calgary and Suncor Energy company, having analyzed over 10 million tweets. The first two projects were conducted in collaboration with the city of Calgary. In those projects, we extracted the requirements ex-

pressed by people on Twitter regarding different city services in the city of Calgary. The first project addressed the newly created and dynamically changing requirements due to the "COVID-19" pandemic aiming to help the city funnel its investments into different city services. And the second project aimed to categorize the requirements regarding the city's "parks and recreation" services such as parks, pedestrian, and biking pathways. And lastly, the third project implemented in collaboration with Suncor Energy aims to improve the "digital workplace" of their employees by extracting the expressed digital workplace requirements published during the past two years.

In addition to the three projects, to improve certain sub-steps of the methodology, two empirical studies have been conducted. The first empirical study focuses on finding the closest automatic method to the human perception that assigns themes to tweet clusters, which is used to eliminate human intervention from assigning subjects of discussion to cluster groups that are created using clustering. And the second one investigates the benefits of emotional analysis on requirements prioritization and its correlation with taking action based on tweets, which is applied at the end of the methodology pipeline and aims to go beyond only extracting requirements by suggesting tweets that would trigger action.

The benefit of DeKoReMi in the three-stage decision-making process (which are namely, intelligence, design, and choice) ([Pomerol and Adam \(2004\)](#)) flourishes in the first two stages being intelligence and design. The "intelligence" stage employs textual analysis to represent the cognitive aspects and the situation of the decisions to be made, which could be either in enhancing the existing decision scenarios or formulating and understand-

ing new decision areas. DeKoReMi is an extreme help in the "intelligence" stage, especially in extracting and explaining the new decision problems using new discussion-theme and requirements extraction from textual data. And the "design" stage is the synthesis and consolidation of the related knowledge elicited around the decision problems defined and explained in the "intelligence" stage. DeKoReMi also plays an integral role in this decision-making stage by providing supporting knowledge regarding the extracted requirements and problem scenarios.

The main contributions of this thesis are as follows:

- Designing and implementing a comprehensive methodology pipeline called DeKoReMi that extracts requirements and their supporting knowledge from large volumes of unstructured textual data.
- Forming a completely automatic method to extract different subjects and themes of discussion from tweets and map all of the previous and future tweets to the extracted themes of discussion.
- Performing three industrial-academic projects along with two empirical studies to develop and evaluate the proposed methodology using "Action Research".
- Going beyond information extraction to merely support the decisions by helping during the decision-making process by studying the correlation between emotional analysis and prioritizing requirements and taking action.
- Investigating closest automatic methods to human perception in assigning themes of discussion to tweet clusters to eliminate human in-

tervention from the requirements and knowledge extraction pipeline. The investigated methods are, namely selecting the top tweets of the clusters, auto-cluster summarization, and keywords extraction using two methods, TD-IDF and KeyBERT (as described in Section 4.9).

1.2 Research Questions

In this section, the main research questions of this thesis are listed. For each research question, the motivation behind asking the question (why?) and the method used to address the research question (how) is also discussed.

RQ1: How effective does the DeKoReMi methodology perform in an industrial setup, how helpful is the resulting outcome, and what are the main perceived benefits by the field experts?

(Why?) As this research was conducted using "Action Research" during three industrial projects, the resulting methodology pipeline is aimed to be applicable in real-life Decision Support Systems in the industry, academia, and politics or any other environment where making decisions requires analyzing textual content to support the decisions made. This research question is formed to assess how effective the DeKoReMi methodology performs in an industrial setup and how helpful is the resulting outcome of the methodology to the domain experts for supporting their decisions to be made in the industry.

(How?) To answer this research question, we apply every step of the DeKoReMi methodology pipeline in an industrial setup in the "Suncor Energy" company. We discuss each step, how they are done, and what are their inputs and outputs in a complete real-life implementation of the method-

ology. Finally, as the implementation of the methodology was done in interaction with the domain experts from Suncor Energy, their weekly and final feedback, along with their usage of the methodology and the impact of the output, will be provided in this research as a qualitative answer to this research question. **RQ1** is addressed in Section 5.2.

RQ2: How effective is DeKoReMi in extracting themes of discussion and assigning tweets to them?

(Why?) One of the main parts and contributions of this research is segregating and extracting the different themes and subjects from the text. The different themes are represented in the form of tweet clusters which will further be converted to the requirements. This research question is formed to evaluate the accuracy of the proposed method in clustering tweets and grouping semantically similar tweets together.

(How?) We address this question during the empirical project conducted in collaboration with the City of Calgary. The context is the related tweets to "parks and recreation" city services, such as the city's natural parks and pedestrian and biking pathways. After creating clusters of tweets and assigning themes of discussion to them during the implementation of the methodology, we evaluate the quality of the tweet clusters in terms of how close are the assigned subjects to human perception and what portion of the tweets are semantically similar to each other within clusters. The evaluation was done manually by the domain experts from the city of Calgary, and the evaluation process is addressed in Section 5.1.

RQ3: Does emotional analysis of the tweets help prioritize the tweet clusters in terms of importance and urgency?

(Why?) The DeKoReMi methodology pipeline not only extracts the

themes of discussion, the requirements, and the knowledge hidden in textual content but also helps the decision-makers prioritize the extracted requirements using emotional analysis of the tweets and clusters. By forming this research question, we examine whether emotional analysis could help the domain experts prioritize the requirements and the themes of discussions to be addressed. And if the emotional analysis is beneficial, how does it provide more information regarding the requirements and their priorities.

(How?) We address this research question by performing emotional analysis on the tweets and tweet clusters and investigating the semantic correlation between emotions expressed in tweets and how important is the addressed topic and whether emotion could be an indicator of the priority in the extracted requirements. This research question is discussed in Section 5.4.

RQ4: How well emotional analysis helps detect clusters and tweets that result in taking action?

(Why and How?) After prioritizing the extracted requirements in **RQ3** using emotional analysis, we further aim to detect the parts of the text that potentially triggers an action to further help the decision makers where to focus in their data before taking action. In this research question, we investigate the correlation between tweets expressing emotions and whether there could be actions taken based on tweets with different emotions. This research question triggers an empirical study to investigate the correlation between tweet emotion and actionability which is addressed in Section 5.3.

RQ5: What is the closest automatic method to human perception in assigning themes of discussions to tweet clusters in DeKoReMi?

(Why?) One of the goals of the DeKoReMi methodology pipeline is to minimize human intervention as much as possible during the Decision Support process. However, as the task of clustering text falls under the unsupervised machine learning tasks, the subjects of the resulting clusters are initially unknown and need to be semantically examined to be assigned a theme or subject of discussion. As understanding the semantic meaning of a text can only be perceived by humans, which is extremely time-consuming, we proposed four different automatic methods in Section 4.9 to declare the overall topic of a cluster using language models, eliminating the necessity of human intervention. However, not all methods will result in the same themes of discussion. This research question is formed to investigate the closest automatic theme assignment method to human perception.

(How?) To answer this research question, we conduct an empirical experiment in an academic setup during the "Suncor Energy" industrial project (discussed in Section 5.2) to manually investigate the closeness of each proposed automatic theme assignment method to human perception. The answer to this research question, the experiment design, and the experiment results are explained in Section 5.4.

1.3 Thesis Outline

In this section, first, we describe the overview and outline of all the different experiments and steps and how they are connected and complete one another. And then, we describe the structure of this thesis and its chapters to guide you through what questions each section is trying to answer.

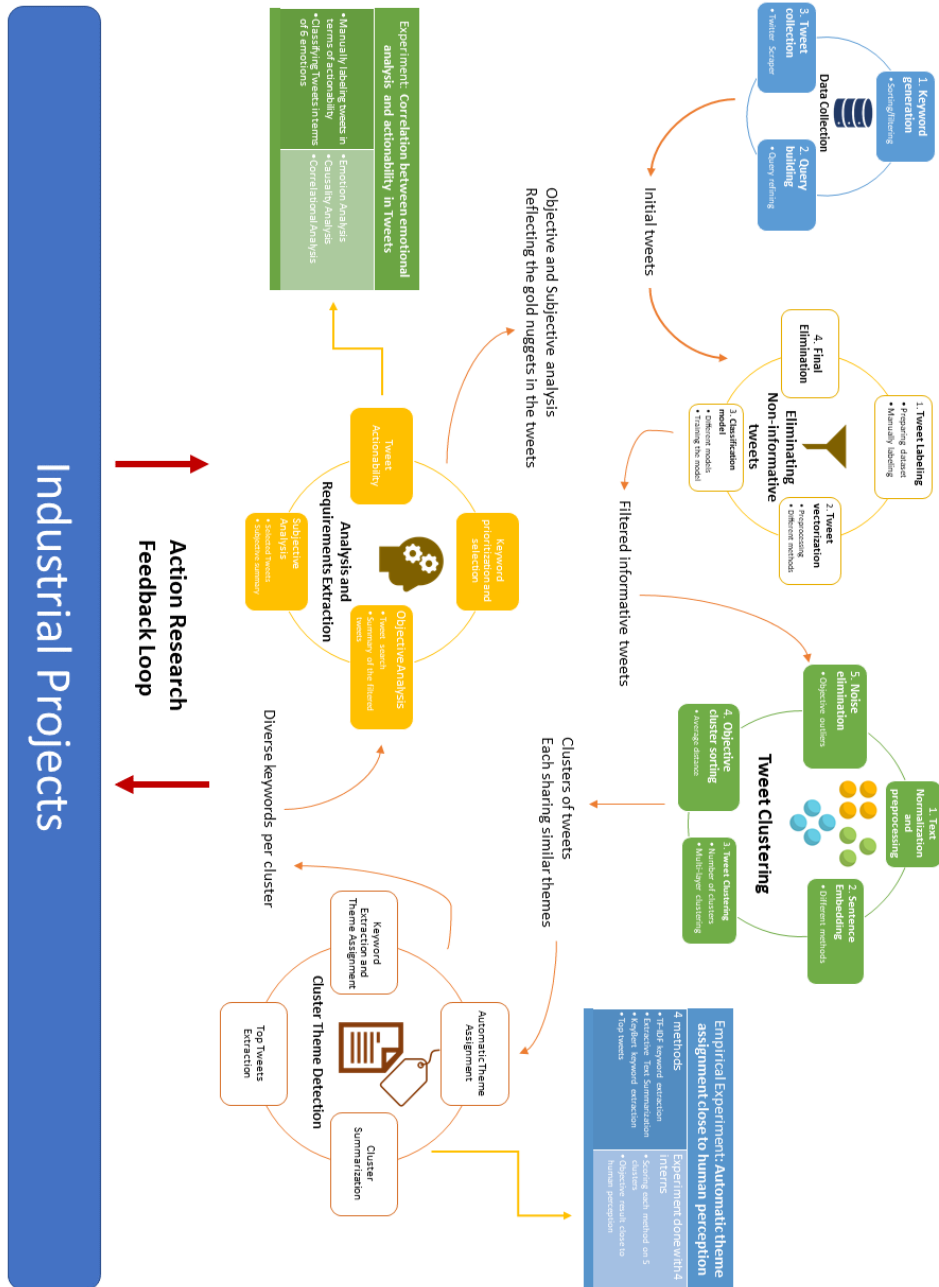
There are multiple research steps, experiments, and studies done in

the course of this thesis. Some of them might seem not related to others, but all of them are designed and performed moving towards the same goal, which is better extracting gold nuggets of information used for decision-making from textual content (we define the "gold nuggets of information" as the most important and valuable pieces of information hidden in massive datasets. These pieces of information are the assets that bring value to the whole dataset). Figure 1.1 demonstrates an overview of the different parts of this thesis and how they are connected, what are the inputs and outputs of each section, and how they gradually merge towards the same goal.

The different sections visualized in Figure 1.1 can be broken into three main parts. First, there is the main pipeline of the thesis, which creates the DeKoReMi that is responsible for extracting gold nuggets of information from large amounts of mostly non-informative content. The steps of the DeKoReMi consist of Data Collection, Noise Elimination, Tweet Clustering, Cluster-Theme Detection, and lastly, Analysis and Requirements Extraction. The second parts are the intra-results which are then results and the outputs of each section and the inputs of the next sections, which are the initial tweets, filtered informative tweets, clusters of tweets, keywords describing the theme of each cluster, and lastly, the objective and subjective analysis reflecting the requirements and the most important parts of the initial dataset that will be used for decision making. The process of how each step of DeKoReMi works and how they generate each result, and how each consumes the previous result to process and prepare the next section is described in Chapter 4 in detail.

And lastly, there are two experiments that contribute to information retrieval and extracting actions from tweets automatically. The first ex-

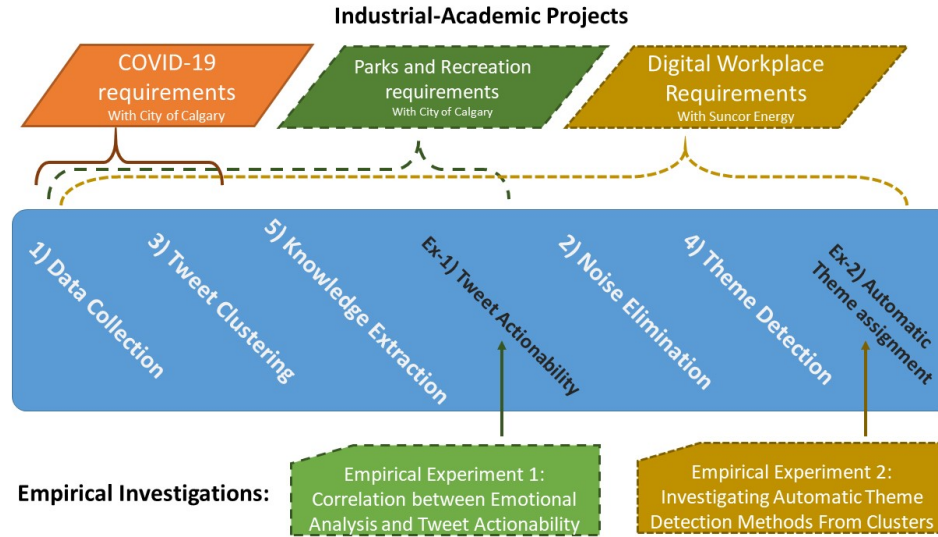
Figure 1.1: The outline of the experiments and steps done for this thesis and how they connect to each other.



periment is "automatic theme assignment to each cluster close to human perception". And the second empirical study is the study of the correlation between tweets' attributes and their actionability (which is the potential of being a source in deciding to form one or more actions). These two empirical studies are deeply connected in the sub-process of the corresponding step, which is the "cluster theme detection" and "analysis and requirements extraction" steps, respectively. The results of the first experiment are used in the "cluster-theme detection" step to automatically assign keywords and themes to each cluster that are close to human perception. And the results of the second experiment is used in the final step to automate the process of detecting the actionable tweets in each theme of discussion.

As illustrated in Figure 1.1, the development of the DeKoReMi methodology pipeline has been done using the "Action Research" Methodology throughout three industrial-academic projects collaborating with the city of Calgary and Suncor Energy. Which means there is a simultaneous process of developing the methodology (conducting research) and performing action (implementing in the industrial projects), which reflect each other in an interactive feedback loop. Figure 1.2 shows how each project contributed to the development of each step and empirical experiment in this thesis. As a note, the order of the steps in Figure 1.2 is not necessarily ascendingly because the development of the steps through the implementation of the projects was not in the drafted order in the methodology chapter 4 and the chronological development of the steps are as depicted in Figure 1.2 from left to right. The industrial projects and empirical experiments mentioned in Figure 1.2 are fully described in Chapter 5.

Figure 1.2: The development process of DeKoReMi using action research over the industrial-academic projects and empirical investigations.



Structure of the Thesis

This chapter is delivered in 7 chapters and the following gives a general description of what could be expected to understand from reading the rest of the thesis.

Chapter 2 (Background) provides the background knowledge required to follow up with the methodology. This chapter also describes the tools and techniques used to base and develop the methodology pipeline proposed in this thesis. Each component described in the background is used in different parts of the methodology and is very important to understand in order to justify the usage in the methodology pipeline.

Chapter 3 (Literature Review) is a detailed review of the studies conducted related to the proposed methodology in this thesis. Although, since the methodology consists of many different elements and NLP techniques, no single research could be a good fit for being fully related to the material

delivered in this thesis. Therefore, the reviewed literature in Chapter 3 has been classified into three different classes, namely "Requirements Engineering and Extraction from the text", "Topic Modeling from Social Media", and "Knowledge Extraction". Each is related to different sub-sections of DeKoReMi.

In Chapter 4 (Methodology) we cover the main body of the proposed DeKoReMi methodology pipeline in detail. The methodology is explained in detailed step-by-step guidelines on how to replicate the methodology and analyze the results, along with references to the tools used. An outline of the methodology is illustrated in Figure 4.1 and each element in the Figure is explained in its own sub-section.

Chapter 5 (Empirical Studies) first explains how the proposed DeKoReMi methodology pipeline is developed using "Action Research" (as depicted in Figure 5.2) over the course of three real-life industrial projects while explaining each projects, their contexts, and their results. The chapter also delivers the detailed outcome of implementing the methodology in the industrial setups along with the feedback received from the domain experts. And then, the extra two empirical experiments conducted complementary to the proposed methodology are described along with their goals and results. This chapter contains most of the answers to the defined **RQs** in Section 1.2.

In Chapter 6 (Discussion and threats to validity), we compare the proposed methodology with the existing and reviewed literature in Chapter 3 and discuss how DeKoReMi contributes to the body of knowledge in requirements and knowledge extraction and bringing value to the Decision Support Systems in the industry and academia. Then, we mentioned the

identified possible threats to the validity of the research and the results and our efforts to mitigate the issues.

Lastly, Chapter 7 (Conclusions and Future Works) delivers a summary of the thesis and discusses the possible future works and ideas on how to continue the research path conducted in this thesis.

Chapter 2

Background

Throughout this research, we have used many state-of-the-art technologies and concepts for processing textual data and extracting data-supported decisions. Some of these tasks are text classification, clustering, representation, and summarization. In this chapter, we introduce and describe the tools and concepts used to perform the aforementioned text-processing tasks in detail, which is integral for understanding the underlying methodology designed in this study.

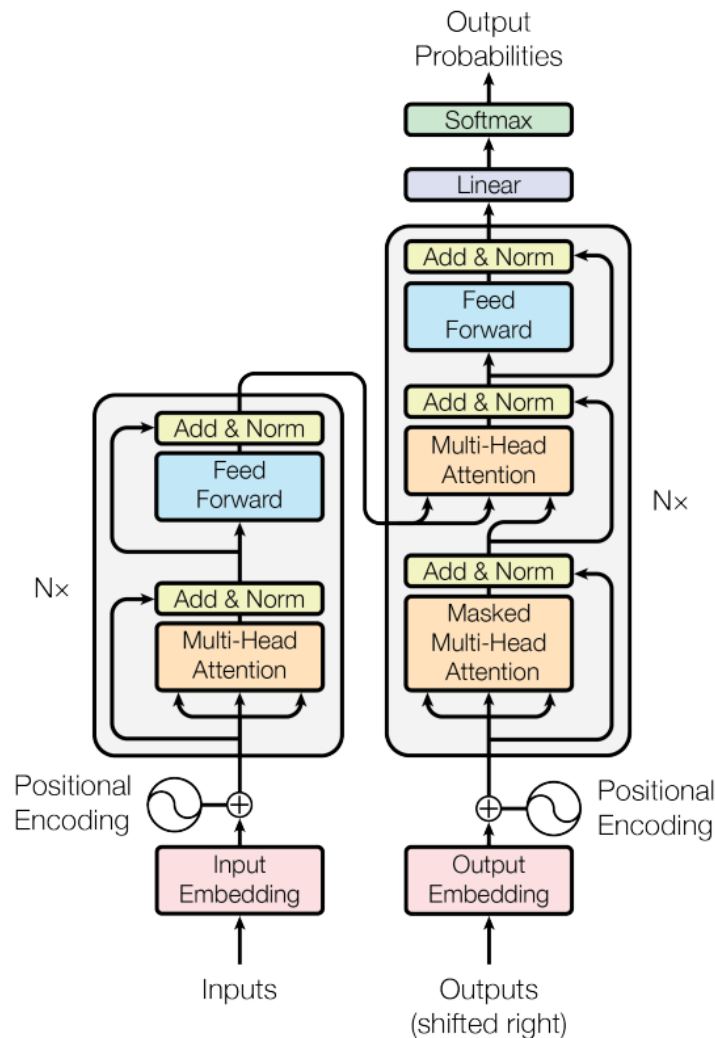
2.1 Transformers

Transformer is a novel neural network model proposed and developed by Google in 2017. Transformers are encoder-decoder sequential neural networks that combine a series of recurrent neural networks (RNNs) to employ the attention mechanism on the input sequence to assign different weights of attention and importance to different parts of the input. Transformers are highly effective in tasks that require deeply interrelated sequence analysis,

such as Neural Machine Translation (NMT), Named Entity Recognition, and emotion and sentiment analysis in natural language [Vaswani et al. \(2017\)](#). Unlike pure RNNs where the input should be passed to the model in sequence, in Transformers, the input sequence should be passed (and further processed) simultaneously. So unlike most RNNs, Transformers could be trained much faster in parallel by GPU cores.

The architecture of Transformers is depicted in Figure 2.1.

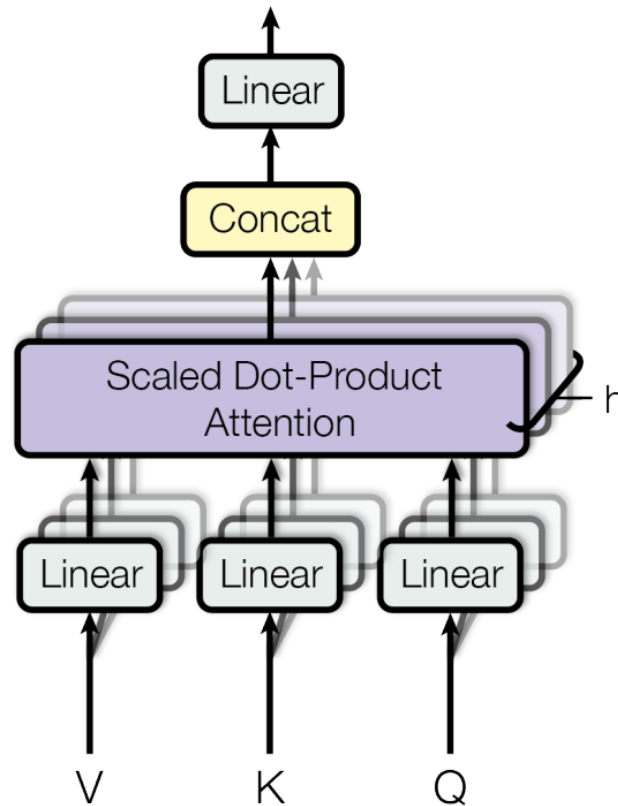
Figure 2.1: The transformer - model architecture ([Vaswani et al. \(2017\)](#))



As could be seen in Figure 2.1, the Transformer architecture consists of an encoder (the left half) and a decoder (the right half).

The encoder block (the left side of Figure 2.1), receives the input sequence of words and groups similar words close to each other in an open space called embedding space. In return, the encoder block converts the given words to a multi-dimensional vector space, representing the input text as numbers. The resulting vectors also convey the context of the word(s) corresponding to the surrounding words. Since the idea behind Transformers is based on "attention", the Multi-Head Attention part in the Encoding block plays an integral role. The Multi-Head Attention section generates the attention vector by focusing on how related each word is to in the context of the sentence compared to its surrounding words in that sentence [Vaswani et al. \(2017\)](#). As depicted in Figure 2.2, the Multi-Head Attention part calculates all the attention weights for all the words per every word in a sentence based on the importance of the role it plays in the sentence and then concatenates the attention vectors and returns the final 3-dimensional vector.

The responsibility of the Decoder block (the right side of Figure 2.1) is to generate the target sequence correlated to the output of the Encoder block considering the previous training. Assuming that the task is to train the Transformer to translate a sentence from English to German, the input English sequence is passed to the Encoder, and the resulting German sequence is passed to the Decoder to be trained. Each word in the output sequence of the Decoder block is first masked in the Masked Multi-Head attention section, and the probability of the next word is estimated to replace the masked word. Therefore the Decoder block is trained and can further gener-

Figure 2.2: Multi-Head Attention [Vaswani et al. \(2017\)](#)

ate the output sequence in the target language by masking and calculating the probability of the word in each position of the resulting sequence.

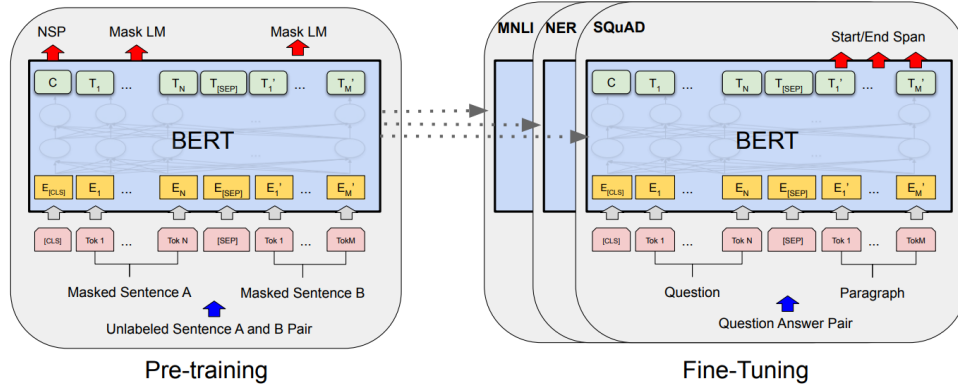
BERT

BERT (Bidirectional Encoder Representations from Transformers) ([Devlin et al. \(2018c\)](#)) is one of the most important forms of Transformers 2.1 and has been used in a wide variety of NLP tasks in recent years both in academic studies and in the industry. Today, almost every Google search query makes use of BERT to better understand the context of the search query and provide the most related results [Google](#). Previous Language models like Generative Pre-Training (GPT) ([Radford et al. \(2018\)](#)) were unidirectional. Meaning

that they would learn, process, and understand the language sequentially in one direction, either from left to right or from right to left, and each word can only focus on its previous words. The main constraint of the unidirectional models is that they cannot be designed in an architecture in which pre-training is possible.

The key innovation of BERT is training Transformers bidirectionally, which is in contrast with the previous language models that process the text sequentially, as mentioned before. The paper shows that bidirectional training of Transformers will result in a much deeper understanding of the context of natural language text as opposed to unidirectional training. BERT alleviates the aforementioned constraint of unidirectional models by applying a "masked language model (MLM) pre-training" objective ([Devlin et al. \(2018c\)](#)). Another key difference between BERT and other language models is that BERT consumes and analyses the input text all at once instead of reading them sequentially. This way of consuming the input allows BERT to understand the context of each word based on their surrounding words, which is very important because the same words might convey different meanings depending on the surrounding words. Also, passing all the words all at once allows the model to be trained and executed in parallel using GPUs which makes the training much faster.

In BERT's framework, there are two main steps, namely pre-training and fine-tuning (depicted in [Figure 2.3](#)). In the pre-training phase, the model is pre-trained by a series of unlabeled textual data to understand the concepts, contexts, and meanings of different parts of natural language. Some of the commonly used text corpora include BooksCorpus (800M words) ([Zhu et al. \(2015\)](#)) and English Wikipedia (2,500M words). Pre-training is done using

Figure 2.3: Pre-Training and Fine-Tuning in BERT [Devlin et al. \(2018c\)](#)

two tasks, namely Masked LM (MLM) and Next Sentence Prediction (NSP). In the former method, the model masks words and tries to fill in the blanks, and learns the correlation of words with their surrounding words. The paper mentions that they mask 15% of the words to pre-train the model. However, the language model should also learn the contexts of sentences and their relations to each other as well to be able to perform inter-sentence related tasks such as Question Answering. Which is why the model is also trained by the Next Sentence Prediction method. In this task, BERT tries to guess whether a certain sentence can follow another sentence and learn.

Despite the pre-training phase, which was designed for training the model for general understanding and usage of natural language, the fine-tuning phase is designed to train the model for task-specific jobs, such as text classification and question answering (sentence pairing). For each specific task, we should only pass the corresponding input and output, and all the parameters will be updated end-to-end. Despite pre-training, the fine-tuning phase is relatively computationally inexpensive ([Devlin et al. \(2018c\)](#)).

Text Classification with BERT

Text classification has a wide variety of applications, and there have been many studies conducted and many tools developed to enhance the accuracy of classifying text into different groups. Some of the important applications include email or SMS spam detection, sentiment analysis, and problem area mapping.

As described in Section 2.1, BERT can be fine-tuned for specific tasks, one of which is text classification. The task of text classification is to assign pre-defined categories to a series of texts. Which is normally done by training a machine-learning language model over a dataset of labeled text. BERT has shown significant results in common natural language understanding (NLU) tasks. Since BERT language models are a form of transfer learning and are already pre-trained, they have a good understanding of language and only need to be further fine-tuned to classify text with a high accuracy [Sun et al. \(2019\)](#).

There are many BERT models with different sizes and structures, and purposes already pre-trained by different organizations with enough resources (such as Google) that is available online for public use on HuggingFace ¹, which is a public library or repository for machine learning models. One of the most used models pre-trained by Google is BERT-base ([Devlin et al. \(2018a\)](#)). BERT-base has an encoder with 12 Transformer blocks, 12 self-attention heads, and the hidden size of 768. Which means that it generates 768 features (or a vector with a size of 768) per word embedding. We can pass an input sequence of 512 blocks at most it will return the numerical

¹<https://huggingface.co/>

representation of the input sequence.

BERT-base also adds two more tokens to the input sequence and generates their corresponding representations. The first one being the [CLS] token, is added to the beginning of every sentence and represents special classification embedding representation of the whole sentence, and the second one being the [SEP], which is added at the end of every sentence and is used for segment separation. For classification, BERT uses the vector representation of the whole sentence, which is the final hidden states of the [CLS] token. In the fine-tuning state, BERT tries to learn the mappings between the hidden state of the [CLS] token and the final label, and in the end, it applies the softmax classifier on top of its predictions to estimate the label \mathbf{c} for the hidden state \mathbf{h} [Sun et al. \(2019\)](#).

$$p(\mathbf{c}|\mathbf{h}) = \text{softmax}(W\mathbf{h})$$

In which W is the task-specific matrix representation.

2.2 Text Vectorization

Similar to the existing language barrier between two people communicating in the same but non-native language (for example, in English) different from their native languages, there is a language barrier when trying to make a computer model understand human's natural language. Because the computer cannot understand words and only consumes and returns ones and zeros and everything is converted into ones and zeros before being passed to the CPU. So in order for a computer model to be able to work with natural language, we must first convert the natural language into numbers

representing the original text before passing it to computer models for analysis. **The act and process of representing natural language in terms of numbers is called "Text Vectorization".**

Although natural language is very complicated and, in some cases, cannot even be fully understood by humans. Natural language concepts like sarcasm, metaphors, idioms, Euphemism and similar concepts make it even more difficult to interpret human language. Often words and sentences have more complex meanings beyond their actual translations when they are being used in combination with other words, sentences, or even passages. This is why it is essential to pay attention to these delicacies in natural language when developing a text vectorization and representation method.

There are several popular text vectorization techniques, each having its own benefits and detriments. Depending on the NLP task, different text vectorization techniques could be suggested that *might* outperform others in that specific task. As mentioned before, a text vectorization technique is more effective when it conveys the most information, meaning, and contextual data in terms of numbers. Deeper and more complex models have a deeper understanding of the natural language; however, they might be more computationally expensive to work with, which could be one of their detriments.

Here are some of the most widely used Text Vectorization techniques, along with their strengths and weaknesses:

Text Vectorization with BERT

As mentioned in Section 2.1, BERT is a state-of-the-art natural language model pre-trained on massive textual content which has multiple use cases, such as word embedding, sentence embedding, text classification, and guessing missing words in sentences. Here in this section, we are going to explain the process and power of the word and sentence embedding using BERT. Word embedding is the process of converting each word into a vector representing both meaning and the context of that word depending on the surrounding words in the given text using a pre-trained BERT model. This means that word embeddings have context, and the same words used in different sentences could result in different vector representations (or word embeddings) depending on the words that come before and after that word in the given sentence. Sentence embedding means transforming sentences into vectors representing the content and the overall meaning of the sentences, which could either be done using the word embeddings generated using BERT as mentioned before or using a pre-trained Sentence-BERT model [Reimers and Gurevych \(2019\)](#). The difference between different methods of sentence embedding is explained in detail in Sub-section 2.2. But here we are going to explain the fundamentals of word embedding using BERT.

Before feeding the input text into a BERT model to generate word embeddings, the input needs to be tokenized into a special type of tokens that BERT requires. Tokenization is the process of mapping each word or meaningful part to an ID representing that word. The tokenization process in BERT has two main steps. First, a BERT tokenizer adds two special tokens, one at the

beginning of the text, and one for separating sentences from each other. The former is "[CLS]" which is the representation of the whole text (or line of text) and is usually used in classification tasks. The latter or the separation token is "[SEP]" which is the separator of each sentence and comes in between sentences. For example, take the following sentence as an example: *"This is a lovely day. The sun shines the brightest today."*. This sentence will be converted to the following text before mapping IDs to each word: *"[CLS] This is a lovely day. [SEP] The sun shines the brightest today."*. The second step of tokenization is to break the text into a list of meaningful parts. The result of this step is to form the following list: *"['[CLS]', 'This', 'is', 'a', 'love', '##ly', 'day', ':', '[SEP]', 'The', 'sun', 'shine', '##s', 'the', 'bright', '##est', 'today', '']"*. Notice how a compound word like "lovely" is broken into two smaller sub-words, 'love' and '##ly'. And the 'ly' suffix, which is responsible for creating adjectives, is represented as '##ly' to denote that this token is a part of another word and does not mean anything when used alone. After tokenization, each token is mapped into an ID representing that token. So the former example will be presented as the following list: *"[101, 3844, 5968, 8576, 11350, 9800, 798, 1012, 102, 1996, 2344, 6800, 1200, 1996, 6898, 12309, 6489, 1012]"*. This list is the input that we feed into BERT, which will further be converted into word embeddings. The result for this example will contain 18 embeddings (one per each token), each containing 768 dimensions² which are the final word embeddings (or the final vectorized text using BERT).

The result has many use cases, such as using the embedding of the *[CLS]* tokens for classification or using all the word embeddings to convert them

²This is if we use the 'best-base-uncased' pre-trained model [Devlin et al. \(2018b\)](#). The number of dimensions may differ based on the model used.

to sentence embeddings for semantic search in the text. The importance of the resulting word embedding is that they contain the context in which each word is used because they are generated regarding the surrounding words and sentences using a model that is pre-trained on massive textual content that knows and represents natural language much deeper than statistical models in semantic tasks.

TF-IDF

Another statistic-based method to vectorize textual documents is term frequency-inverse document frequency (TF-IDF) (Jones (1972)). TF-IDF is a statistical method and a weighting scheme that tries to assign weights to each word in a document within a text corpus representing their importance in defining the text corpus. The output of TF-IDF is a vector of size N , which represents the number of unique words in the text corpus. Each number in the resulting vector is the result of the TF-IDF equation (2.3), statistically representing how each word is important to the whole document. TF-IDF comprises of two sections, namely term frequency (TF) and inverse document frequency (IDF). The TF of the term "t" in the document "d" with N_d total words is calculated using the following equation:

$$tf(t, d) = \frac{count(t, d)}{N_d} \quad (2.1)$$

In which $count(t, d)$ is the number of times the term "t" occurs in the document "d". And the IDF of the term "t" in the document "d" with N_d total words is calculated using the following equation:

$$idf(t) = \log\left(\frac{N_{docs}}{df(t) + 1}\right) \quad (2.2)$$

In which $df(t)$ or document frequency is the number of documents that contain the term "t", and N_{docs} is the number of documents in the corpus. And finally, the TF-IDF is the product of TF and IDF as described in the following equation:

$$tfidf(t, d) = tf(t, d) \times idf(t) \quad (2.3)$$

TF-IDF does not take the positioning of the words with respect to each other into account, and therefore the outcome of using TF-IDF for semantic studies is lower compared to other techniques like LSI or multi-word (Zhang et al. (2011)). But the word weighing system that the method is based upon makes the resulting vector well aware of the essential words in the document compared to all the words in the text corpus, which makes this method a powerful statistical vectorization technique for information retrieval and text classification (Zhang et al. (2011)).

2.3 Text Summarization

Text summarization is the task of converting verbose text to a shorter version such that it preserves the most important information and the underlying messages that the original text was conveying. There are multiple applications for automatic text summarization in NLP, such as question answering, legal or news document summarization, information retrieval, and content generation. Overall, there are two main approaches for automatic text summarization using machine learning, namely extractive text summarization and abstractive text summarization.

Extractive text summarization identifies the important sections of the text using a scoring function to form a coherent summary. Then it crops and picks up those portions of the text and attaches them to form the extractive summary of the text. It means that it extracts the summary from the content itself. Extractive text summarizations are relatively easier and faster to use, and they extract the key parts of the text automatically.

Whereas abstractive summarization methods try to interpret the whole content and generate a new shorter text using advanced deep-learning models that semantically and contextually convey the key meanings of the original text. The new more concise text does not necessarily contain any parts of the original text and is similar to an abstractive and subjective summary generated by a human after reading the original content.

Here we describe a powerful extractive summarization technique that we used throughout this research.

Extractive Text Summarization using BERT

As described in Section 2.1, BERT is a pre-trained transformer that understands natural language by training over a massive amount of textual content and could be used in many NLP tasks that we have already talked about a few of them in previous sections. One of the other applications of BERT is extractive text summarization. [Allahtari et al. \(2017\)](#) has developed a method that uses BERT text vectorization (explained in Section 2.2) to generate sentence embeddings and then K-Means to select the closest sentences to the cluster's centroid to extract as the key parts and sentences representing the important portions of the lecture (or textual content) to form the extractive

summary³. The result of the mentioned tool is powerful for extracting the key sentences of a long text to demonstrate the important information in a shorter text.

2.4 Summary

This research is based on many state-of-the-art Natural Language Processing (NLP) and Machine Learning (ML) technologies and is trying to move the body of knowledge and science forward. And it is crucial to understand the underlying technologies on which this research is based. In this chapter, we explained the main concepts and technologies we use in this research in detail, all of which were designed by other engineers and scientists before this research. Throughout the pipeline of this research, we use many NLP and ML tasks such as text classification, text vectorization, unsupervised learning, clustering, information retrieval from text, and automatic text summarization. For each task, we have described their definitions and the latest technologies used to perform those tasks and compared them with each other. From now on, we will refer to the sections documented in this chapter to prevent re-explaining the base technologies to perform the required task.

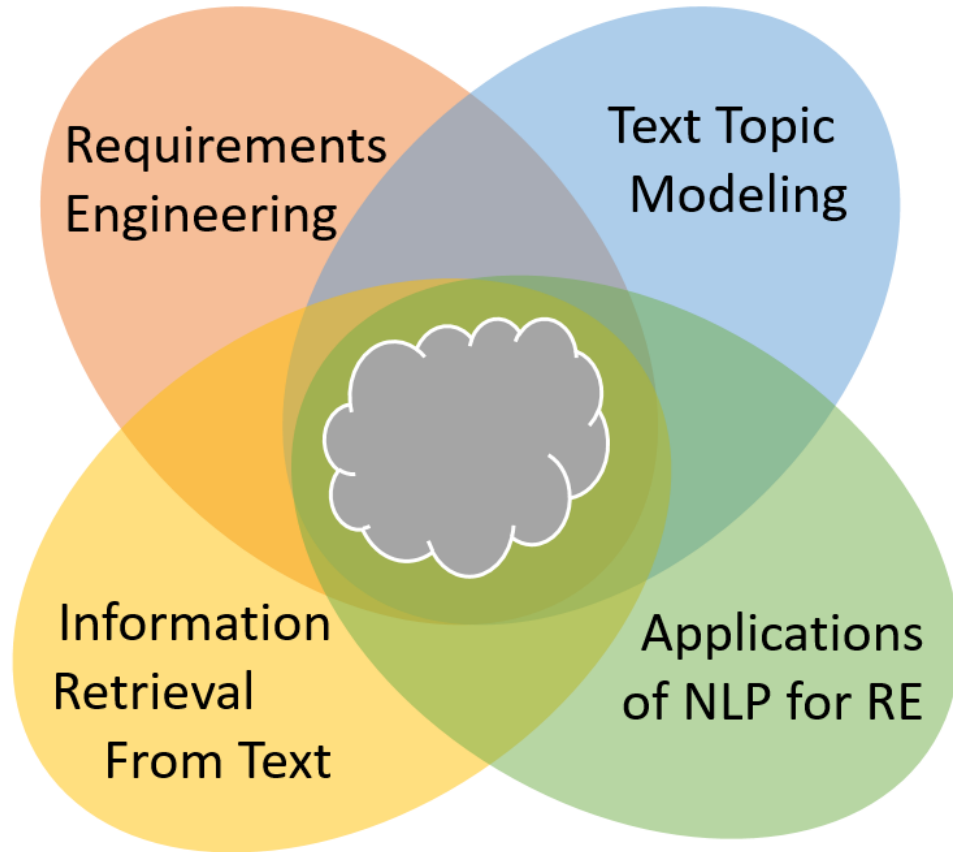
³<https://github.com/dmmiller612/bert-extractive-summarizer>

Chapter 3

Literature Review

In this chapter, we provide a review of the research and experiments conducted related to the DeKoReMi methodology pipeline developed throughout this thesis. Since the outcome of this study is a methodology pipeline consisting of different steps for extracting requirements, segregating themes of discussion, and providing knowledge around the retrieved information, it touches multiple research criteria. Overall, this study is a combination of "requirements engineering (RE)", "information retrieval from text", and "topic modeling in social media". Also, the methodology pipeline consists of different use techniques to deliver the purpose, such as text classification, text clustering, text summarization, and keyword extraction, which classify under "the applications of NLP in RE". Here we go over the studies that address these research areas to deliver similar results and will further compare the reviewed literature with the developed methodology and the conducted experiments in Section 6.1. Figure 3.1 visualizes the main components of involved in this thesis. In which the gray area in the middle shows the overlap of the four research areas addressed in this study.

Figure 3.1: The main components of this research and the overlapping area highlighting the focus of this thesis.



3.1 Requirements Engineering and Extraction from Text

As there is an abundance of textual data withholding important information that if extracted properly, could be very valuable for many purposes in both industry and academia, there have been numerous efforts to develop a method to extract important information from textual content. The studies are mainly different in two areas. Firstly there are different methods, technologies, and tools used to provide similar results, which is extracting

requirements. And secondly, there are different fields, contexts, and textual data sources from which the studies aim to extract the requirements.

As mentioned, the first main area of difference between similar research in requirements extraction is the methods and techniques used for analysis. Text classification is a simple and widely used approach for classifying and, thus, extracting requirements. [Haque et al. \(2019\)](#) uses text classification as the primary method to classify non-functional requirements. They conducted an empirical study using the PROMISE empirical study dataset [pro \(2015\)](#) to train and evaluate different text classification techniques to improve the classification accuracy of the non-functional software requirements. They use a combination of basic statistic text vectorization methods such as TF-IDF (2.2) and Bag of Words (BoW) with seven different classification models (such as Multinomial Naive Bayes (MNB), KNearest Neighbors (KNN), Support Vector Machines (SVM)) to build the classification models under test and found out that the combination of TF-IDF feature extraction (text vectorization) and Stochastic Gradient Descent SVM (SGD SVM) will result in the highest average accuracy, precision, and F1 score. Similarly, [Kumar et al. \(2022\)](#) uses a supervised classification method by training classification models (such as Random Forest and Linear SVM) over a tera-PROMISE data package (similar to [pro \(2015\)](#)) containing 625 requirements categorized as functional and non-functional to map non-functional requirements to pre-existing categories. [Ye et al. \(2020\)](#) employs the same text classification ideology as [Haque et al. \(2019\)](#) and [Kumar et al. \(2022\)](#) but developed and used a much more advanced zero-shot text classification technique via Reinforcement self-learning that uses existing classes to develop into extracting new classes. There are more sophisticated methods of

requirements modeling and extraction other than mere text classification methods. [Haris et al. \(2020\)](#) use sequence Point of Speech (POS) tagging (assigning roles like VERB and ADJ to words in sentences) to build a framework that automatically detects sentence patterns for feature requests among Software Requirement Specification (SRS) documents. Which means this research focuses on the sentences' structure rather than the actual meaning of words (as used in the formerly explained text classification methods) to develop rules that automatically detect feature requests (requirements). To combine both ideas [Jafari et al. \(2021\)](#) and [Hassan and Le \(2020\)](#) did very similar research integrating the classification ideology with sentence pattern analysis to develop a rule-based classification model. They used labeled statements within construction contracts that demand requirements (such as "shall be reviewed" or "the contractor shall submit") to develop two classification models, namely rule-based (with an ideology similar to [Haris et al. \(2020\)](#) but combined with classification) and machine learning based (similar to [Haque et al. \(2019\)](#)) to compare the two methods for requirement statement labeling in construction contracts. [de Araújo and Marcacini \(2021\)](#) defines "the extraction of software requirements from app reviews as a token classification problem" or classifying different words in terms of **B**eginning, **I**nside, and **O**utside tokens or BIO format. They use a more recent and advanced technology (BERT) to develop a token classification method called RE-BERT and fine-tune it over app reviews to classify and extract the tokens which represent the different parts of the software requirements.

The researches are also different in the aiming context for requirements extraction. Most of the previously mentioned research, such as [Haque](#)

et al. (2019), Kumar et al. (2022), and Haris et al. (2020) use software-related requirements like non-functional software requirements and SRS documents. Maalej et al. (2019) also uses software-user feedback classification for Data-Driven Requirements Engineering on software products. Other requirements for engineering-related data type is static data sources. Such as Jafari et al. (2021) that use construction contracts (as mentioned before) and Manrique-Losada et al. (2016) that investigate business documents. On the other hand, the requirements extraction could be from multiple sources, as Manrique-Losada et al. (2016) mined heterogeneous data sources such as emails, forums, reviews, user stories, and different methods like classification, named entity recognition, and sentiment analysis to gather, aggregate, and elicit the requirements.

3.2 Topic Modeling from Social Media

Topic modeling in textual content is a broad field of research and has numerous use cases both in academia and the industry. Topic modeling is, first, the unsupervised act of segregating the parts of text into distinct groups that are different from each other, but internally share similar meanings; and second, assigning topics to each group representing the overall theme of the group. As this thesis focuses on social media and, more specifically Twitter as the main source of textual content, we examine the topic modeling research done investigating the topics of discussion in social media.

One of the most widely used methods for topic modeling is Latent Dirichlet Allocation (LDA) Blei et al. (2003) and many researches employed LDA for modeling topics in social media in different contexts. Hong and Davison

(2010) did an empirical study on topic modeling on Twitter. They employed and compared user classification scheme and LDA for topic modeling and assigning topics (classes) to messages. As a result, they concluded that the result of topic modeling of tweets using LDA even with is comparably poor even with enhancement and text pre-processing. [Alvarez-Melis and Saveski \(2016\)](#) employed two topic modeling techniques on a dataset of gathered tweets, namely the famous LDA, and Author-Topic Model (ATM). Comparison of topic modeling results is difficult by their nature because there are no baseline and absolute truth about the existing topics in the dataset, specifically when it is unlabeled. In this study, they gathered the tweets using pooling and querying different major subjects of discussion, such as politics, music, sports, etc. And therefore used a soft membership system to evaluate the result of the topic modelings. They concluded that, overall, ATM performs better than LDA. [Negara et al. \(2019\)](#), which is a (relatively more recent study) concluded that in general, LDA works better than the opposing method in the study Latent Semantic Analysis (LSA)), which is also a relatively old method. However, LDA is stated to be better at topic modeling in documents and cannot work optimally when dealing with shorter texts such as tweets and short messages.

There is another research field used in the studies that perform indirect topic modeling which is clustering similar messages together. [Rosa et al. \(2011\)](#) gathered around 1 million tweets and used TF-IDF to convert the tweets to vectors. Then used K-Means clustering technique aiming to group similar tweets in the same groups. They also used LDA as the baseline. Although, as mentioned previously, since topic modeling and clustering are unsupervised tasks, the effort of validating the resulting topics and clusters

is extremely difficult. In this study, the authors take the initial hashtags used for gathering the tweets as soft-level topics and compare the method using them. They concluded that clustering tweets show surprisingly high evaluation scores and outperform LDA in topic modeling. However, they did not specify how to assign topics to clusters of tweets and merely used hashtag membership as a form of evaluation of the clustering methodology. [Rejito et al. \(2021\)](#) used a similar approach as [Rosa et al. \(2011\)](#). They also employed a combination of TF-IDF vectorization and K-Means to cluster tweets. They did not provide any evaluation of the quality of the clustering; however, they used the most requested words of the clusters for labeling the clusters as the representation of the clusters. [Lossio-Ventura et al. \(2021\)](#) did a comprehensive study on the application of different topic modeling techniques on health-care-related tweets, particularly in distinguishing between tweets related to different diseases. They set up seven well-known topic modeling techniques available for short text topic modeling, one of which is GSDMM [Yin and Wang \(2014\)](#), which is a collapsed Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture model (DMM) designed for short text clustering and topic modeling and is one of the most recent short text topic modeling techniques. In terms of clustering, they used a combination of two vectorization techniques, namely TF-IDF and Word2Vec [Le and Mikolov \(2014\)](#) and K-means for tweet clustering. They used both internal (density and separation) and external (evaluation with a subset of tweets with hashtags) indices to evaluate the results [Kumar et al. \(2020\)](#). Overall, Online Twitter LDA followed by GSDMM, outperformed other methods in resulting in dense while distinct clusters judging by the internal index, and the combination of TF-IDF and K-means outperformed others in the

external index (while Online Twitter LDA showed a significant decrease). The last but most related topic modeling research is conducted by [Ito and Chakraborty \(2020\)](#). The main purpose of this study was to analyze the tweets regarding the COVID-19 pandemic. They tested three different text vectorization methods (namely TF-IDF, BERT, and Sentence-BERT) in combination with K-Means clustering to cluster the tweets and extract the topics. Even though they did not mention how they would extract the topic of each cluster, they provided a quantitative cluster evaluation method based on cosine similarity and euclidean distance. They concluded that clustering with Sentence-BERT will result in much more cohesive clusters compared to BERT and TF-IDF. Finally, the overall results from the reviewed studies show that the clustering method with either TF-IDF or Sentence-BERT and K-means results in acceptable outcomes in most cases.

3.3 Knowledge Extraction

Requirements and different topics extracted from textual data are not enough for domain experts to make data-supported decisions. The domain experts require more information regarding the extracted elements to be able to use them in their decision support systems. There is another line of research that focuses on extracting knowledge from text that has not been paid much attention to compared to mere requirements extraction and topic modeling from text. Moreover, validating the extracted knowledge is extremely difficult because the extracted knowledge could be evaluated only after applying them in practice in real-life setups. Which is why this field lacks proper effort in developing methods for knowledge extraction.

Here we cover the important studies conducted in this research path.

[Kitamura et al. \(2007\)](#) proposed an ontology-based system that uses domain ontologies during the requirements elicitation process to supplement domain knowledge to requirements analysts. Although, the creation and evolution of the requirements list by the requirements analysts. They use the ontology rules created by the requirements analysts to map the requirements to the proper domain ontology to provide related knowledge regarding the extracted requirements. [Abad et al. \(2018\)](#) developed a mobile app for assisting analysts during the elicitation process. The app uses a publicly available industrial dataset named ThyssenKrupp which is a dataset of interviews with clients about requirements. The answer to the questions in the interviews will further be mapped to the requirements as additional information is provided about the extracted requirements. Unlike the pure requirements extraction and topic modeling methods, most of the existing knowledge extraction methods.

3.4 Conclusions

As depicted in Figure [3.1](#), this study is a combination of four main components, namely requirements engineering, topic modeling, information retrieval, and applications of NLP for RE. We have covered a review of the studies conducted related to these components in the sections of this chapter. We have also listed the studies that are related to different parts of the developed methodology throughout this thesis. Each contains strengths and weaknesses that will be addressed in the Conclusions [7.1](#) of the thesis as a comparison with the developed DeKoReMi methodology pipeline.

Section 3.1 have gone over different requirements extraction methods, such as text classification and sentence analysis, and mentioned the most related and cited research conducted to develop each methodology. Section 3.2 discusses the studies that employ the most common techniques for extracting different topics of discussion from text and discusses the results, which highly influenced the development of DeKoReMi. And lastly, in Section 3.3 we explained the necessity of knowledge extraction and mentioned those studies related to requirements' knowledge extraction. Overall, in the body of research in requirements engineering and the discussed research, there are multiple features that a method could provide and bring value to the table for decision-makers during the decision-support process. Table 3.1 is a summary and lists the features and how each of the mentioned studies contributes to each of the values. 3.1.

Features	Studies
dynamic topics (or requirements)	Rosa et al. (2011) , Lossio-Ventura et al. (2021) , Rejito et al. (2021)
providing knowledge	Kitamura et al. (2007)
empirical evaluation	Hong and Davison (2010) , Haque et al. (2019)
semantic analysis	
multi-purpose or domain	Rosa et al. (2011) , Lossio-Ventura et al. (2021)
automated process	Jafari et al. (2021) , Haris et al. (2020)
used deep learning	Ito and Chakraborty (2020)
requirements (or topic) validation	Rosa et al. (2011)
method for topic assignment to text	Lossio-Ventura et al. (2021)

Table 3.1: The list of features and values offered by the reviewed studies

It could be understood that none of the mentioned related studies are classified under "deep learning usage" and "semantic analysis," which will

be further discussed in Section [6.1](#).

Chapter 4

Methodology

4.1 Overview

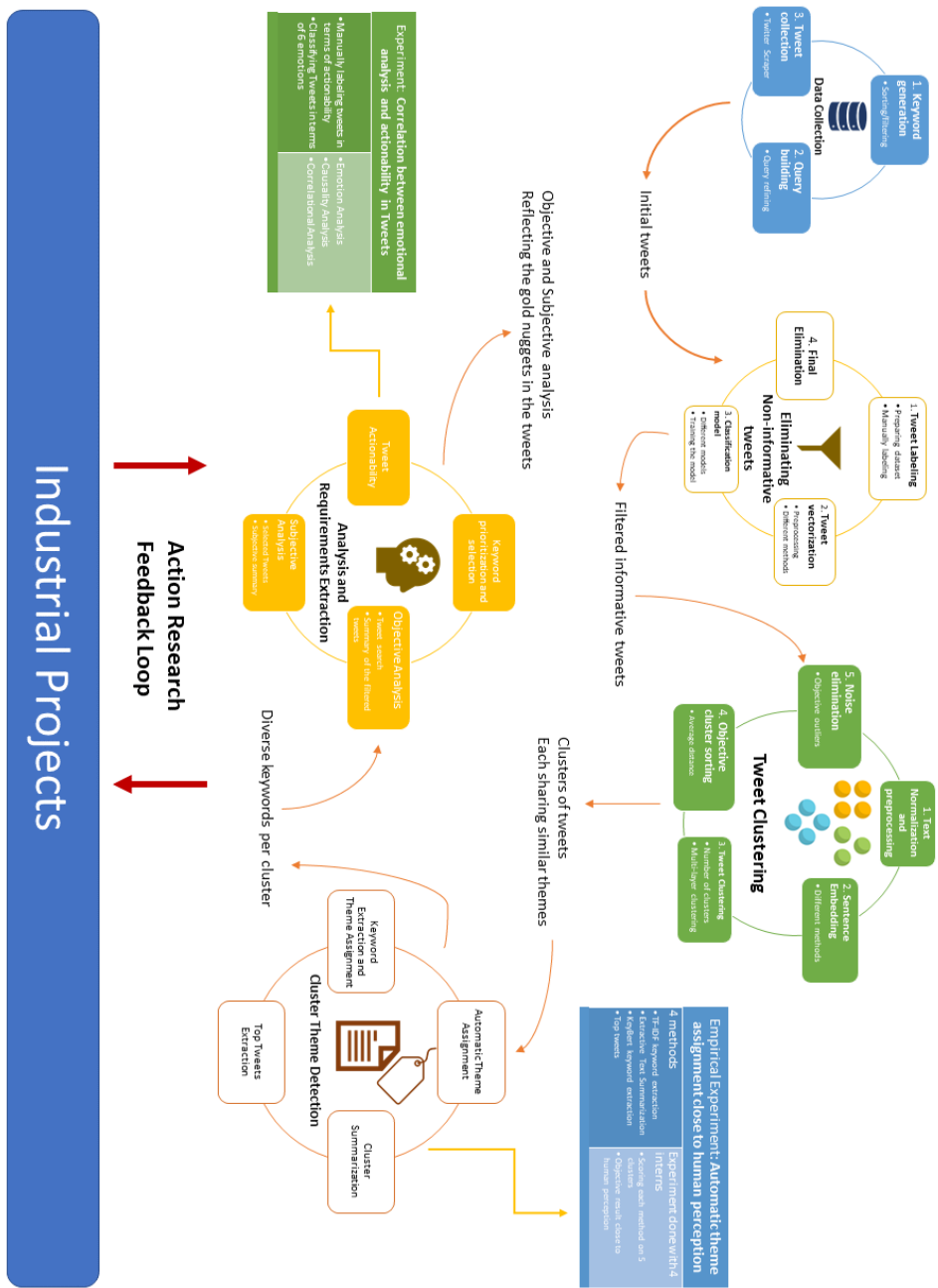
In this chapter, we introduce the DeKoReMi and explain in detail how to extract requirements and the most important parts from large and mostly non-informative textual content for decision-making. The DeKoReMi consists of multiple steps, and each will be explained in detail in this chapter.

4.2 Workflow

The overview of the pipeline of DeKoReMi is depicted in Figure 4.1.

Figure 4.1 consists of three main parts. First, there are the main steps for the DeKoReMi; each will be further explained in detail in the following sections of this chapter. These main steps of the DeKoReMi pipeline are namely Data Collection (4.3), Noise Elimination (4.4), Tweet Clustering (4.7), Cluster-Theme Detection (4.8), and Analysis and Requirements Extraction (4.9). Each is crucial to the track of the methodology to achieve the goal of

Figure 4.1: The outline of the experiments and steps done for this thesis and how they connect to each other.



the thesis and answer the RQs. The overall process is explained in the rest of this sub-section.

Secondly, there are the inter-step results which are the inputs and outputs of each step listed before. To explain the inter-step results, we should go over the process one step at a time and explain the input and output of each step. First, in the data collection step (Section 4.3) we use a few sub-steps to filter, target, and collect the most related tweets to our research subject. This will create the output of this step, the "initial tweets", which is also the input of the next step, "Noise Elimination". In the noise elimination section (4.4), we feed the initial tweets to a classification model we develop during this step to remove unnecessary and non-informative tweets to output the "filtered informative tweets" which will be the input of the next step. In the next step, we build a clustering process (Section 4.7) to group the tweets into contextually similar clusters of tweets. Then we assign topics and themes of discussion to each tweet-cluster in the next step (Section 4.8) to understand what topics and discussions the tweets of each cluster are addressing. The output of this section is keywords assigned to each cluster that describe the context of that cluster that is close to human perception. Finally, we take all the gathered information up until now and do two analyses (Section 4.9). The first analysis is the objective analysis which is generated completely automatically. The objective analysis consists of two sections, the automatically filtered tweets that contain the most important and valuable keywords in each cluster and the automatically generated summary of all the gathered tweets containing those keywords. And one is the subjective analysis which field experts author after digesting all the gold nuggets generated in the objective analysis by state-of-the-art NLP models.

The third and last part of the thesis is the complementary experiments done to accelerate the process of extracting requirements and making decisions based on the extracted requirement and gold nuggets of information (explained in detail in chapter 5). The two auxiliary experiments are first, an experiment on automatically assigning themes of discussion to tweet clusters that are close to human perception (Section 5.3). In this experiment, we studied 4 different methods for assigning themes of discussion to each cluster and found the best one that generates the closest themes to human perception. The results of this experiment are used in the 4'th main step in the pipeline of requirements extraction to automatically assign themes of discussion to each cluster. And the second experiment investigates the correlation of a tweet's attributes, like sentiment, emotion, news related, etc., to the tweet being the source of one or more action(s) - the study of the actionability of the tweets (Section 5.3). The result of the second experiment will help us develop a model to auto-detect actionable tweets related to our research subject without any other analysis.

The original workflow and pipeline consisting of the main steps designed for DeKoReMi are explained in this section in detail. The experiments are described in Chapter 5 as empirical studies done in the course of this thesis.

As introduced in Chapter 1 and could be seen in Figure 4.1, the DeKoReMi has been developed using "Action Research" methodology over three real-life industrial projects. This means there is an interactive feedback loop and inquiry process between doing research and performing action (as depicted in Figure 4.1). In this chapter, we only explain the developed methodology pipeline and the detailed explanation of its steps. The development process of DeKoReMi using industrial projects have been delivered

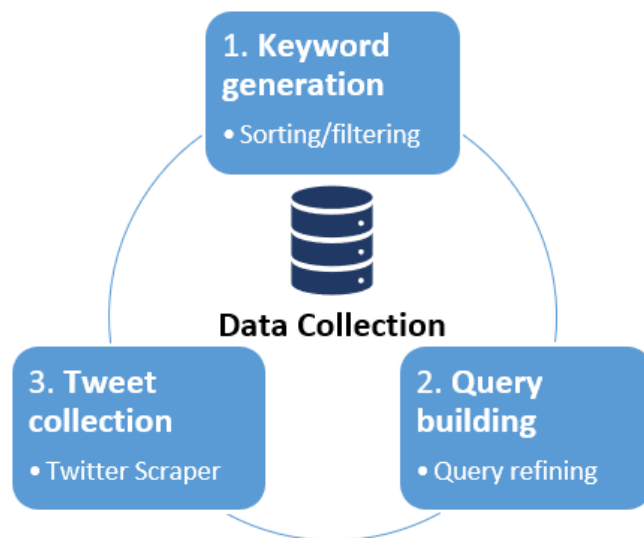
in Chapter 5.

4.3 Data Collection

Data collection is the first step of most Machine Learning based tasks and methodologies and is one of the most important parts of every data-oriented method. The quality of the input data has an absolute positive correlation with the quality of the resulting output. The better the input, the better the output. So it is very important to pay attention to every step of the data collection in order to be able to expect reasonable outcome.

This section is the first part of the pipeline visualized in Figure 4.1, and the more comprehensive diagram of this section completing the previously mentioned overall diagram can be seen in Figure 4.2. The output of this

Figure 4.2: Diagram for the first part (Data Collection) from Figure 4.1



step is the initial set of tweets as demonstrated in Figure 4.1.

Variations of Data Types and Data Sources

First, we need to determine what type of data we need for the task. For example, if it is going to be images for image processing tasks or multimedia for computer vision or structure modeling tasks. The type of data has a direct impact on selecting the data source or data sources that would be the next step in data collection. Even the same data types may have multiple sub-types. For example, textual content could be either structured (i.e., data stored in databases in a structured manner) or semi-structured (i.e., tweets organized by hashtags or emails), or unstructured (i.e., online reviews or media posts), which may differ depending on the data source.

Then we need to decide which data source or data sources to use to collect the data and, furthermore, whether we need multiple data sources or a single data source to gather the data. If we decide to use multiple (heterogeneous) data sources, the data could have different structures; therefore we need to consolidate the different data gathered from different data sources, and consolidate them before analyzing or passing them to the methodology pipeline. Otherwise, we need to implement different methods or models for each data source and then consolidate the outcome, which sometimes could be the only option.

Here in this research, since we are mining public requirements, the data is English tweets, and thus we have only one data source (Twitter API). Thus, the data type is semi-structured textual content which is not necessarily grammatically correct. Even though we do not have multiple data sources yet still, different tweets may have different structures. For example, they may contain images and links, and will have different lengths. Therefore

we need to normalize and pre-process the tweets before passing them to the Machine Learning pipeline. The reason behind choosing Twitter as the data source is that people express themselves and share their thoughts and opinions candidly, which would make Twitter a great source to look for people's concerns and difficulties to extract requirements worth investing in fixing them, which would ideally have a direct impact on the quality of the citizens' lives.

Data Collection Method

As mentioned before, the data is chosen to be tweets, so the data source is Twitter. There are multiple methods for crawling tweets from Twitter. For example, web scraping is a technique where we use tools like Scrapy¹ built using Python programming language to impose as web browsers to download web pages and then crawl over and process the downloaded web pages to retrieve the data (Hernandez-Suarez et al. (2018)). But it can trigger many obstacles, such as throttling for web page requests and the difficulty and complexity in crawling the different and ever-changing versions and designs of the web pages. However, one of the most convenient methods for fetching tweets is using the Twitter API provided by Twitter². There are Software Development Kits (SDKs) already developed for fetching tweets using Twitter API, such as Tweepy³ developed in Python, but since we need more flexibility and the ability for customization in terms of fetching with customized attributes and filters and saving the tweets in specific forms, we

¹<https://scrapy.org/>

²<https://developer.twitter.com/en/docs/twitter-api>

³<https://www.tweepy.org/>

have developed our custom SDK⁴ for crawling the tweets⁵. The developed toolkit can be very flexible in terms of using all the filtering and searching abilities that Twitter API v2 provides and storing the tweets in any required forms. The tool also is designed to be robust, saving snapshots of data in case of any error and handles the request throttling measurements that Twitter forces on its APIs to be able to fetch high volumes of tweets fast without any issues (Masahati (2021)).

Query Generation for Data Collection (Data Targeting)

As discussed in Section 4.3 we use an SDK that we developed to target and retrieve the tweets from Twitter. The SDK used Twitter API v2 to fetch the tweets, which requires multiple input attributes to adjust the required response. A few of the important attributes are as follows: the search query to target the tweets, the date and time in which we are searching for the tweets, the maximum number of tweets that we require to retrieve, and the required fields^{6,7} for the response, such as the text, the author id, geo information, and public metrics such as the number of likes, replies, or retweets. The single most important attribute among the mentioned attributes is the query that is responsible for targeting the related tweets to the study subject. The quality of the used query drastically affects the quality of the resulting tweets, which is the basis of the whole research. So it needs to be built carefully. In the following sub-section, we describe the process of how

⁴Software Development Kit

⁵<https://github.com/mammalofski/Twitter-Scraper>

⁶<https://developer.twitter.com/en/docs/twitter-api/fields>

⁷<https://developer.twitter.com/en/docs/twitter-api/data-dictionary/object-model/tweet>

to generate the most effective query to target and gather the related tweets.

Keyword Generation

The query mainly consists of a combination of keywords using logical operations like 'and', 'or', or 'exclusion' which will be discussed in the Sub-section [4.3](#) in detail. Since the keywords constitute the majority of the query, it is very important to select the most related keywords. So here is the process of coming up with the keywords:

The keywords need to be generated seeking consultation from the selected field experts in each empirical study. We request the field experts to come up with keywords in three classes: the keywords that are the most related to the subject under study, the keywords that could target the related tweets if combined with other keywords, and the keywords that may seem related, but should be excluded from the search (the negating or excluding keywords). For the second type of keyword (the keyword combinations), they should also explain the subject under study for us (researchers) to be able to further correctly combine the keywords that target the tweets.

As an example, suppose that we are investigating the tweets that are related to "digital workplace" in the oiling industry; an example of the first keyword class is "collaboration" or "workplace". An example of the second keyword class is the combination of the words "hybrid" and "remote" with the word "work", so "hybrid work" or "remote work". And an example of the third (excluding) keywords is "software development" because they most often are included in the tweet search results, and we are aiming to experiment with the oil industry.

We take the final keywords and generate the queries in Sub-section [4.3](#).

Query Generation and Optimization using Keywords

After we gathered the underlying keywords to use in building the queries, we started drafting initial queries based on the keywords and the discussions we've had with the field experts as mentioned in Sub-section 4.3.

To be able to generate the queries, we must follow the API and logic for generating search queries for tweets in Twitter API v2⁸. There are certain notations and operations to form a query, such as the Boolean operations that are "AND", "OR", "NEGATION" (or exclusion), and "GROUPING" logic. For example, if we want the tweets that contain "happy" or "happiness" but do not contain "birthday" the resulting query will be as follows: *"(happy OR happiness) -birthday"*. We can also remove the retweets by adding a *"-is:retweet"* at the end of the query, which is very helpful because it filters the retweets out since they add no value to the natural language processing pipeline. So, using the three types of keywords mentioned in Sub-section 4.3 and the mentioned query notations, we form multiple queries that search for the related tweets to the subject under study. The rough and general query template for the keywords are as follows: *"(Class 1 keywords separated by "OR") OR (Class 2 keyword combinations using "AND" and "OR" operators) -(Class 3 keywords) -is:retweet"* where Class 1, 2, and 3 keywords are the three different keyword classes mentioned in Sub-section 4.3. An example query of the mentioned example keywords in Sub-section 4.3 is as follows: *"(collaboration OR workplace) OR ((hybrid OR remote) AND work) -(software development) -is:retweet"*

After generating the initial draft queries, they need to be tested and

⁸<https://developer.twitter.com/en/docs/twitter-api/tweets/search/integrate/build-a-query>

optimized. Luckily, to test the draft queries, Twitter provides us with a built-in advanced search in its website⁹ that we can both generate and test our search queries easily in a graphical user interface provided by Twitter (link in the footnote). Using the provided search UI, we test our different queries and skim through the top results. Judging by the top results, we understand how satisfied we are with the results of the tested query. An example of testing a search query is shown in Figure 4.3. Judging by the top results from Twitter advanced search bar, we could distinguish what keywords we want to add/remove/adjust to/from/in the search query. This tweaking is necessary to eliminate most of the unwanted results and narrow down the resulting tweets to be as related as they could to our subject under study. After optimizing and finalizing the search queries, we now should use the SDK developed in Sub-section 4.3 with the generated queries to gather the initial dataset and the target-related tweets.

4.4 Eliminating Non-informative Tweets

Tweets are public opinions by their nature and are mostly non-informative for business decision purposes. Eliminating the non-informative tweets helps us extract the most important ideas and requirements from the tweets quicker, as most of the tweets may be obstacles and are irrelevant when it comes to making crucial business decisions. In this section, we develop an optimized method to eliminate most of the non-informative tweets using Machine Learning and classification techniques.

Noise elimination in Natural Language Processing is a classification task

⁹<https://twitter.com/search-advanced>

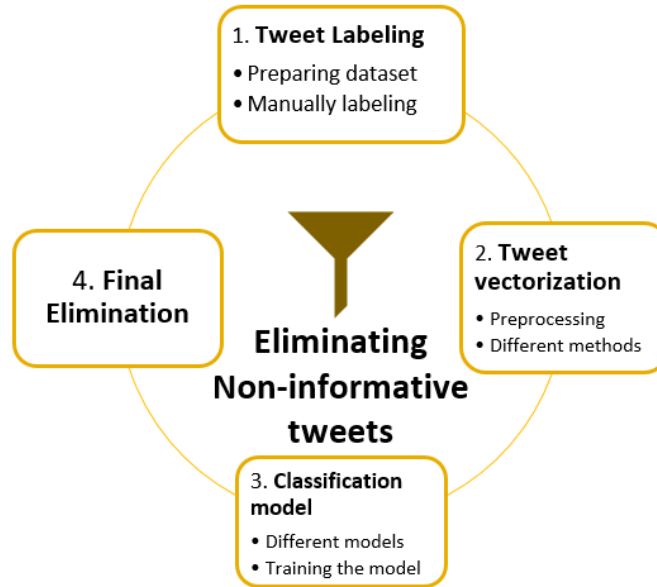
Figure 4.3: An example of searching the query "*(collaboration OR workplace) OR ((remote OR hybrid) AND work) -(software development)*" in Twitter advanced search.



that classifies the tweets in terms of informative and non-informative to our subject under study. Each classification task in Machine Learning has two main parts. The first one is Data collection, and the second one is Model development and training. And lastly, after we trained the developed model with the collected data, we applied the trained model to the dataset

to classify the tweets and filter out the non-informative tweets.

This section describes the second part of the pipeline visualized in Figure 4.1, and the more comprehensive diagram of this section completing the previously mentioned overall diagram could be seen in Figure 4.4. The Figure 4.4: Diagram for the second part (Noise Elimination) from Figure 4.1



output of this step is the filtered and informative tweets as demonstrated in Figure 4.1.

Data Annotation

In this research, even though the methodology is a general pipeline that can be applied to various case studies, the data annotation needs to be specific to the subject under study. Since here we aim to distinguish between informative and non-informative tweets, the definition of informativeness may change depending on the subject under study. Thus, when the subject

under study changes, we need to prepare the training data specific to that subject under study.

Data annotation needs to be manual and prepared by human beings who understand the subject and the nature of informativeness in the given subject. Which is why we coordinate the first round of annotation with the field experts who define what "informative" means in the subject under study. We prepare a random subset of tweets for each field expert and ask them to define informativeness, and label the given tweets in terms of informative and non-informative. The required amount depends on the developed classification model and is mostly limited by the manpower for data annotation. If the classification model requires more training data, we annotate more, seeking help from our research assistants using the same process.

The annotated data needs to be balanced, meaning that it has to have a sufficient amount of tweets labeled both as informative and non-informative. If the annotated data is imbalanced, it might create bias when training the classification model. However, as mentioned before, here we are working with tweets as the main source of data, and tweets are non-informative by nature; thus, it is only expectable that the prepared data is highly leaned and imbalanced in favor of the non-informative tweets. On the other hand, if the number of non-informative tweets is high, the accuracy of the final model will be higher in detecting the non-informative comparing the informative ones. It means that if the model eliminates tweets detecting it to be non-informative, the prediction will most probably be correct, but if the model predicts a tweet to be informative, the accuracy may be lower in comparison. This imbalance here is acceptable because we aim to eliminate as much as

non-informative tweets as possible while preventing falsely eliminating the informative ones.

Model Selection

There are various machine learning models for text classification; each has its pros and cons and use cases considering the application and its context. Not all of them perform the same way in different classification applications, as in each application, there are different attributes to the text and the nature of the classification task that are the most important in the classification process. For example, some classification tasks may be relevant to the statistical attributes of the text, and in others, the context of the text plays a more significant role in grouping the text into different classes. Here we introduce the context of our research and introduce multiple applicable methods and demonstrate how we choose one to use for each subject under study, as the used model may be different depending on the subject under study.

Since we cannot predict the accuracy and the effectiveness of each text classification technique, we should design different models based on the general application (which is classifying tweets in terms of informativeness), and validate all of them using the annotated data in Sub-section 4.4. Finally, we choose the model with the highest accuracy to train and filter out the non-informative tweets.

In NLP classification tasks, there are two parts for each classification model. The vectorization technique and the classification model. We cannot predict what vectorization technique works best with what classifica-

tion model, and we should test all the combinations of them and select the classification technique (combination of vectorization technique and classification model) with the highest accuracy. The proposed vectorization techniques are TF-IDF (as described in Section 2.2), and BERT (as described in Section 2.2), and the classification models are Naive Bayes (NB), Random Forest (RF), and BERT built-in text classifier (as described in Section 2.1). Since the BERT built-in text classifier only works with BERT text embeddings, we have five classification techniques in total (which are: TF-IDF + NB, TF-IDF + RF, BERT + NB, BERT + RF, and BERT built-in classifier). The model selection procedure is that we train all five classification techniques and select the one combination with the highest accuracy specific to the subject under study.

Model Training, and Noise Elimination

After we calculated the accuracies of all the different classification techniques in Section 4.4, we select that model and train in on the whole dataset annotated in Section 4.4. And then, we use the trained model on the actual initial dataset collected in Section 4.3 and classify the initial dataset into the two classes of informative and non-informative. Finally, we distinguish the tweets classified as non-informative as noise, and we remove them from the dataset. The remaining tweets are classified as informative and kept for further investigation on the following steps in the methodology pipeline.

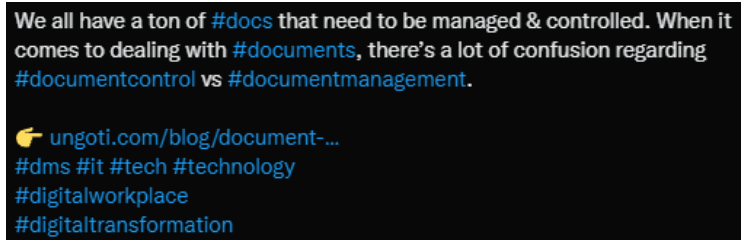
4.5 Text Normalization and Processing

This section describes the first step (Text normalization and Pre-processing) in the five-step process for tweet clustering demonstrated in Figure 4.6.

Now that we have eliminated the noise or the non-informative tweets from the initial dataset, we have to prepare and pre-process the rest of the tweets for the future NLP tasks in the pipeline. The pre-processing and text normalization consists of two phases. The first one is removing the parts that add no contextual value to the tweets and may distort the input for the following NLP techniques that will be applied to the text. And the second one is pre-processing the tweets such that they would be acceptable for BERT since the next phase is tweet embedding with BERT.

For the first text normalization part, we remove the usernames starting with "@" that are used to notify and mention another username in a tweet. Next, we eliminate all the URLs since they add no value when used alone using their URLs. For the second text normalization part, we should pre-process the tweets such that they would be accepted as valid input for BERT text embedding. To do so, first, we lowercase all the text, and then we should adjust the tweets such that the text gets closer to correct English grammar. This is because BERT is pre-trained on grammatically correct textual content and would have a better understanding and result while processing text with correct grammar. As an example, looking at Figure 4.5 we notice that there are a few elements that disrupt correct English grammar in the given sample tweet. The first element is the hashtag sign itself (#) in between the text. The second element is the ending hashtags which have no position in English grammar. The third element is the Emojis which cannot be understood

Figure 4.5: Example tweet for text normalization



We all have a ton of #docs that need to be managed & controlled. When it comes to dealing with #documents, there's a lot of confusion regarding #documentcontrol vs #documentmanagement.

👉 ungoti.com/blog/document-...

#dms #it #tech #technology
#digitalworkplace
#digitaltransformation

by BERT. Thus, to make the tweets closer to correct English grammatically written text, we should fix the mentioned issues as well.

So, in general, we perform the following text normalization procedures on all the tweets: 1) remove the usernames starting with the character "@". 2) remove the URLs. 3) convert to lowercase. 4) remove the hashtag characters ("#") in between the sentences. 5) remove all the ending hashtags that usually are appended at the end of tweets for search purposes. 6) remove the Emojis.

All the above-mentioned procedures are done using Regex [Aho \(1991\)](#) search, and replacement performed using the `re`¹⁰ or the `RegEx` library in Python programming language.

4.6 Text Vectorization

After normalizing the tweets, removing the non-valuable parts, and pre-processing them to prepare them for BERT, we convert the tweets into context-aware vectors to further cluster them in Section 4.7 into contextually similar groups of tweets.

This section describes the second step (Sentence Embedding) in the

¹⁰<https://docs.python.org/3/library/re.html>

five-step process for tweet clustering demonstrated in Figure 4.6.

BERT

As fully described in Section 2.2, BERT has the ability to extract word and sentence embeddings from textual content. The reason that we selected BERT for sentence embedding over other text vectorization techniques described in Section 2.2 is that BERT is a pre-trained transformer over large textual content paying attention to the surrounding words when being pre-trained (as described in Section 2.1) and thus, generates context aware embeddings for words and therefore also for sentences. This very attribute of BERT (being context-aware) is the main reason we selected BERT for text vectorization, because the resulting vectors will be further clustered, and the aim of the clustering is to group similar contextually similar tweets together to elicit different themes of discussion among the tweets in the subject under study.

There are multiple methods to generate sentence embeddings using BERT. Three of them will be explained in detail in the following sub-sections.

SEP Tokens

As mentioned in Section 2.2, BERT adds two more special tokens to the text before generating the word embeddings, namely [CLS] token, which comes before each document in the corpus and is used for representing the whole document and for text classification purposes. The second one is [SEP] token which comes after each sentence and is used for separating each part of the document. Since BERT generates word embedding for all the

words, including the two special tokens, we can use the word embedding generated for the [CLS] at the beginning of each tweet as the representation of the whole tweet for applying the clustering.

Average Weighted Tokens

There is another method for extracting a representation of the whole tweets from the word embeddings generated by BERT. We calculate each feature (F_i) of the final embedding representing a tweet with N tokens and each token containing M features using the Equation 4.1.

$$F_i = \frac{\sum_{j=1}^N F_{ij}}{N} \quad (4.1)$$

in which i is between 1 and M . In other words, if we assume that the tweet contains N tokens and therefore, the word embedding includes N embeddings (or vectors representing each word), and each word embedding contains M features, we define the i 'th feature the sentence embedding representing the whole tweet as the mean of all the i 'th features in all the word embeddings in which i is between 1 and M .

Sentence BERT

There is another version of BERT designed specifically trained for understanding sentences called Sentence BERT (Reimers and Gurevych (2019)). We can use Sentence-BERT to extract sentence embeddings for the tweets¹¹ instead of first extracting the word embeddings and then converting the word embeddings to sentence embeddings. Similar to BERT, Sentence BERT has multiple pre-trained models, each used for different purposes. Here

¹¹<https://www.sbert.net/>

we use the "all-mpnet-base-v2" pre-trained model¹² which is an all-round model tuned for many general use-cases. This model is trained on a large and diverse dataset of over 1 billion training pairs ([huggingface \(2021\)](#)). This model will generate 768 features (a vector of 768 numbers) per tweet.

Final method selection of Sentence Embedding with BERT

To compare the effectiveness of the above-mentioned sentence embedding methods using BERT, we need to first perform the clustering and then analyze the quality of the resulting clusters and then decide which one performs the best. We had done no objective analysis to decide which one outperforms the others, but subjectively we observed that they all are acceptable and will result in roughly the same number of clusters using the elbow method. We finally chose Sentence-BERT over the others due to a subjective observation by field experts, which will be discussed in Section 5.1. Of course, there is a need for objective analysis and study to confirm (or reject) this subjective observation even though it was done by the field experts.

4.7 Tweet Clustering

Before explaining the process of clustering the tweets, we should explain why the idea of clustering the sentence embeddings representing the tweets will group tweets with similar contexts together. The idea comes from the theory suggested by [Reimers and Gurevych \(2019\)](#) that we could measure the semantic textual similarity of two texts by mathematically calculating

¹²<https://huggingface.co/sentence-transformers/all-mpnet-base-v2>

the similarity of the sentence embeddings representing the texts (i.e., using cosine similarity as suggested by [Reimers and Gurevych \(2019\)](#) or euclidean distance). Therefore, if we cluster the sentence embeddings representing the tweets into clusters of vectors close to each other, we could group semantically similar tweets together by clustering their sentence embedding representations.

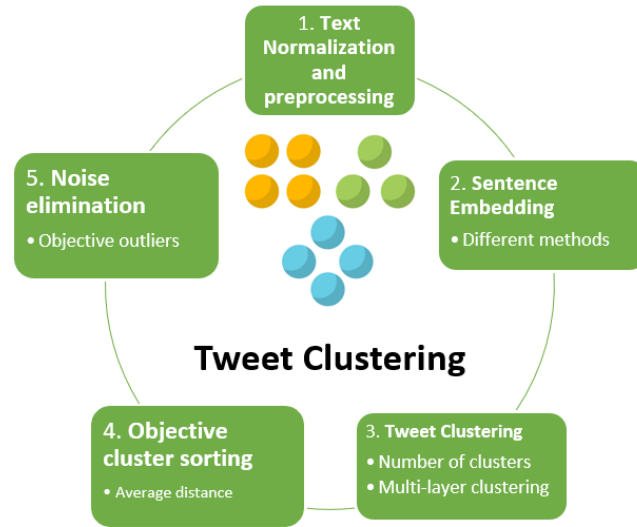
After generating sentence embeddings (converting the tweets into vectors) from the tweets, each containing 768 features using a pre-trained Sentence BERT model mentioned in [Section 4.6](#) we need to cluster them into groups of tweets sharing similar context to distinguish the different themes of discussion. We use a multi-layer K-Means clustering process to break the tweets into meaningful and fine-grained clusters of tweets, each discussing different subjects.

The sections [4.5](#), [4.6](#), and this section ([4.7](#)) together make the third part of the pipeline visualized in [Figure 4.1](#) and the more comprehensive diagram of this section completing the previously mentioned overall diagram could be seen in [Figure 4.6](#). The output of this step is clusters of tweets, each sharing similar themes as demonstrated in [Figure 4.1](#). Currently, we are at stage 3 of the diagram, as the first two stages were explained in sections [4.5](#) and [4.6](#), respectively.

K-Means

As mentioned in [Section 4.7](#), we cluster the sentence embedding representing the tweets aiming to group semantically similar tweets in the same clusters. Here we use K-Means algorithm (which is the most famous vector

Figure 4.6: Diagram for the third part (Tweet Clustering) from Figure 4.1



clustering method) to cluster the sentence embeddings representing the tweets. To be able to apply K-Means, we first use the elbow method to determine the optimal number of clusters that creates the lowest number of clusters with the highest variety in context for this specific dataset. After determining the optimal number of clusters using the elbow method, we apply the K-Means on the sentence embeddings and cluster the vectors into K clusters. Now, after we map the sentence embeddings back to the original tweets, we have K clusters of tweets each sharing similar semantic themes of discussion.

Cluster Meta-Data Extraction

After clustering the tweets to K clusters, we have K cluster centroid, one per each cluster. Using the K cluster centroids, we could extract extra data from the clusters and tweets that will be useful in the future in the cluster noise elimination (Section 4.4), the multi-layer clustering phase (Section

4.7), for finding the top tweets for each cluster, and for determining how focused or dispersed each cluster or sub-cluster is.

The first meta-data that could be extracted from the clusters, is the size of each cluster which is calculated by counting the tweets in each cluster and representing it as N_c . The second and most important metric is the distance of each tweet from its corresponding cluster centroid. Since each tweet is represented as a 768-feature vector, all the centroids also contain 768-features and the distance of each tweet from its centroid could be calculated using the euclidean distance equation 4.2:

$$d(t, c) = \sqrt{\sum_{i=1}^n (t_i - c_i)^2} \quad (4.2)$$

in which t is the vector representing the tweet, c is the vector and the position of the corresponding centroid, and n is the number of features which in this case is 768.

After calculating the distance of each tweet from its corresponding cluster centroid, we can calculate the mean distance (average radius) of each cluster. The mean distance for each cluster is the mathematical average of the distances between all the tweets and the centroid in that cluster, calculated using equation 4.3.

$$md(C) = \frac{\sum_{i=1}^{N_c} d(t_i, c)}{N_c} \quad (4.3)$$

in which c is the cluster centroid for cluster C , N_c is the number of tweets in cluster C , and t_i represents each tweet in cluster C .

We will be using these two metrics in the following sections, as mentioned before.

Multi-Layer Clustering

(filter out the clusters using the field experts and the re-cluster the valuable clusters)

In most cases, the initial dataset is very large, even after performing the noise elimination in Section 4.4. Thus, the number of tweets in some of the clusters may still be large, or the clusters may be dispersed, which would be detrimental because a relatively large cluster (compared to other clusters) or a dispersed cluster may contain tweets that are contextually not as close to each other. Therefore, we select a number of the clusters from the first round of clustering and re-cluster them into smaller sub-clusters.

We use the metrics calculated in Section 4.7 to determine which clusters to re-cluster. The aim here is to choose relatively large and diverse clusters. Thus we sort the clusters in two ways, firstly, we sort them based on their sizes descendingly, and secondly, we sort them based on their mean distances from their centroids (or their average radius) descendingly. This way, we have the largest clusters at the top of the first list, and the most diverse clusters at the top of the second list. We choose the top 30% largest and most diverse clusters from these two sorted lists, and we select the intersection of the 30% large and 30% most diverse clusters to perform the re-clustering. The final selected clusters are at the top 30% of both sorted lists in terms of size and diversity, which means they are in comparison to other clusters, relatively large and diverse and are good choices for re-clustering.

After selecting the largest and most diverse clusters, we perform the K-Means method as mentioned in Section 4.7 on each individual cluster with separate and specific values for K and break them into sub-clusters

of tweets. We call this a multi-layer clustering because we have performed two layers of nested clustering.

This method could also be beneficial for breaking down even smaller but valuable clusters to extract sub-topics from the important clusters. For example, if we have a cluster labeled with the topic of "the state of the health-care system" it could be broken down to "response time in the healthcare system", "availability of healthcare", "mental health", etc.

4.8 Cluster Analysis and Cluster Theme

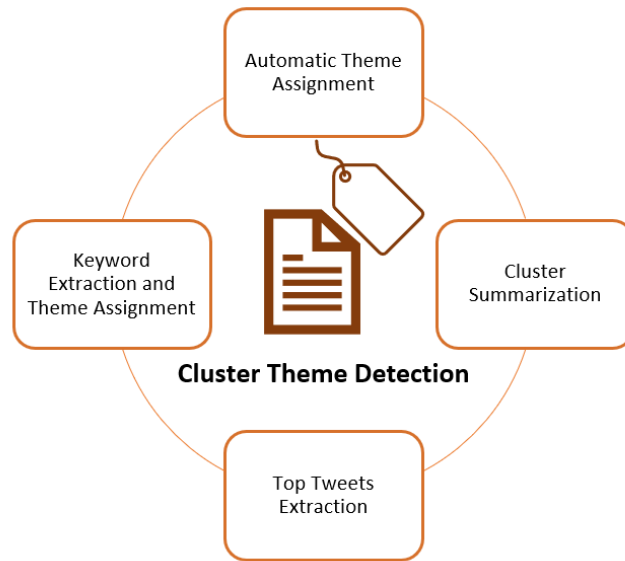
Detection

After clustering the tweets and breaking them into semantically similar groups of tweets, we analyze the clusters, firstly based on their numeric meta-data and then based on their content. In the content analysis, the aim is to automatically assign themes of discussion to each cluster and represent their overall themes and subjects using keywords. Furthermore, towards the goal of this section, we have designed an experiment to detect an optimal method to assign themes to each cluster that are close to human perception, which will be explained in detail in Section 5.4 in chapter 5. In this section, we describe four main methods for assigning themes of discussion to each cluster that helps us better understand the content of the cluster without having to read numerous tweets from the cluster. The effectiveness of the four methods and the closeness of their results to human perception is studied in the experiment described in Section 5.4 in detail.

This section describes the fourth part of the pipeline (Cluster Theme

Detection) visualized in Figure 4.1 and the more comprehensive diagram of this section completing the previously mentioned overall diagram could be seen in Figure 4.7. The output of this step is keywords representing the

Figure 4.7: Diagram for the fourth part (Cluster Theme Detection) from Figure 4.1



content and the theme of discussion for each cluster as demonstrated in Figure 4.1.

Cluster Meta-Data Analysis

Prior to the analysis of the content of the clusters and tweets, we could have an initial analysis of the metrics and meta-data extracted in Section 4.7 from the tweets. As mentioned in Section 4.7, we could sort the clusters based on their average radius (or mean distance from centroid) and size. It means that we could identify the clusters that are more focused on their theme of discussion than others, and similarly, we could identify the clusters that are too diverse to be able to assign a single theme of discussion to

them. Although, those clusters are most probably re-clustered in Section 4.7. We define the density of a cluster as a one-to-one correlation to the mean distance of that cluster (as defined in equation 4.3). The denser a cluster, the more focused that cluster is toward a single semantic theme of discussion. This conclusion is based on the theory suggested in [Reimers and Gurevych \(2019\)](#), saying that we could demonstrate the semantic similarity of two sentences by measuring the distance between their representing sentence embeddings. Therefore, the closer the sentence embeddings of the tweets in a single cluster, the more focused and semantically similar their theme of discussions is.

Outlier Elimination from Clusters

We also could identify the outliers in each cluster judging by their distances from the centroid and by comparing them with the mean distance in the same cluster. We define the outliers of a cluster as the 25% most distant tweets to the corresponding cluster. It means that we keep the 75% closest tweets to the centroid of each cluster and eliminate the others, and we call the remaining tweets as the effective radius of each cluster. In the future, we use only the effective radius to analyze the content of each cluster, i.e., when summarizing the cluster in Section 4.8, or when extracting the keywords from the cluster in Section 4.8.

Cluster Text Summarization

Since each cluster may contain thousands of tweets, reading all of them to determine the overall semantic theme may be difficult. Here we use an

automatic text summarization technique to generate a short summary of all of the tweets in each cluster. Reading a short summary of the tweets will definitely save much time in determining the overall theme of discussion in a cluster.

BERT Extractive Summarization

As mentioned in Section 2.3, one of the applications of BERT is extractive text summarization [Allahyari et al. \(2017\)](#). Here we use the BERT's extractive text summarizer which is also developed in Python¹³ to generate summaries for all the clusters.

To do so, we first filter out the 50% farthest tweets to the centroids of each cluster and only keep the 50% closest tweets to the centroids to generate the summary, since they are better representations of the general theme of the whole cluster. Then we concatenate all the remaining tweets and normalized them as we did in Section 4.5 to remove the unwanted parts of the tweets that bring no value to the summary, such as the links, usernames, and Emojis. And then we pass the final text to the BERT's extractive summarizer to generate the summary for each cluster. We use the tool's default settings as it generates acceptable results with a resulting number of sentences between 5 and 10, depending on the size of the cluster.

Keyword Extraction

Apart from the summary of a cluster generated in Section 4.8, the content of a cluster could also be represented as a series of diverse or similar keywords

¹³<https://github.com/dmmiller612/bert-extractive-summarizer>

that together convey the overall theme of discussion in the cluster. There are multiple methods for extracting important and effectively representative keywords from a text corpus, here we describe and use two of them that we use to extract the keywords for tagging the clusters.

TF-IDF

TF-IDF is a well-known method for information retrieval, text vectorization, and keywords extraction and is fully described in Section 2.2. In that section, it is mentioned that (quoting) "TF-IDF is a statistical method and a weighting scheme that tries to assign weights to each word in a document within a text corpus representing their importance towards defining the text corpus". And further, in that section, we described how to use the weighting scheme to generate the vector representing documents in a text corpus. The weighting scheme (defined in Equation 2.3) is designed to assign higher values to the words that statistically are more important in defining the documents than others, which is the very definition of keywords representing the documents (or in this case, tweets). Thus, in order to extract the important keywords for documents in a text corpus (or for tweets in a tweet cluster) we should select the most valued words after applying the TF-IDF to a text corpus.

Keyword Extraction using BERT (KeyBERT)

We use another keyword extraction technique based on BERT developed by Grootendorst (2020) called KeyBERT¹⁴ that is capable of detecting the important part of textual content and extracting the definitive keywords

¹⁴<https://maartengr.github.io/>

from those parts. KeyBERT uses the word embeddings generated by BERT (as discussed in Section 2.2 and cosine similarity to find the most similar phrases in the document that is closer to the content of the document itself (Grootendorst (2020)). A simple example of the KeyBERT usage and output could be seen in Figure 4.8. As could be seen in Figure 4.8, the tool is very

Figure 4.8: KeyBERT example taken from <https://maartengr.github.io/KeyBERT/#usage>

```
from keybert import KeyBERT

doc = """
Supervised learning is the machine learning task of learning a function that
maps an input to an output based on example input-output pairs. It infers a
function from labeled training data consisting of a set of training examples.
In supervised learning, each example is a pair consisting of an input object
(typically a vector) and a desired output value (also called the supervisory signal).
A supervised learning algorithm analyzes the training data and produces an inferred function,
which can be used for mapping new examples. An optimal scenario will allow for the
algorithm to correctly determine the class labels for unseen instances. This requires
the learning algorithm to generalize from the training data to unseen situations in a
'reasonable' way (see inductive bias).
"""

kw_model = KeyBERT()
keywords = kw_model.extract_keywords(doc)

>>> kw_model.extract_keywords(doc, keyphrase_ngram_range=(1, 1), stop_words=None)
[('learning', 0.4604),
 ('algorithm', 0.4556),
 ('training', 0.4487),
 ('class', 0.4086),
 ('mapping', 0.3700)]
```

easy to use, and returns as many keywords as requested, and each keyword could include any required number of words. The resulting keywords also include a score in terms of how accurate and representative the keyword is. This tool could also return diverse keywords if requested, as could be seen in Figure 4.9.

We use this tool to generate 3 different types of keywords per cluster, each containing 5 keywords (15 keywords per cluster in total as a representation of the content and overall theme of discussion in the cluster). Firstly,

Figure 4.9: KeyBERT diverse keywords example taken from <https://github.com/MaartenGr/KeyBERT>

```
>>> kw_model.extract_keywords(doc, keyphrase_ngram_range=(3, 3), stop_words='english',
                             use_mmr=True, diversity=0.2)
[('algorithm generalize training', 0.7727),
 ('supervised learning algorithm', 0.7502),
 ('learning machine learning', 0.7577),
 ('learning algorithm analyzes', 0.7587),
 ('learning algorithm generalize', 0.7514)]
```

we use the default settings to generate 5 single-word keywords. Then we generate 5 keywords that may contain up to 3 words per keyword using the 'keyphrase_ngram_range' attribute. And lastly, we use the 'diversity' attribute (as used in Figure 4.9) to generate 5 diverse keywords each containing up to 3 words. All these different types of keywords would be a good representation of the overall theme and content of each cluster.

Top Tweets of Each Cluster

Even though there may be thousands of tweets in each cluster, there could be found a few tweets in each cluster that could be fine representations of the whole cluster as their semantic meaning is clearly towards the same content of the whole cluster. To find those tweets objectively, we sort all the tweets in a cluster based on their distances to the centroid of the cluster (calculated in Section 4.7). We pick the top tweets that are the closest to the centroids of the correlated cluster regarding their distances from the centroid as the representation of the whole cluster. This method is very similar to the method used in BERT extractive text summarizer described in Section 2.3 to detect the most important sentences of a text document. Here we use the same method to extract the most important tweets in a

cluster and represent them as the overall theme of discussion in a cluster.

Combination of the Theme Assignment Methods

So far we have introduced four different automatic methods, namely cluster summarization using extractive text summarization 4.8, two keyword generation methods using TF-IDF 4.8 and BERT 4.8, and extracting the top tweets of each cluster 4.8. We can use these automatic methods to assign a theme of discussion to each cluster. Although, each may be different in terms of resulting in themes that are closer to human perception. Even though reading them for assigning themes to all of the clusters may still require human interception (other than the keyword extraction methods) and may be time-consuming, all of the mentioned methods are definitely faster and require less time in comparison to manually reading the tweets of each cluster and assigning topics to them.

Moreover, we will further discuss these four methods in detail in terms of closeness to human perception in the complementary experiments in Section 5.4. As the result of the mentioned experiment (which is discussed there in detail), we will be using the keywords generated by KeyBERT as the main representation of the content and overall theme of the cluster which is completely automatic and requires no human interception in the loop. Although, the other three methods, especially the text summarization technique will further be used in the process of analysis and requirements extraction in Section 4.9.

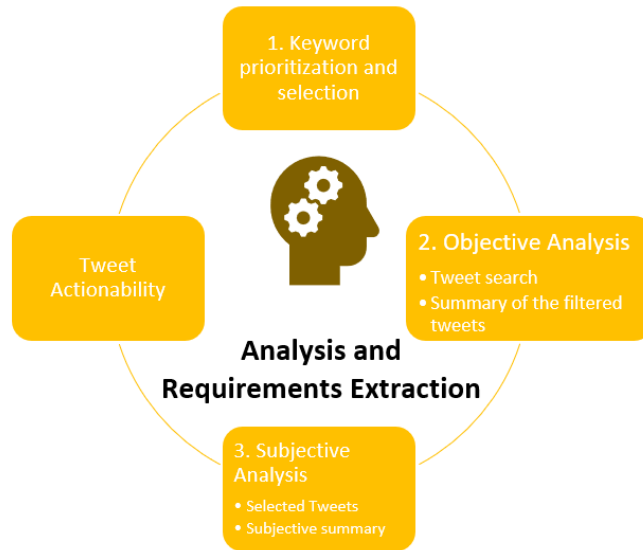
4.9 Analysis and Requirements Extraction

The last step of the whole introduced methodology of extracting requirements and gold nuggets from massive textual content such as tweets is the analysis and requirements extraction phase. In this phase, we will be preparing the core and final material a field expert needs for basing their decisions on the most valuable parts of their related textual content. This was the main aim of this research to prepare and extract the most valuable and informative parts of a large textual content to be used for supporting the important decisions made by the field experts.

In this section, we will provide two analysis templates, one as an objective analysis generated with no human interception, and the other one as a subjective analysis which is authored by the field experts after examining and refining the objective analysis. These two templates are the cheat sheet and the important insights from the textual content on the decision makers to base their decisions on them. Both templates follow the same two-part scheme, one is the most valuable and actionable tweets (or sections of the text), and the other is the summary of the most important parts of the text. One pair of the mentioned templates will be generated per the important theme of discussion (which will be defined in Sub-section 4.9).

This section describes the fifth and the last part of the pipeline (Analysis and Requirements Extraction) visualized in Figure 4.1 and the more comprehensive diagram of this section completing the previously mentioned overall diagram could be seen in Figure 4.10. The output of this step is keywords representing the content and the theme of discussion for each cluster as demonstrated in Figure 4.1.

Figure 4.10: Diagram for the fifth and the last part (Analysis and Requirements Extraction) from Figure 4.1



Keyword prioritization and Important Theme Selection

Up to this point of the proposed methodology pipeline, every single step was performed objectively using state-of-the-art machine learning and natural language processing techniques. We have extracted the different themes of discussion, and the most related and representative sections (or tweets) from the initial text corpus corresponding to each cluster or discussion theme. It is only understandable that not all of the themes of discussion bring value to the decision-making process within the subject under study. Thus, in the analysis step which is also the last step of the pipeline, we should select those themes of discussion that are informative, valuable, and could be helpful to the decisions that are to be made based on the given dataset. The act of selecting the informative clusters cannot be done completely objectively and needs to be done by the decision-makers and

the field experts. Because it is only the decision-makers who understand the connection between the different parts and themes of discussion among the dataset and the subject and the application of the decisions they are about to make based on the dataset.

To ask for the selection of the clusters from the field experts systematically, we first use KeyBERT to extract 5 keywords per 3 different types of keywords as described in Section 4.8 for all of the clusters to represent the content and the theme of each cluster using 15 keywords (per cluster). Then we list all the keywords for all the clusters in an excel sheet and provide them to the field expert and ask them to read all the keywords for all the clusters and rate them in terms of how valuable they are for their decision-making process. We repeat the same process for multiple field experts to partially eliminate the subjectivity of a single field expert and to prevent missing any important information from further analysis in the following sub-sections. A sample of a resulting excel sheet document with highlighted keywords (only 1-word keywords) by the field experts of Suncor¹⁵ in the empirical study described in Section 5.2 could be seen in Figure 4.11. The value of

Figure 4.11: Sample highlighted keywords by Suncor field experts.

1	2	3	4	5	6	7	8	9	10
(jobemployees', 0.3445)	(appleid', 0.4079)	(remotework', 0.4616)	(workchild', 0.4063)	(remotework', 0.6063)	(remoteworkthefuture', 0.4723)	(podcast', 0.4272)	(worklifebalanceest', 0.61)	(remotework', 0.4807)	(remoteworkplace', 0.5137)
(socialising', 0.3308)	(biometricbased', 0.3796)	(remotework', 0.4524)	(dadid', 0.3178)	(remoteworkplace', 0.5691)	(telecommuting', 0.4376)	(remotework', 0.4144)	(worklifebalance', 0.6286)	(collaborative', 0.437)	(remotework', 0.4779)
(socialize', 0.3237)	(biometric', 0.3847)	(onlinegroup', 0.3487)	(esierid', 0.3068)	(remotework', 0.4281)	(worklifebalance', 0.4133)	(worklifebalance', 0.4133)	(worklifebalance', 0.4133)	(collaborating', 0.426)	(telecommute', 0.3989)
(remoteworkadvice', 0.3071)	(passportid', 0.3126)	(onlinegroup', 0.421)	(adopt', 0.3043)	(remoteworktherapy', 0.5247)	(singapore', 0.3909)	(podcasts', 0.4058)	(lifebalance', 0.5571)	(collaborating', 0.4227)	(telecommuted', 0.39)
(workforce', 0.3003)	(biometrics', 0.3194)	(remotecompany', 0.3932)	(parental', 0.2996)	(remotecollaboration', 0.5176)	(remotefuture', 0.3894)	(worklifebalance', 0.3937)	(worklifebalance', 0.5158)	(workshops', 0.4148)	(remotely', 0.3866)

each keyword is shown using different shades of coloring, from red (not important) to green (important and valuable). In this example, the content related to "collaboration", "online identification", and "remote workplace" are more valuable than others in the given sheet. And thus clusters 2, 5, 9, and 10 are marked for further analysis and extracting requirements and

¹⁵<https://www.suncor.com/>

information. We keep the clusters with valuable keywords along with their own specific selected keywords by the field experts for further analysis and generating analysis and report templates in the following sub-sections.

Objective Analysis

The aim of the objective analysis is to prepare an analysis template for each to demonstrate and showcase the most valuable parts of the selected clusters in the previous sub-section (4.9) for basing the decision-making process. The objective analysis template is prepared completely objectively by NLP models and is comprised of two main parts. The first part is the selection of the tweets that contain the highlighted during the keyword prioritization step at Section 4.9. Such that we gather the main keywords that are marked as valuable to the decision-making process by the field experts and do a keyword search on the root of their words. As an example, as could be seen in Figure 4.11, there are multiple keywords highlighted, such as "collaborativenw", "collaborators", "collaborating", and "collaboration innovation". If we wish to do a simple keyword search and gather all the tweets that contain all the mentioned variations of the word "collaboration" we should break them to a simpler subword or root of those keywords that is shared among all of them, and that subword in this example is "collaborat". Converting all the mentioned keywords to the shared root (that may not be meaningful alone) is called stemming. Therefore, we apply stemming to all the keywords before performing a keyword search within the tweets of a cluster. The resulting tweets are the tweets that contain some variation of the stemmed keyword and are the reason those keywords have been

extracted from the cluster. A sample result for gathering all the tweets that contain any variations of the keywords "collaboration" and "hybrid OR remote work" are shown in Figure 4.12.

Figure 4.12: Sample tweets containing different variations of the keywords "collaboration" and "hybrid OR remote work" in the Suncor project 4.11

- 66% of employees expect to work from different locations post-pandemic, but 68% didn't have a clear implementation plan to embrace remote and hybrid work. So we collaborated with @CIO_CAN to compile and address top drivers of change <https://t.co/MPc9lqY28O>
- The paradox of hybrid work: while remote work is fine for plowing through day-to-day work, it has the potential to put a serious damper on collaboration and innovation long-term.
- Transitioning from a remote work model to a hybrid one, poses multiple challenges that many CIOs are not able to effectively address. Therefore, it is crucial that they collaborate with their counterparts to develop strategies to ease this adaptation. <https://t.co/ES9yzltg8B> <https://t.co/XRI4DePfld>

For the second part of the objective analysis, we gather all the filtered tweets using a stemmed keyword search and use the extractive text summarizer described in Section 2.3 to summarize them and represent as an objectively generated gold nugget of information that includes the summary of the tweets that are filtered 4 times throughout the methodology and are the most important parts of the whole dataset to read for making decisions. A sample of the second part of the objective analysis (the summary of the selected tweets) can be seen in Figure 4.13.

Figure 4.13: Sample summary of the tweets containing different variations of the keywords "collaboration" and "hybrid OR remote work" in the Suncor project 4.11

noticed a decrease in the innovation and collaboration coming from your employees its tempting to blame remote or hybrid work but the real reason could be very different . why hybrid work might not be the future of office culture times now hybrid work enables the employees to focus on collaboration and team building with their coladd your highlights #remote. how does a hybrid work environment allow us to build products faster work collaboratively and thrive with a remote culture hear from our ceo amp founder mario ciabarra in this entrepreneur article to learn more on building a successful hybrid workplace . as many are returning to the office agencies are creating collaborative and inclusive environments for employees how can tech bridge hybrid work amp create productive environments learn how helps facilitate a futureready workplace . online whiteboarding platform unveils new tools to strengthen hybrid work these are tools that will enable developers to integrate it with other applications . ctxs msft citrix systems collaborates with microsoft for simplifying transition to hybrid work

The two parts (the filtered tweets and the summary of the filtered tweets) together created the objective analysis. The objective analysis contains the most important and the summary of the most valuable parts of the whole initial dataset that is generated objectively and could be used to easily access the pivotal parts of the textual content and make data-supported decisions based on them.

Subjective Analysis

In a decision-making process, there has to be a human expert at the end of any pipeline to read the analyses and come up with the final conclusion and analysis, and make the final decisions. The subjective analysis is the **very**

final part of the whole pipeline where one or more decision-makers and field experts read the objective analyses generated in the previous section and generate their own analysis and conclusion. We propose a similar template as the objective analysis for the subjective analysis. Where the field experts read the tweets from the first section of the objective analysis and cherry-pick the tweets that seem more informative and valuable to them regarding their own decision-making process. They create the first part of the subjective analysis by manually filtering the very few but important tweets selected in the objective analysis template. Thus the remaining parts (or tweets) are the parts of the textual content that directly impact the decision-making process and the decisions to be made.

And as for the second part, the field experts read the summary as well and judging by the tweets selection process and the summary in the objective analysis template, they write their own perceptions and analysis of the provided material representing them as the second part of the subjective analysis. This two-part subjective analysis will be used by the company (or the rest of the decision-makers) as another main source of making data-supported decisions regarding the subject under study.

Conclusions of the Analysis

Initially, all the organizations have very large textual data that they need to process and use to make data-supported decisions. Here during the process of this methodology, we made sure to extract the most important and valuable parts of the large textual content and present them as the objective and subjective analysis of the textual content. The size of the final analyses to be read by the decision makers to make data-supported decisions is extremely

smaller and more effective than the original dataset. Therefore, all the steps are automatic and even the final analyses are designed to be extremely short and informative to be read and digested by the decision-makers who wish to make their data-supported decisions based on their initial dataset that is very large.

Chapter 5

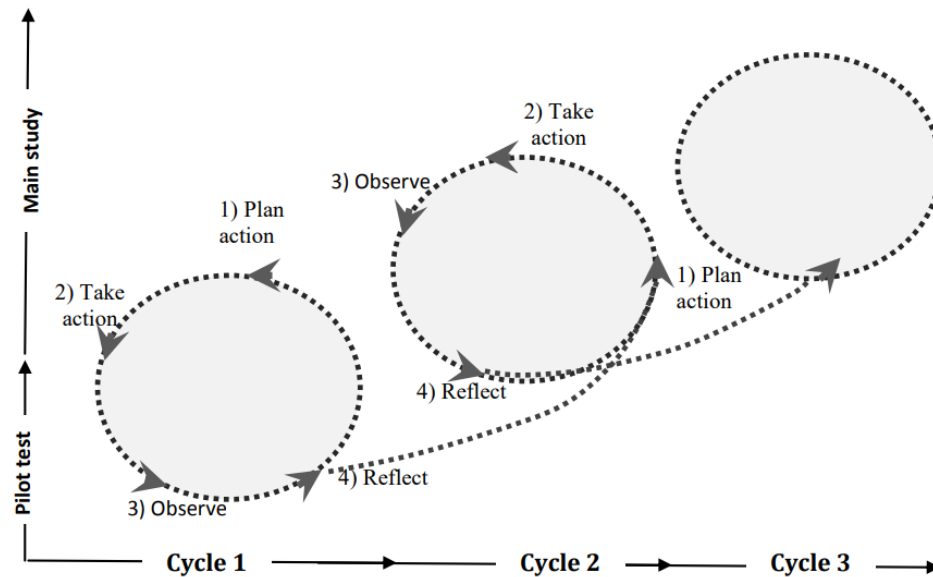
Empirical Studies

The methodology for DeKoReMi explained in chapter 4 has various applications. Any organization that has large much textual content that needs to analyze and extract certain information from to make data-supported decisions or requires understanding what the important discussions around certain topics in social media or any communication data (such as intra-company communications) are could apply DeKoReMi on their textual data to extract the gold nuggets of information and the requirements that could be used to be the base and support for their decisions. The DeKoReMi is also extremely flexible and could be adapted to many different types of applications by adjusting its steps accordingly.

The development of DeKoReMi has been done using the "**Action Research**" methodology. Action Research is a simultaneous process of doing research and taking action in real-life projects taking feedback from each other in an iterative approach to develop and improve the methodology over time through applying the research (or taking action), refining the research, and then re-applying them in an iterative feedback loop ([Reason](#)

and Bradbury (2001)) (as depicted in Figure 5.1).

Figure 5.1: The iterative cycle of action research (Hur et al. (2013))



The methodology pipeline of DeKoReMi has been developed using this methodology over the course of three different real-life industrial-academic projects in collaboration with the City of Calgary and Suncor Energy. Each project contributes to different sub-sections of the methodology and each sub-section has been fine-tuned using different empirical investigations. In this chapter, these three projects and the complementing empirical investigations that played integral roles in developing DeKoReMi will be discussed in detail.

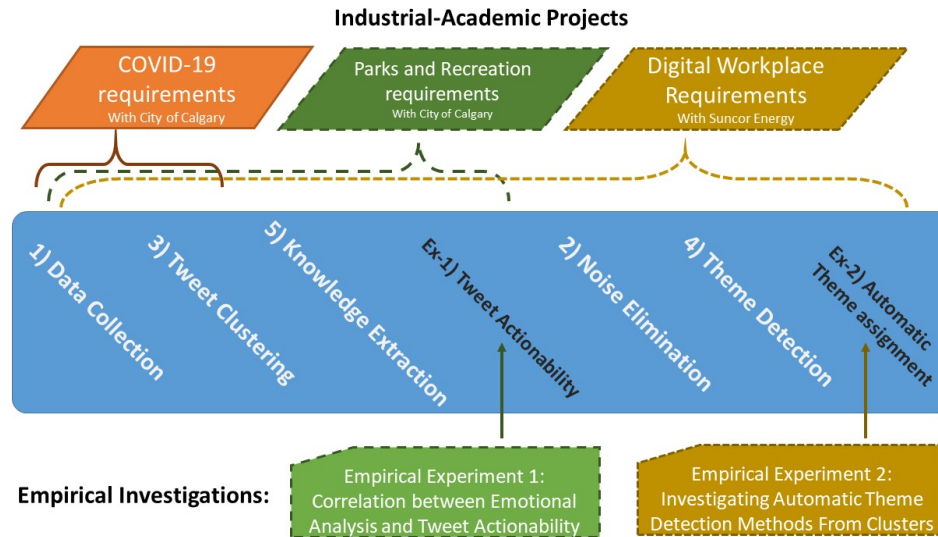
This chapter consists of three main sections. In Section "Study Design" 5.1, we will discuss the three main projects we employed and finished to develop the methodology through Action Research, and how each contributed to the development of DeKoReMi. In the second section, proof of concept

5.2, we will explain the last project in detail using the DeKoReMi notation explained in the Methodology chapter 4 as the proof of concept and showcasing the full application of the methodology and the related results and evaluations. Lastly, in section Empirical Experiments ??, we discuss three empirical investigations conducted during the development of DeKoReMi to adjust and perfect three different steps in the methodology pipeline, each conducted during one of the projects that are described in Section 5.1.

5.1 Study Design

As mentioned before, the methodology pipeline of DeKoReMi has been developed using the Action Research methodology while performing three different real-life projects. Each project has a different context, data, expected results, and collaborators that will be explained in this section. The first two projects have been done in collaboration with different municipal organizations in the city of Calgary, and the third one was completed working with the field experts from Suncor Energy company. An overall process of the development of the DeKoReMi methodology pipeline using "Action Research" over the three real-life industrial-academic projects and two empirical investigations has been depicted in Figure 5.2. Figure 5.2 shows how DeKoReMi has been developed over the course of the previously mentioned projects and empirical experiments, and how each project and empirical investigation contributed to the development of each step in the methodology pipeline of DeKoReMi. Figure 5.2 does not necessarily display the steps in ascending order, since the order of the steps in the methodology chapter was not drafted in chronological order, and the chronological de-

Figure 5.2: The development process of DeKoReMi using action research over the industrial-academic projects and empirical investigations.



velopment of the steps is illustrated from left to right in Figure 5.2. Hereby we explain the projects and how they contributed to the development of DeKoReMi.

Project: COVID-19 with City of Calgary

Context

This project was started in the middle of the COVID-19 pandemic (Jan 2021) when there were new requirements and challenges created by the citizens every day. And the city services in the city of Calgary had a difficult time locating and keeping track of people's newly developed needs and challenges and were aiming to find the best city services to invest in to improve people's lives aiming to have higher Return on Investment (ROI).

One of the main sources to locate people's challenges and concerns regarding the COVID-19 pandemic is Twitter as people mostly express them-

selves along with their challenges by tweeting. The city of Calgary formerly had methods to analyze Twitter, but they were not as effective in a fast-changing environment and situation like COVID-19. We aimed to develop a semi-automatic method to extract different and new topics of discussion among the tweets related to COVID-19 and assign tweets to the extracted topics and requirements.

Data

The dataset used for this project was about 7 million tweets that are related to the pandemic and COVID-19 published in Alberta. The period of time in which the tweets were retrieved is the period of the first three COVID-19 waves in Alberta, which is from July 2020 until Jan 2022, as could be seen in Figure 5.3.

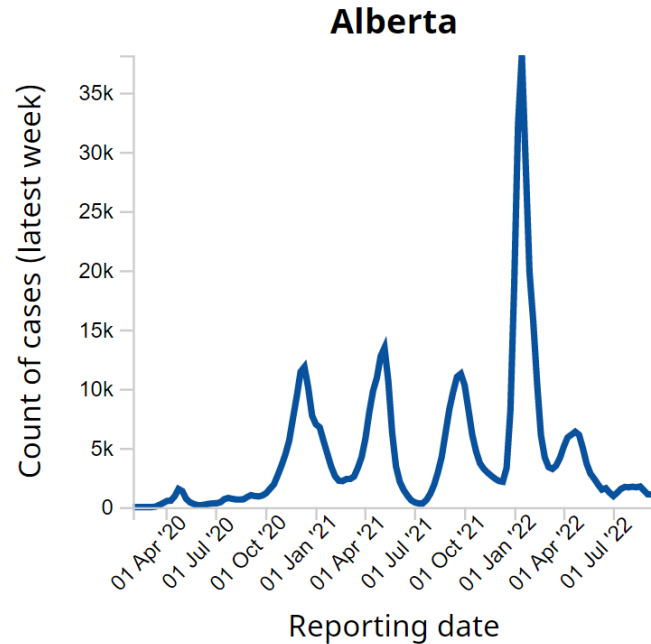
Expected Results

The goal of this project is to extract different subjects of discussion among the related tweets to COVID-19. The extracted subjects of discussion (or clusters of tweets) will further be mapped to different city services by the city of Calgary that will affect the investments in the city services to help address people's new challenges during COVID-19 and improve citizens' lives.

Contribution to the Methodology Pipeline of DeKoReMi

During this project, the foundation and the first steps of the DeKoReMi have been developed. Which are namely the data collection (section 4.3), tweet embedding (section 4.6), and tweet clustering (section 4.7). To extract

Figure 5.3: The COVID-19 new cases rate in Alberta (source: <https://health-infobase.canada.ca/covid-19/>).



knowledge and assign subjects of discussion to the clusters, (which at the time required manual work), the automatic cluster summarization (section 4.8) and top tweet extraction (section 4.8) has been developed and used in this project which are two of the four theme assignment methods mentioned in section 4.8.

Project: Parks and Recreation with City of Calgary

Context

This project is a second version continuing the contribution with The City of Calgary from the previous project for requirement extraction from Twitter. The subject under study for this project is all the related tweets to the different services that fall under the parks and recreation topic, such as parks,

roads, pedestrian pathways, and bike pathways. The aim of this study, similar to the previous project, was to elicit the public's requirements regarding the recreational features of the city to hand it to the proper authorities that could address those extracted requirements.

The results of this project address the "**RQ2: How effective is DeKoReMi in extracting themes of discussion and assigning tweets to them?**". During this industrial project, we provide the result of the manual evaluation of the clusters in terms of how close are the assigned subjects to human perception and what portion of the tweets are semantically similar to each other within clusters. The results are discussed in Sub-Section 5.1.

Data

The dataset used for this research is all the tweets related to the parks and recreation inside The City of Calgary. The search query used to gather the tweets is generated using the notation given in section 4.3 and is: *"(((parks OR "open spaces" OR "public spaces" OR "open space" OR "sidewalks" OR pathways OR bike) (calgary OR #calgary OR #yyc)) OR (#yycwalk OR #calgaryparks OR #yycbike OR #yycparks OR #yycpathways OR #yyccycle OR #calgarybike OR #yycplan OR #yycliving OR #yycroads)) -is:retweet "*. This query tries to collect all the tweets that contain the related keywords to parks and recreation system which are marked to be published in Calgary. The resulting dataset is gathered between March 9th, 2021 and March 9th 2022 which is about 1 million tweets.

Validation of Results

(This Sub Section discusses the quantitative evaluation of the accuracy of tweet clustering and theme assignment in the DeKoReMi methodology which addresses and answers the **RQ2** as mentioned in Section 1.2)

The initial dataset is distributed into 17 groups by applying the DeKoReMi methodology to group the tweets and analyze their themes of discussion. As mentioned in section 5.2, the themes of discussion (or the tweets clusters) that are valuable to the subject under study need to be selected by the domain experts. The summaries generated in section 4.8 and the keywords generated in section 4.8 were presented to the field experts from the city of Calgary for them to select the informative and valuable clusters for further analysis and validation. The selected subjects of discussion by the field experts each representing one cluster are as follows: *"Allowing drinking in parks"*, *"pedestrian and pathways safety"*, *"bike lanes"*.

The validation of clusters means that the tweets in the cluster are related to the assigned subject which should be validated manually by the field experts. To eliminate the subjectivity from the validation, three field experts from the city of Calgary were involved in the validation process. They read the tweets in the selected clusters and manually annotated them as 'related' and 'not related to the assigned topic'. The validation results are demonstrated in Table 5.1.

Discussion of the Results

To explain the Table 5.1, each column represents the percentage of tweets that were marked as 'related' to the assigned cluster them by at least 1, 2,

Cluster subject	at least 1 expert	at lease 2 experts	all 3 experts
Allowing drinking in parks	99%	99%	92.7%
Pedestrian and pathways safety	83.9%	75%	41%
Bike lanes	91%	70%	40.3%

Table 5.1: Validation of the selected clusters by three domain experts from the city of Calgary.

and all 3 of the domain experts respectively. To understand the numbers in Table 5.1, as for the first row, 92.7% of the tweets in the first cluster were labeled as 'related' by all three of the domain experts meaning that the tweets marked by all annotators are extremely semantically related to each other and the assigned subject of discussion to the cluster. Similarly, for the second row (or cluster) 41% of the tweets were labeled as 'related' by all of the domain experts, 34% were marked by 2 of the domain experts (or 75% by at least 2 domain experts) as 'related', which is a good indicator that 75% of the tweets are 'related' to each other and the assigned topic. To understand the acceptability of the numbers as effective, we could take a look at the classification results from table 5.4 which were done supervised and are between 75% and 82%. Whereas this task was done completely unsupervised without training data and the validation rate is between 70% and 99% which is relatively impressive!

This means that the DeKoReMi has successfully broken the clusters into groups that share semantically similar themes of discussion.

Contribution to the Methodology Pipeline of DeKoReMi

During this project with the city of Calgary, the methodology steps from section 4.6 were significantly improved, and the methods for assigning themes

of discussion to clusters mentioned in 4.8 were fine-tuned and tested in a larger scale and scientific setup. Also, the empirical experiment explained in 5.4 and 5.3 which are about the correlation between the emotional analysis and the actionability of the tweets were developed and conducted.

5.2 DeKoReMi - Proof of Concept Analysis

Project Description and Outline

Due to COVID-19, many companies have been forced to move away from working in complete person for all employees toward online and hybrid working and collaborating. The situation still persists even after eliminating the strict restrictions caused by COVID-19 after seeing the many benefits that online or hybrid working brings on to the table, such as being cost-efficient, saving office space for many others, and being able to hire employees from cities and even countries far from the main company. But the hybrid type of work brings many limitations and difficulties in managing the company and effectively collaborating with the peers of a company. Especially the large ones that have been struggling to accommodate themselves to the new requirements that arise from the new hybrid type of working for their employees.

Suncor Energy is a Calgary, Canada-based company that specializes in the production of synthetic crude from oil sands¹. Suncor is a large company having more than hundreds of thousands of employees that aims to extract the requirements of their employees' digital workplace environment to

¹https://en.wikipedia.org/wiki/Suncor_Energy

improve their digital workplace lives. Initially, the company was aiming to process all their textual documents and intra-company communications to locate and extract the employees' digital workplace requirements from their own textual content collaborating with us, but due to unresolved privacy issues, we pivoted to retrieving and analyzing related tweets and extracting publicly published requirements related to "digital workplace". Throughout the course of this research, we work with two of the field experts from Suncor to get feedback and evaluate the results during the project.

The subject under study in Suncor as mentioned before is "digital workplace". Digital workplace is referred to any online interaction that the employees have with each other, either working in person, online, or hybrid, such as communication systems, online meetings, remote collaborations, and online working environments. Thus we aim to extract the requirements that are addressed in the given textual documents that are concerned with the quality of the digital workplace of Suncor's employees.

This section provides a detailed description of "**RQ1: How effectively does the DeKoReMi methodology perform in an industrial setup, how helpful is the resulting outcome, and what are the main perceived benefits by them?**". As mentioned in Section 1.2, addressing this research question requires applying the DeKoReMi methodology pipeline in an industrial setup in a real-life industrial project. Here we apply the methodology on the described project conducted collaborating with Suncor Energy and provide the qualitative feedback results from the domain experts in section 5.2.

Data Collection

We apply the data collection steps as described in section 4.3. First, we come up with the related keywords, then we use them to generate the queries to target the related tweets, and finally, we collect the tweets from Twitter.

Initial Keywords

As mentioned in Sub-section 4.3, to come up with the initial keywords to search for the related tweets, we must consult with the field experts and ask them to brainstorm and list the keywords to their own subject under study that might target any related tweets. After consulting with the field experts, we generated a total of 75 keywords. Then we have classified the keywords into three classes, the "important keywords" that directly target the related tweets, the "compound keywords" that would target the related tweets if combined with other related keywords, and the "redundant" that are either not as effective as the first two classes, or they are secondary to the previous keywords and will catch many unrelated tweets even though they might seem related. The keywords in the first two classes could be seen in Table 5.2.

Generated Queries

According to section 4.3, now that we have the initial keywords for fetching the tweets, we should generate the queries combining the keywords using the Twitter API v2² query notation according to the query template mentioned in section 4.3. The generated queries and their specific time period

²<https://developer.twitter.com/en/docs/twitter-api/tweets/search/integrate/build-a-query>

important keywords	compound keywords
Digital Work Augmentation Drone / Robots Remote Work Hybrid Work Work-life balance Digital Work Monitoring Digital Experience Monitoring Smart Workspace Digital Work Context Digital Work Environment Physical work environment Digital workplace	Privacy Gamification Cubes (or Workcubes) Portal Meetings Focus Attention Interruption Video Quality / Compression / Face Recognition / Focus Video Conferences Design -> UX Problems Whiteboard, story board, wireframe Audio / Noise Cancelling / Resolution / Compression Stutter Jitter device related problems, like PC, tablet, smartphone Task Prioritization

Table 5.2: Initial keywords for gathering the tweets related to the Suncor project.

search limitations for the creation date of the tweets are mentioned in Table

5.3. As could be seen from the drafted search queries in Table 5.3, many of

Search Query	Time limit
((digital OR smart) work (place OR environment)) OR ((online OR hybrid OR remote) (work)) OR ("on-line meeting")) -is:retweet	between 2022-01-01 and 2022-06-19
(work life balance OR remote work) -is:retweet	between 2021-06-19 and 2022-06-19

Table 5.3: Search queries generated for fetching related tweets to the subject under study of "digital workplace" for the Suncor project.

the keywords mentioned in Table 5.2 are covered by the permutation of the keywords in the generated search queries by logical operators. For example, a part of the first query in Table 5.3 is *(digital OR smart) work (place OR*

environment) which covers all these keywords: digital workplace, digital work environment, smart workplace, smart work environment. Similarly for other parts of both queries. The first query is generated mostly to target specific important keywords as mentioned in Table 5.2 and the second query is generated to target general tweets around the subject of balance in the digital workplace. The first query fetches the matching queries for 6 months, whereas the second query fetches the matching tweets for 1 year since the first query targets a much more comprehensive number of tweets and would have resulted in more than 5 million tweets for 1 year.

Collected Data

We use the Software Development Kit (SDK) that we developed³ for fetching large amounts of tweets from Twitter API to fetch all the tweets that match the queries drafted in Table 5.3. The first query resulted in 2,083,033 tweets, and the second query resulted in 1,103,911 tweets within their pre-defined periods of time. This makes a total of **3,186,944 tweets** in total for this research as the initial dataset.

Of course, many more queries could have been generated using the given keywords and the duration of the search queries could have been longer, but a total of about 3 million tweets is enough for the purpose of this research.

³<https://github.com/mammalofski/Twitter-Scraper>

Noise Elimination

Following the steps mentioned in section 4.4, the process of eliminating non-informative tweets from the initial dataset gathered in the previous section comprises three steps, namely data annotation, model selection, and noise elimination.

Data Annotation

To annotate the training and evaluation data for building a noise elimination model, we selected a random sample of 6,000 tweets from the initial dataset and labeled them manually thanks to the two field experts from Suncor Energy, and four interns at the SEDS lab (every intern was handling 1,000 tweets). The labeling consists of two labels, namely informative and non-informative. The definition of informativeness was delivered by the field experts to the rest of the annotators to unify the understanding of the context while labeling the training dataset.

The final annotated data consists of **78% negative labels** (non-informative tweets) and **22% positive labels** (informative tweets). As could be understood from the distribution of the tweets in the annotated dataset in terms of being informative and non-informative, the data is extremely biased towards the non-informative tweets. This will impact the accuracy of the final trained model in detecting the informative tweets vs non-informative tweets, but as we will explain in the next sub-section, it will be to our benefit.

Model Selection

As mentioned in section 4.4 in the methodology, we cannot predict what classification model will outperform others unless we test all of them. The tested classification models in this research as proposed in section 4.4 are the combination of two vectorization techniques namely TF-IDF (section 2.2) and BERT (section 2.2) with two classification models namely Naive Bayes, Random Forrest, and the BERT’s built-in text classification model (2.1). We used the annotated data from section 5.2 to train all 5 previously mentioned text classification models. For each classification model, we select 80% of the annotated data for training and the other 20% for validation. The accuracy, recall, and F1 score of the mentioned classification models could be seen in Table 5.4.

Classification model	accuracy	precision	recall	F1-score
TD-IDF + Naive Bayes	76.9%	77%	100%	87%
TD-IDF + Random Forest	78.3%	79%	98%	87%
BERT + Naive Bayes	76.8%	77%	100%	87%
BERT + Random Forest	77.1%	77%	99%	87%
BERT built in classification	63.2%	-	-	-

Table 5.4: Validation results for five classification models over training data.

As could be seen from the validation results of Table 5.4, the classification model with the combination of TF-IDF and Random Forest has the highest accuracy score among all the classification models. Thus, we train this classification model again, using 100% of the annotated data and we apply the trained model on all the initially gathered tweets to remove all the non-informative tweets. After applying the trained classification model, **230,572 were labeled as informative**, and others were labeled as non-informative.

This means that **only 7.2%** of the tweets were labeled as informative, and the other 92.8% were labeled as non-informative. This imbalance was predictable due to the similar imbalance noticeable in the annotated data for model training.

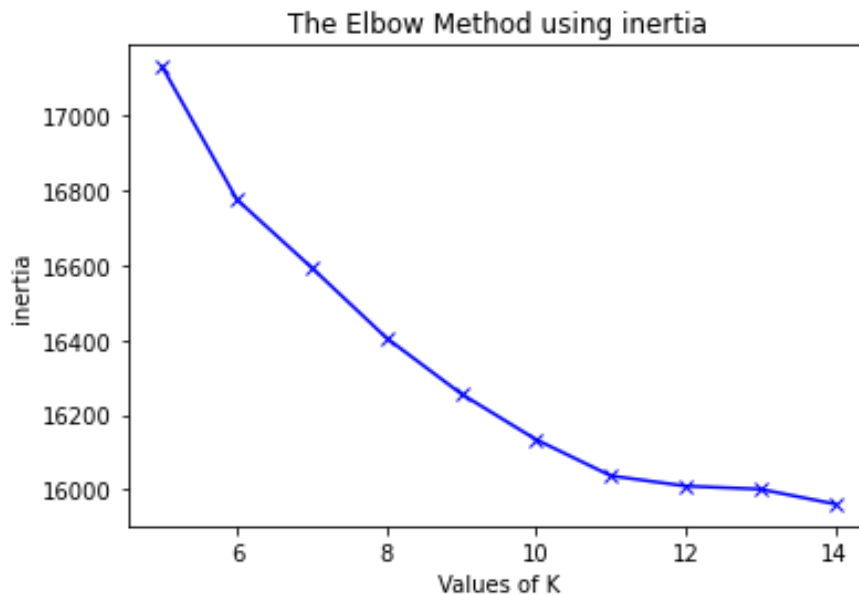
There is a threat to the validity of this classification model, which is the bias and the non-balance distribution of the informative vs non-informative tweets in the annotated data. As mentioned in section 5.2, this bias works to our benefit, because the model will have more data to learn the pattern of the negative (non-informative) tweets, and will remove the non-informative elements with high accuracy (referring to the negative recall on Table 5.4). And thus, we are sure that almost all of the removed tweets are in fact, non-informative and no informative tweets are removed. The downside is that there still may be some tweets that might be non-informative, but have not been eliminated from the dataset which does not cause any threat to the validity of the results, since we have not eliminated any informative tweets in the noise elimination phase.

Clustering

After eliminating the non-informative tweets from the initial dataset of tweets, we are left with a subset of (230,572) tweets that are informative. According to the DeKoReMi methodology, now we should cluster the remaining tweets into groups of tweets that are semantically related and close to each other. As mentioned in section 4.7, the clustering process consists of two main parts. Firstly, we use Sentence-BERT with the 'all-mpnet-base-v2' pre-trained model to generate the sentence embeddings for the tweets (or

represent them as vectors). After converting all the tweets to sentence embeddings representing the semantics of the tweets numerically, we should do the second part which is the clustering itself. For clustering, we use the elbow method which is a well-known heuristic to determine an optimal number of clusters given the dataset of sentence embeddings representing the tweets. The inertia plot of the given dataset using different cluster number values ranging between 5 and 15 is visualized in Figure 5.4. As could

Figure 5.4: The inertia plot of the sentence embeddings representing the Suncor tweets used for determining the optimal number of clusters for the elbow method.



be seen from Figure 5.4, the detected number of clusters using the elbow method for the sentence embeddings representing the Suncor tweets is 11. Therefore, we apply the K-Means method (section 4.7), with 11 as the value of K (the number of clusters), and perform the clustering. A sample of the result of the clustering could be seen in Figure 5.5. As could be seen from Figure 5.5, we have also calculated the distances of the tweets from

Figure 5.5: Sample clustering result for the tweets of the Suncor project.

text	label	cluster	distance_from_level_1_centroid
@maenad_as_hell @JJDemonic @LizzieCosmos To ma...	y	8	0.847661
Looking to improve your employees' remote coll...	y	5	0.791013
Work at a place that supports an achievable wo...	y	8	0.864902
And one thing we need to be doing anyway is RE...	y	6	0.952637
Deep sigh. Just realized all the documents I n...	y	10	0.912875
...
@uhmhailie I'd suggest looking at other's onli...	y	3	0.946108
You don't need a remote UI/UX Design? I'd like...	y	5	1.000822
i cant do much else other than do my own resea...	y	1	0.981483
i dont rily wanna go to school in the fall but...	y	8	0.994013
@TerminX13 Yeah, the boss is still a work in p...	y	4	0.966897

their corresponding cluster-centroids as mentioned in section 4.7 for future analysis.

Now, we have the informative tweets clustered in 11 different clusters each tweet belongs to one of the 11 clusters. Each cluster is supported to share semantically similar themes of discussion that will be extracted in the following steps.

Theme Assignment to Clusters

After clustering the remaining 230,572 informative tweets into 11 clusters, we should assign topics and themes of discussion to each cluster, since each of them discusses semantically similar topics within each cluster that are different from other clusters. According to the DeKoReMi, we have 4 different methods for detecting the theme of discussion in each cluster (section 4.8). As a result of the experiment 5.4 we use KeyBERT as the main method to automatically and objectively assign keywords or themes of

discussions to all the clusters. As mentioned in section 4.8, we use the features that KeyBERT provides to generate 5 keywords per 3 different types of keywords (15 keywords in total per cluster). The three-type keywords generated for all the 11 clusters are mentioned in Tables 5.5, 5.6, and 5.7.

Cluster 0	Cluster 1	Cluster 2
('hybridwork', 0.607) (('hybridworking', 0.6014) (('hybridworkforce', 0.4811) (('hybridoffice', 0.4554) (('flexibleworking', 0.4441)	('jobemployees', 0.3445) (('socialising', 0.3308) (('socialise', 0.3237) (('remoteworkadvocacy', 0.3071) (('workforce', 0.3003)	('appleid', 0.4079) (('biometricsbased', 0.37 (('biometric', 0.3467) (('passportid', 0.3326) (('biometrics', 0.3194)
Cluster 3	Cluster 4	Cluster 5
('remotework', 0.4616) (('remotejobs', 0.4524) (('onlinejoin', 0.4347) (('onlinejob', 0.42) (('remotecomensation', 0.3997)	('workchild', 0.4063) (('daddad', 0.3178) (('easierdad', 0.3046) (('adopt', 0.3043) (('parental', 0.2996)	('remotework', 0.6063) (('remoteworkplace', 0.56 (('collaborationinnovation', (('remoteworkteracy', 0.5 (('remotecollaboration', 0.
Cluster 6	Cluster 7	Cluster 8
('remoteworkisthefuture', 0.4723) (('telecommuting', 0.4376) (('remotework', 0.4291) (('singapore', 0.3909) (('remotelife', 0.3694)	('podcast', 0.4272) (('remotework', 0.4144) (('worklifetwitter', 0.4133) (('podcasts', 0.4058) (('worklifetwitch', 0.3937)	('worklifebalancecest', 0 (('worklifebalance', 0.62 (('relaxedworklifebalance', (('lifeworkbalance', 0.55 (('worklifesleep', 0.515
Cluster 9	Cluster 10	
('remotework', 0.4807) (('collaborativenw', 0.437) (('collaborators', 0.4268) (('collaborating', 0.4227) (('workshops', 0.4148)	('remoteworkplace', 0.5137) (('remotework', 0.4779) (('telecommute', 0.3989) (('telecommuted', 0.39) (('remotely', 0.3866)	

Table 5.5: 1-word keywords generated by KeyBERT along with their scores for the 11 clusters for the Suncor project

As could be seen from the previous tables 5.5, 5.6, and 5.7, a score has also been assigned by KeyBERT to each keyword representing their relativity and the degree of representation of the whole cluster.

5. Empirical Studies

Cluster 0	Cluster 1
(‘hybrid workplace’, 0.6735) (‘hybridwork environment’, 0.6697) (‘work hybridwork’, 0.6612) (‘hybridwork collaboration’, 0.6561) (‘hybridworking today’, 0.6515)	(‘work twitter’, 0.4423) (‘work social’, 0.4312) (‘work communication’, 0.4066) (‘work drama’, 0.4025) (‘work bullying’, 0.3985)
Cluster 2	Cluster 3
(‘biometric id’, 0.4848) (‘biometric ids’, 0.4816) (‘remote id’, 0.4665) (‘apple id’, 0.4629) (‘id verification’, 0.4567)	(‘remote jobinternship’, 0.6084) (‘remote jobsthank’, 0.587) (‘remote jobs’, 0.5699) (‘work remotely’, 0.5632) (‘remote careersbusinesses’, 0.5503)
Cluster 4	Cluster 5
(‘workchild’, 0.4063) (‘child daddad’, 0.3913) (‘parent work’, 0.3897) (‘for workchild’, 0.3746) (‘remotethe trash’, 0.3568)	(‘remote productivity’, 0.6881) (‘collaboration remotework’, 0.6873) (‘remote collaboration’, 0.6513) (‘remote workwhile’, 0.651) (‘collaboration remote’, 0.6491)
Cluster 6	Cluster 7
(‘remotework savings’, 0.5692) (‘remotework life’, 0.5523) (‘remotely work’, 0.5147) (‘remote living’, 0.5031) (‘work remotely’, 0.5)	(‘podcast future’, 0.4941) (‘podcast experience’, 0.4864) (‘podcast with’, 0.4768) (‘podcast talking’, 0.4735) (‘have podcast’, 0.4731)
Cluster 8	Cluster 9
(‘worklife balancefriendly’, 0.6529) (‘worklife balancepleasing’, 0.6522) (‘worklife balance’, 0.6491) (‘worklifesleep balance’, 0.644) (‘worklife balanceive’, 0.6393)	(‘remote participation’, 0.5382) (‘remote ideaspls’, 0.5105) (‘collaborative remote’, 0.5064) (‘collaborate remote’, 0.5007) (‘fully remotework’, 0.489)
Cluster 10	
(‘remote workattitudes’, 0.5716) (‘working remotelyunfortunately’, 0.5668) (‘remotely work’, 0.5667) (‘work remotely’, 0.5631) (‘remote workworking’, 0.5586)	

Table 5.6: 3-word keywords generated by KeyBERT along with their scores for the 11 clusters for the Suncor project

5. Empirical Studies

Cluster 0	Cluster 1
('future hybrid workplace', 0.711) ('way noticed decrease', 0.0863) ('problems associated remote', 0.2273) ('make amp easy', 0.1561) ('important recapping keynote', 0.2156)	('work remote majority', 0.487) ('apologised paul ive', 0.044) ('realize twitter analysis', 0.3055) ('expressed worries layoffs', 0.4459) ('draw diff balance', 0.1287)
Cluster 2	Cluster 3
('work remote id', 0.5366) ('having appocalypse instead', 0.0251) ('care determining diff', 0.1973) ('saying gender delusion', 0.1759) ('administering account refuse', 0.1815)	('remote jobs available', 0.6256) ('balance mf 95', 0.1039) ('register id time', 0.1555) ('path opens eventually', 0.0319) ('summarizes kinds', -0.0039)
Cluster 4	Cluster 5
('remote work spend', 0.4362) ('easierdad child id', 0.4074) ('ios coredata storage', 0.3321) ('deliberation decided cancel', 0.0293) ('car functionality roll', 0.225)	('collaboration remote work', 0.7054) ('shouldnt stop running', -0.0178) ('hurt innovation say', 0.3369) ('video window incubation', 0.2077) ('smartsheetsmartsheet transforms work', -0.0192)
Cluster 6	Cluster 7
('remote living commute', 0.5762) ('increased considerably pressure', -0.0144) ('balance singapore', 0.4561) ('discussions regarding hiring', 0.157) ('means better worklife', 0.3414)	('podcast future work', 0.5572) ('control screen', 0.0757) ('sucess online relationship', 0.3575) ('people run bakery', 0.0095) ('help fulltime job', 0.3007)
Cluster 8	Cluster 9
('worklife balance ill', 0.7142) ('drive remote location', -0.1276) ('especially toughest ones', 0.0265) ('working hybrid format', 0.0929) ('theres woman', 0.2146)	('future remote work', 0.566) ('manifestkeprayers id way', 0.0898) ('royston councillors community', 0.1775) ('balance amp insecurities', 0.0496) ('refund online fellowshipmanifest', 0.2315)
Cluster 10	
('advice remote workers', 0.5919) ('year amp id', 0.1668) ('hybrid 20x harder', 0.2078) ('precovid stay valued', 0.0832) ('placements campus considering', 0.2351)	

Table 5.7: 2-word diverse keywords generated by KeyBERT along with their scores for the 11 clusters for the Suncor project

Keyword Selection by Field Experts

At this point, we have 11 clusters (from section 5.2) and 15 keywords for each cluster representing the overall theme of each cluster. It is only trivial that not all of the clusters and their themes could be valuable to the end goal of the project. There is no automatic method for determining which cluster and theme of discussion could be beneficial for further analysis and exploration for knowledge and related requirements for the subject under study for this project, which is about digital workplace as explained in section 5.2. Therefore, in this step, we consult with the domain experts from Suncor Energy to cherry-pick the valuable clusters along with their most important keywords. So we present all the keywords extracted for all the clusters (mentioned in Tables 5.5, 5.6, and 5.7) to the domain experts and ask them to select the cluster-themes that are the most valuable to them along with specific keywords from each cluster. The chosen cluster themes and the selected keywords can be seen in Table 5.8. As could be seen from Table 5.8, the clusters 0, 2, 5, 9, and 10 are chosen by Suncor's domain experts to be valuable for further exploration and analysis.

Analysis and Requirements Extraction

After selecting the valuable clusters and their corresponding important keywords, in this step, we do further analysis to extract and provide the gold nuggets of information and the related knowledge hidden in the selected clusters to the field experts. The extracted knowledge will further be used by the field experts in their Decision Support System (DSS) to be the basis for their data-supported decisions to be made.

Cluster 0	Cluster 2
('hybridwork collaboration', 0.6561) ('future hybrid workplace', 0.711) ('problems associated remote', 0.2273) ('hybridworking', 0.6014)	('remote id', 0.4665) ('work remote id', 0.5366) ('biometric id', 0.4848)
Cluster 5	Cluster 9
('collaboration remotework', 0.6873) ('remote collaboration', 0.6513) ('collaboration remote', 0.6491) ('collaborationinnovation', 0.5311) ('collaboration remote work', 0.7054)	('collaborativenw', 0.437) ('collaborators', 0.4268) ('collaborating', 0.4227) ('remote participation', 0.5382) ('remote ideasppls', 0.5105) ('collaborate remote', 0.5007) ('future remote work', 0.566)
Cluster 10	
('remoteworkplace', 0.5137) ('telecommute', 0.3989) ('hybrid 20x harder', 0.2078) ('remotework', 0.4779)	

Table 5.8: 2-word diverse keywords generated by KeyBERT along with their scores for the 11 clusters for the Suncor project

As mentioned in section 4.9, the analysis step consists of two results, namely the objective analysis and the subjective analysis. The two analyses are provided in two templates that contain the promised gold nuggets of information and knowledge to be used for decision understanding the key information that the initial dataset withholds.

The objective analysis, as described in section 4.9, is generated purely by Language Models and consists of two sub-sections. Firstly, the tweets that contain any of the roots of the stemmed version of the selected keywords in Table 5.8. And secondly, they automatically generate a summary of the filtered tweets with specific keywords. The process of generating this template is repeated for each selected cluster shown in Table 5.8, but we only mention one of them (template for cluster number 5) here to demonstrate

the resulting objectively generated template.

If we apply stemming on the selected keywords from cluster number 5 in Table 5.8 as explained in section 4.9, we will have these keywords to perform the keyword search in the tweets: "*collaborat, remote work*". So we do keywords search among all the tweets in cluster 5 that contain these search keywords. The top 30 resulting keywords could be seen in Figure 5.6.

Figure 5.6: The tweets in cluster 5 that match the searching the important keywords mentioned in Table 5.8

```
#FutureComputers highlights 3 ways @Microsoft #Teams makes remote work collaboration easier and more effective. Check out the article for more! https://t.co/jsMIs2U9Ax

#FutureComputers highlights three ways @Microsoft #Teams makes remote work collaboration easier and more effective. Check out the article for more! https://t.co/06D7rMse0G

With the increase in remote work, many of us have grown accustomed to using Teams. You might be surprised (and excited!) to learn about the 3 features we discuss in our latest blog. Check it out!

#msteams #collaboration #remotework

https://t.co/XXqbXVq9ve

Remote work has many benefits, and the tools that have arisen to enable collaboration help teams get work done and achieve success. Read More https://t.co/1V305eA5rk #businessandmanagement #itmanagement #computersecurity

It is clear that remote work is here to stay!

By promoting effective communication and #collaboration strategies, remote teams can be efficient, productive, and successful.

https://t.co/Vfw6GKZK2z

#remotework #zoom

Trusted by more than 270 million users, #MicrosoftTeams enables seamless collaboration while providing the flexibility to work from anywhere at any time. Explore @Microsoft365 and find a Microsoft Teams plan for your business: https://t.co/Sof3Av27nt

#hybridwork #hybridsolutions https://t.co/f8I79TUkxy

The same communication and collaboration platforms that allow co-workers to engage in projects and build relationships across office locations can enable teamwork for a distributed workforce too. https://t.co/trQaXP0UfX via @SWolpa #remoteworktips

Collaboration is no longer limited to being in the same room. Learn how to build a remote work schedule in Microsoft Teams that works for everyone! https://t.co/7Br4imJeq1

Looking to improve your employees' remote collaboration? Or maybe you're not getting everything you want from your cyber platform? Check out the latest solutions designed to boost #remotework productivity. 📱 https://t.co/PWkue4MgEo

The heart of remote work is a businesses communication and collaboration system - with unified communications, your team can access manageable & efficient communications features within a common interface.

Learn more: https://t.co/P00o2BUyA3 https://t.co/tmsyGVb8XM

Microsoft Warns of a Growing Problem https://t.co/S44xjU10oN

Microsoft recently studied the impact of remote work on collaboration in an effort to improve Teams. https://t.co/i978kvIzbF

#remoteworking - not ALL roses - as massive new peer-reviewed study from @Microsoft found that, while #remotework is fine for plowing through day-to-day work, it has the potential to put a serious damper on #collaboration & #innovation long-term. https://t.co/cX0InXJakh

@davidemccune @jenneraub @stewak2 @VICE Here's a very nice study from Microsoft on the effects of remote work on collaboration. https://t.co/tCVyY2vDQ4 https://t.co/v2HVLcSdQp
```

And the second part of the objective analysis template generates the

summary of all the matching tweets with the important keywords, which are shown in Figure 5.7.

Figure 5.7: The summary of the tweets in cluster 5 that match the searching the important keywords mentioned in Table 5.8

```
'highlights 3 ways makes remote work collaboration easier and more effective check out the article for more . these features can help remote workers and with their teams wherever they are read more on how to improve collaboration with on #hybridwork. is the best way to keep hybrid teams connected but dont just take our word for it check out how used workspace apps to support collaboration and worklife balance in the transition to remote work . sampath kumar arunachalam senior interactive designer acl digital discusses some of the best practices of enterprises can leverage to help workforce collaborate from remote locations enabling a work from anywhere culture . while working remotely break work down into easier bitesized tasks like the gulha peoplehums remote workforce management solution contain all the necessary tools to help teams streamline remote work collaboration click here to learn more . through the thick and thin of the pandemic the one thing that has kept all workplaces grounded is the right hr software this technology helped move teams to remote work enabled hybrid work and better collaboration communication and more . 2 junior employees tend to lose out on some training over inperson not everything is written or can be conveyed in a zoom call depending on the nature of the work being done inperson collaboration has been shown to do better than remote . creators developers contributors thinkers anyone who works with their mind online can be connected through technology web3 and the metaverse enable new modes of work collaboration amp payment.'
```

The combination of the full versions of the filtered tweets from Figure 4.12 and the summary generated in Figure 5.7 together construct the final objective analysis that will be provided to the domain experts to be used as the gold nuggets and the most valuable information that could be extracted from the initial 3M tweets fetched in the first place. Reading the provided

objective analysis will result in understanding the most valuable knowledge that is necessary to be obtained in order to make data-supported decisions.

Qualitative evaluation of the Results by the Field Experts

The final results presented and delivered to Suncor Energy as the outcome of the empirical application and evaluation of the DeKoReMi methodology pipeline are as follows:

- The full versions of the two objective and subjective analyses of the tweets pasted about "digital workplace" (Sub-Section 5.2) as stated in Sub-Section 5.2.
- The comprehensive and step by step explanation of the methodology and the required training for Suncor to apply on other textual data (as explained in Chapter 4).
- A presentation series to 8 domain experts from Suncor Energy to explain the business outcomes of the analyses, the methodology, and different applications of the DeKoReMi methodology pipeline in the industry and how to leverage DeKoReMi in the Decision Support System to analyze their textual data to make data-supported decisions.

Here are the ideas, feedback, and qualitative evaluation of the DeKoReMi and its empirical application on the tweets related to "digital workplace" according to 8 domain experts from Suncor Energy during the presentation series:

- **(Analysis Evaluation)** Even though the tweets are public opinions and non-informative by nature, the methodology was able to sort

out the different themes of discussion, the most important pieces of information for each subject and classify the tweets into different themes of information for further analysis.

- **(Analysis Evaluation)** The methodology extracted interesting subjects of discussion from the tweets and the provided analyses provided useful information about the subjects to be used as a reference for making data-supported decisions regarding the different subjects of discussion.
- **(Methodology Evaluation)** The DeKoReMi methodology pipeline is a great tool to be used to analyze any internal textual data to segregate different subjects, extract correlated requirements from the segregated subjects, and retrieve pivotal information regarding the extracted requirements to support a wide variety of decisions in different departments of the company.
- **(Methodology Evaluation)** Apart from the themes of discussion and the requirements, the methodology is able to provide useful knowledge regarding the extracted subjects and requirements explaining them and providing descriptive information about the extracted elements.
- **(Inspired Internal Application)** The methodology could be used on other sources of textual data such as individual and team communications, meeting transcriptions, intra-company emails, and other textual content from Microsoft Teams to further extract the different themes of discussion and the related requirements and the most im-

portant pieces of information hidden in the massive textual data that the company withholds.

- **(Inspired Future Work)** The referenced URLs in the valuable tweet clusters were very related and could be used for further requirements and information extraction possibly using the same methodology in a different context.

This qualitative evaluation and feedback from the domain experts from Suncor Energy also address the **RQ1** as mentioned in Section 1.2. The **RQ1** asks for the possibility of applying the developed methodology in real-life industrial setups. Moreover, the **RQ1** tries to investigate the possible outcomes of the DeKoReMi methodology in implementing it in the industry. This section (5.2) discusses the process of applying DeKoReMi in an industrial-academic project collaborating with Suncor Energy that answers the first part of the **RQ1**. And this Sub-Section addresses the second part of the research question which asks for the evaluation and the perceived benefits of the methodology by the domain experts from the industry.

5.3 Association Between Emotional Analysis and Actionability

Here in this empirical study, we investigate the correlation between emotional analysis and the actionability of the tweets. We define a tweet as being "actionable" when the tweet demonstrates a subject or preferably a concern that could be addressed by taking one or more action(s). In other words,

the ability to define an action based on a tweet is considered "actionability" and a tweet that could result in taking action is called an "actionable" tweet.

This empirical study addresses "**RQ4: How well emotional analysis help detect clusters and tweets that result in taking action?**". In this section, we investigate the degree to which tweets that are expressing different emotions affect the tweets being actionable. Which will further address the **RQ4** as mentioned in Section 1.2. (the concluding answer could be found in Sub-Section 5.3.

To conduct this empirical investigation, we first classify the tweets in two ways, being 'actionability' and 'emotion'. And then we investigate the effect of emotional analysis over actionability using two methods, namely 'probability' of a tweet being actionable if expressing certain emotions, and the 'correlation' between emotions and actionability.

Annotating tweet Actionability

In order to investigate the correlation between emotional analysis and actionability in the tweets, the tweets need to be classified in two areas. Firstly, the tweets are to be classified in terms of actionability. And secondly, the tweets are classified in terms of different expressed emotions which may include multiple classes of emotions.

For the first classification task (actionability), the tweets should be classified into two classes being 'actionable' and 'non-actionable'. Since the definition of actionability is vague and mostly depends on the subject, it is difficult to find a pre-trained classification model or even a training dataset for specific subjects. Here we selected a random subset of 2300 tweets from

the dataset gathered in Sub-Section 5.1 during the Parks and Recreation project (Section 5.1) and manually annotated them in terms of 'actionability'. The annotation was done by four summer interns⁴ who were explained the definition of 'actionability' in this specific subject by the domain experts from the city of Calgary. Together, they read all 2300 randomly selected tweets and annotated the tweets individually labeling them as 'actionable' or 'non-actionable'. To alleviate the subjectivity from the annotation results, each tweet was annotated by two students, and a tweet can only be accepted as actionable if both students have marked them to be actionable.

Classification of Emotions

For the second classification part (classifying emotions) we use the emotion classification method explained in Section 5.4 to classify the expressed emotions by the tweets in the following six classes: sadness (0), joy (1), love (2), anger (3), fear (4), surprise (5). Although, tweets may express multiple emotions simultaneously, as explained in Sub-Section 5.4, it is most probable that a tweet expresses one emotion stronger than other emotions, which is the one we classify as the expressed emotion. Using the discussed classification model, we classify the 2300 annotated tweets that are already classified in terms of 'actionability' in Sub-Section 5.3. Finally, we have 2300 tweets that are classified in two points of view, namely 2 classes for actionability, and 6 classes for expressed emotions. Using the classified dataset we investigate the 'probability' and 'correlation' of tweets being actionable according to their expressed emotion in the following sub-sections.

⁴Divyansh Rana, Anita Das, Reeshad Faiyaz, and Kirtan Kakadiya

Investigating the Probability of Actionability Based on Emotions

As mentioned in the description of this empirical study in Section 5.3, we investigate the effect of emotional analysis over tweet actionability using two methods. The first one is "calculating the probability of a tweet being actionable, if expressing certain emotion". In other words, we calculate the probability of the actionability of a tweet based on and for each emotion using the following equation 5.1:

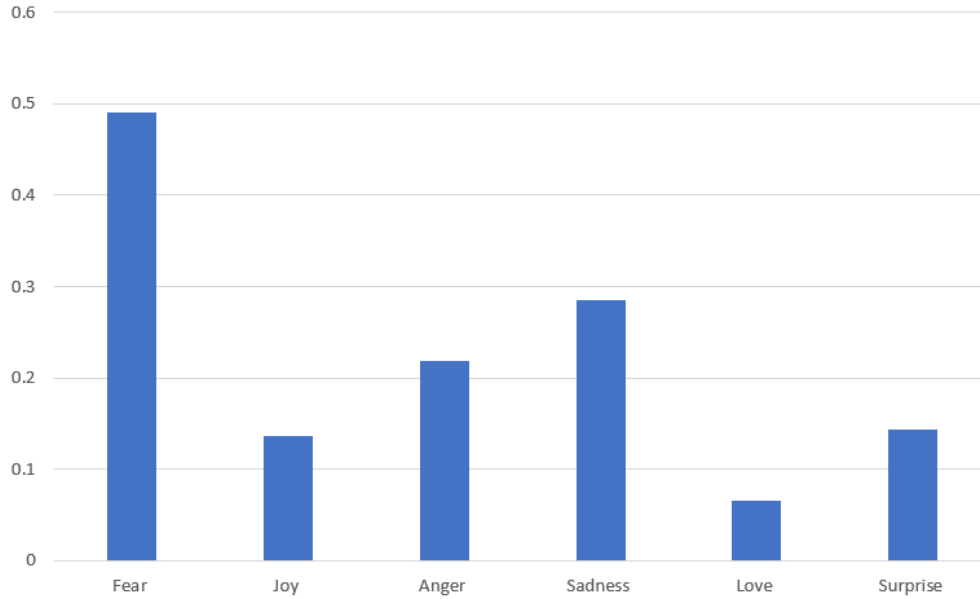
$$P(A|E_i) = \frac{P(A \cap E_i)}{P(E_i)} \quad (5.1)$$

In which $P(A|E_i)$ is the probability of tweets being actionable (A) if they have the emotion E_i .

We applied the Equation 5.1 on all the 6 emotions mentioned in Sub-Section 5.3, and the results can be seen in Figure 5.8.

As could be seen from Figure 5.8, there is a clear gap between the probability of a tweet being actionable if it expresses 'fearful' emotion and other emotions. The probability of a tweet being actionable is 49% which is 72% higher than the second emotion with the highest probability of being actionable which is 'sadness' with 28.5%. The rest of the emotions have significantly lower probabilities of being actionable such as 'love' and 'surprise' with the probabilities of being actionable as 6% and 14% respectively. This gap between the attainability probabilities based on different emotions, specifically the 'fearful' tweets having from 1.72 times to about 7 times higher chance of being actionable is a good indication that "the expressed emotion by the tweet has a high effect in the probability of the tweet being actionable."

Figure 5.8: The result of applying the Equation 5.1 on the 2300 classified tweets to calculate the probability of tweets being actionable if expressing certain emotions.



Investigating the Correlation Between Emotions and Actionability

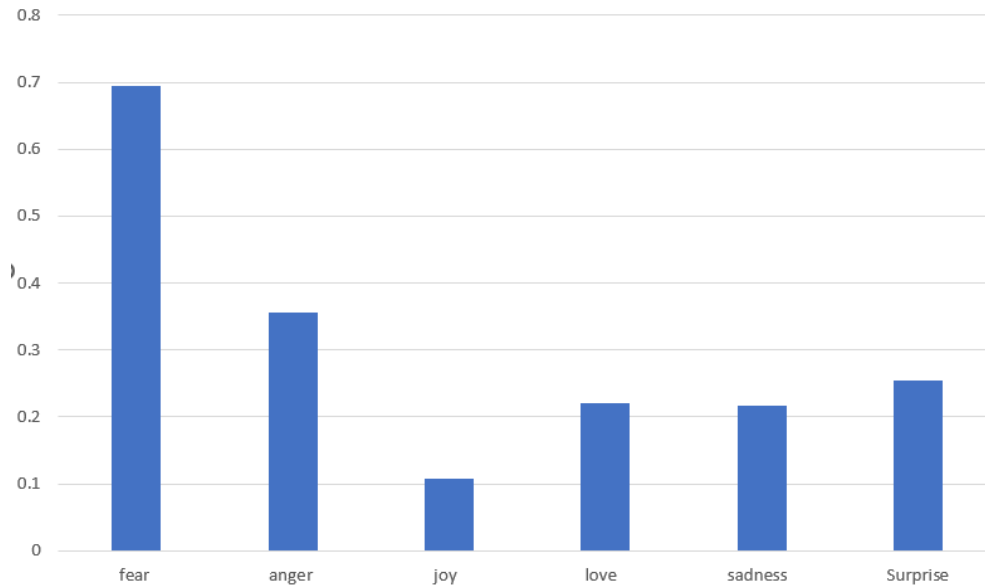
As mentioned in the description of this empirical study, we use two indicators to investigate the effect of emotional analysis on the actionability of the tweets. Here in this sub-section, we investigate the second indicator which is the "Pearson Correlation Coefficient" [Benesty et al. \(2009\)](#) between the tweet expressing different emotions and the actionability of the tweet. The Pearson correlation coefficient is calculated using the following equation:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (5.2)$$

In which x and y are the two variables with r as their correlation coefficient, x_i is the values of the x -variable in a sample, \bar{x} is the mean of values of the x -variable, and similarly for y_i and \bar{y} .

To be able to conclude any noticeable difference in the Pearson correlation coefficient of the actionability of the tweets while expressing different emotions, we calculate all the correlations between the tweet expressing different emotions and the tweet being actionable and compare them with each other. The result of the Pearson correlation coefficient (Equation 5.2) on the 2300 classified tweet dataset is depicted in Figure 5.9.

Figure 5.9: The result of applying the Equation 5.2 on the 2300 classified tweets to calculate the Pearson Correlation Coefficient between the tweet emotions and actinability



As could be observed from Figure 5.9, The correlation between the tweet expressing 'fear' emotion and the tweet being actionable is 0.694 which is a high correlation of its own. Comparing the correlation coefficients between different emotions, the fearful tweets have almost 2 times higher correlation between the second highest emotions with correlations to actionability, which is the 'anger' emotion with the Pearson correlation coefficient of 0.356. Other emotions have significantly lower correlation coefficients com-

pared to the 'fear' and secondly 'anger' correlation coefficients with tweet actionability. As the fearful correlation coefficient is from about 2 times to about 7 times higher than the tweet actionability correlation coefficients with other expressed emotions. With this clear gap between the Pearson correlation coefficients of different expressed emotions and tweet actionability, we could conclude that there is an obvious correlation between the tweet having expressed 'fearful' emotions and the tweet labeled as attainability.

Conclusions

Having experimented with the effect of emotional analysis over tweet actionability over the annotated and classified dataset of 2300 tweets using two different methods, calculating the probability of the tweets being actionable if expressing certain emotions, and applying the Pearson correlation coefficient on the different emotions between tweet actionability we resulted in the two Figures 5.8 and 5.9. Comparing the ordering of the emotions in terms of having higher correlation coefficients with actionability from Figure 5.9 and the ordering of the probabilities of the tweet being actionable if the tweet expresses certain emotions from Figure 5.8, we could see that the 'fearful' emotion has the highest order in both tables, but the second and third order exchanges between 'anger' and 'sadness' emotions. Moreover, there is a clear gap between the emotion with the highest scores (probability and correlation coefficient) which is 'fear' and other emotions. This empirically concludes that firstly, the expressed emotion in the tweets has a clear effect on the actionability of the tweet, and lastly, the tweet expressing fearful emotion has by far the highest probability for and highest correla-

tion with the tweet being actionable. This empirical study and its results addresses and answer the **RQ4** as mentioned in Section 1.2.

5.4 Comparative Analysis of Auto-assigning

Themes to Tweet Clusters

During the DeKoReMi methodology pipeline, we perform clustering on the informative tweets in Section 4.7 to break the dataset into group of tweets that are semantically similar to each other within each group. If the process of sentence embedding in Section 4.6 and clustering in section 4.7 are done correctly, the tweets of each group should share similar themes of discussion. But extracting the theme of discussion manually from a tweet cluster is time-consuming and subjective. To eliminate the manual process being expensive (both time and money) and remove the subjectivity from the process aiming to eradicate human intervention from the methodology we proposed 4 different automatic theme assignment methods in Section 4.9 to elicit the subject of discussion from the tweet clusters automatically and objectively. The theme assignment methods are namely "automatic text summarization", "Top tweets selection", "TF-IDF keyword generation", and "KeyBERT keyword generation" which are explained in detail in Section 4.9.

Here in this experiment, we investigate whether four introduced auto-theme assignment methods are effective, and which of them will result in the closest themes to human perception. This will be an empirical experiment

conducted with four summer interns⁵ as annotators and references for human perception.

This empirical experiment addresses the "**RQ5: What is the closest automatic method to human perception in assigning themes of discussions to tweet clusters in DeKoReMi?**" as explained in Section 1.2. The summary of the result of this experiment is explained in Sub-Section 5.4 as would address the answer to **RQ5**.

This empirical experiment is conducted during implementation the project collaborating with Suncor Energy as explained in detail in Section 5.2. Thus, the data and the tweets clusters used to perform this study were taken from the dataset gathered and clustered in Section 5.2.

Experiment Design

First, we define the point of reference as human perception since this empirical study aims to investigate the closest theme assignment method to human perception. If we compare the results of the four methodologies with the opinion of only one human, the comparison will be highly subjective. To eliminate the subjectivity to a good degree, we aggregate the opinion and annotation of four interns and consolidate their inputs and use it as the main point of reference for human perception.

For this experiment, we aim to set scores (from 1 to 5) for each of the four theme assignment methods representing how close they are to human perception (a higher score means semantically closer to human perception). Here are the steps to the experiment design to set relevance scores to the

⁵Divyansh Rana, Anita Das, Reeshad Faiyaz, and Kirtan Kakadiya

methods by the four annotators:

1. Select a random subset of 200 tweets from all the clusters.
2. Generate four results each for one of the four theme assignment methods.
 - a) Top 10 keywords generated by KeyBERT (using Section 4.8)
 - b) Top 10 keywords generated by TF-IDF (using Section 4.8)
 - c) Top 10 closest to the centroids of the clusters (using Section 4.8)
 - d) Automatically generated text summary (using Section 4.8)
3. Repeat this scoring system for all of the clusters:
 - a) All annotators read the 200 randomly selected tweets to get an understanding of the overall theme of the cluster.
 - b) They read and examine the results for each of the four theme assignment methods.
 - c) They set a score (from 1 to 5) to each of the four methods representing how close the results are to their perceived theme of discussion while reading the 200 tweets.
 - d) Calculate the mean of all the scores assigned by the annotators for each theme assignment method using Equation 5.3.
4. For each of the four theme assignment methods, calculate the mean score from all of the clusters using Equation 5.4. These are the final scores representing the closeness of that theme assignment method to human perception.

In short, the closeness score for each theme assignment method is the mean of all the average scores of each annotator for each cluster calculated using Equations 5.3 and 5.4.

$$mean_score_cluster(C_i, M_k) = \frac{\sum_{j=1}^{N_a} annotator_score[a_j][M_k][C_i]}{N_a} \quad (5.3)$$

In which c_i is cluster number i , M_k is one of the four theme assignment methods, and N_a is the number of annotators, which in our case is 4. Also, $annotator_score[a_j][M_k][C_i]$ is the score that annotator number j gave to method number k in cluster number i .

$$final_score(M_k) = \frac{\sum_{i=1}^{N_c} mean_score_cluster(C_i, M_k)}{N_c} \quad (5.4)$$

In which N_c is the number of total clusters, and $mean_score_cluster(C_i, M_k)$ is defined in Equation 5.3.

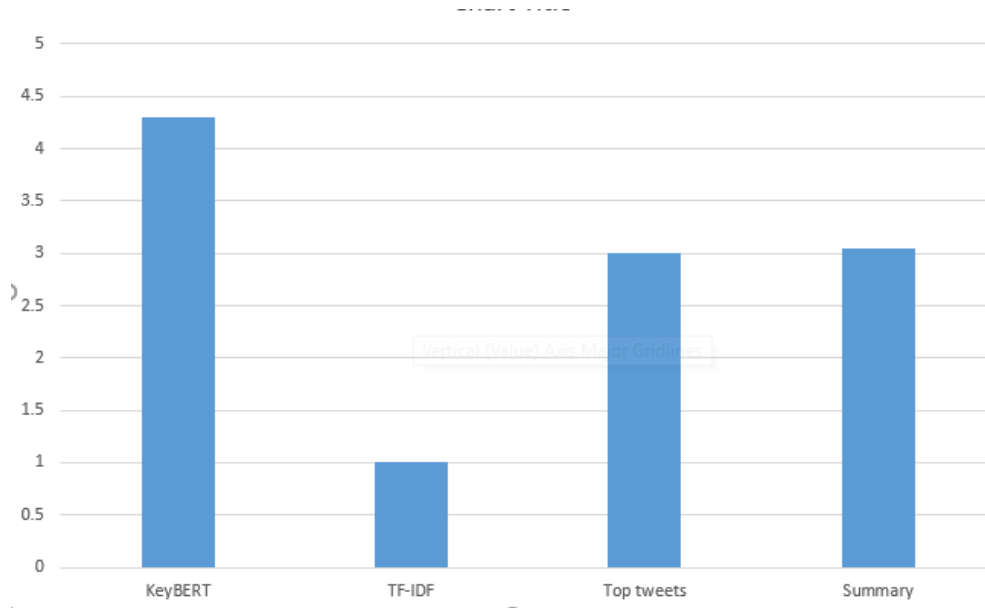
Results

After performing the experiment designed in Sub-Section 5.4 on the clusters generated in Sub-Section 5.2 in the project collaborating with Suncor Energy, the final scores calculated for the four theme assignment methods are visualized in Figure 5.10.

Conclusions

As could be understood from the results depicted in Figure 5.10, the most relevant and closest theme assignment method to human perception as

Figure 5.10: Final relevance scores for the four proposed auto theme assignment methods representing how close each result to human perception.



the result of this empirical experiment is the keywords extracted by the KeyBERT keyword generation method discussed in Section 4.8 having scored 4.3/5, and the least relevant method to human perception is the keywords generated by the TF-IDF keyword weighing method discussed in Section 4.8 having scored 1/5. And the two methods, automatic text summarization, and top tweets, generate almost equally close results to human perception (3/5 and 3.05/5 respectively).

It could be concluded from the results that KeyBERT generates keywords close to human perception, and could be used in the DeKoReMi Methodology pipeline as the main theme assignment method. The result of this empirical experiment also addresses and answers the **RQ5** as mentioned in Section 1.2.

Emotion Analysis

Tweets, similar to any text can express emotions such as joy, sadness, or fear. The emotion of the tweet or the percentage of different emotions in a cluster could be good indicators of the demanding nature of the tweet or the tweet cluster. For example, if a tweet in a cluster with the topic of 'biking pathway' is emotionally labeled as 'anger, it would convey that the author is expressing repelling opinions about biking pathways. Which is very enlightening because that tweet could be helpful in extracting the demands from a cluster of tweets. Similarly, if some clusters contain significantly higher percentages of negative emotions such as 'fear', 'anger', or 'sadness' it demonstrates that those clusters may reflect more frustration and negative attitudes than other clusters, which could prioritize them higher than the clusters with more 'joyful' tweets in terms of requirements extraction. This empirical study was conducted during the industrial project discussed in Section 5.1 which means all the tweets and clusters that are referred in this section are gathered and generated during the implementation of the parks and recreation project mentioned in Section 5.1.

Here in this Section, we investigate the effect of emotional analysis in tweets and in tweet clusters in prioritizing the tweets in a single cluster, or prioritizing some clusters over others in terms of requirements and public demand extraction. This section answers the "**RQ3: Does emotional analysis help prioritize the tweet clusters in terms of importance and urgency to be addressed?**" and demonstrates how emotional analysis could be beneficial for tweet, cluster, and requirements prioritization and the benefits are specifically discussed in Sub-Section 5.4.

Training Dataset

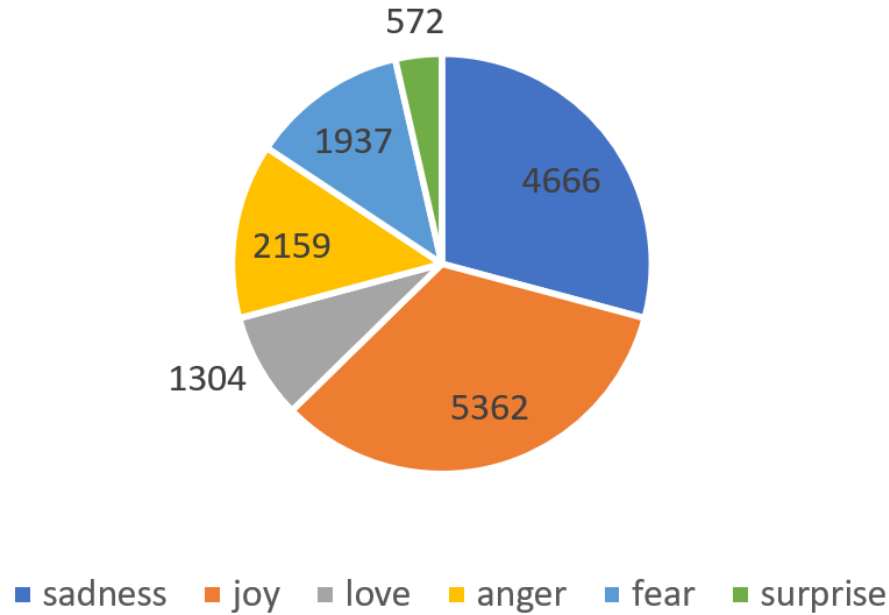
In order to label emotions on tweets, we used an open-source emotion dataset ([Saravia et al. \(2018\)](#)) to train a tweet classification model, which will be explained in this chapter. The training dataset consists of 20,000 labeled tweets. 16,000 for training, 2,000 tweets for validation, and 2,000 tweets for testing. The emotion classes in the used dataset are sadness (0), joy (1), love (2), anger (3), fear (4), and surprise (5). We chose this dataset because first, it has enough labeled tweets to build a reliable classification model over the dataset, and secondly, because the labeled emotions are various enough to polarize and analyze the demanding nature of the tweets. Meaning that we can distinguish between wishful-thinking tweets and tweets that are actually complaining about a subject. Both poles of emotions will be further useful in extracting the nature of the requirements. For example, extracting good-to-have requirements (joy), or requirements that people appreciate (love), or even requirements that people desperately demand but don't have (anger or fear). The distribution of the emotion classes over the training dataset is depicted in [Figure 5.11](#)

As could be seen in [Figure 5.11](#), the majority of the tweet emotions are of joy (33.5%) and sadness (22.1%). And the rest of the dataset consists of 13.5% anger, 13.1% fear, 8.1% love, and 3.5% surprise in descending order.

Classification Model

To classify the tweets in terms of conveying different emotions, we use the same classification method that we have used in [Section 5.4](#) to classify the tweets in terms of being positive or negative.

Figure 5.11: Distribution of emotion classes over the training dataset



Here we fine-tuned a pre-trained Bidirectional Encoder Representations from Transformers (BERT) model called ‘Distilbert’ [Sanh et al. \(2019\)](#) on the emotion dataset described in Sub Section 5.4. The resulting model assigns scores for all 6 emotions. In the end, the final emotion is the one with the highest score. Listing 1 is an example result for the sentence “I am very worried about the safety of bike lanes in this city.”.

The scores assigned to the feelings ‘fear’ and ‘anger’ are 79% and 19% respectively which are much higher than other emotion classes. We tag this sentence as ‘fear’ since it has the highest score of 79% among all the emotions.

The training was done on the training dataset described in Sub Section 5.4 using the High-Performance Computing servers provided by the University of Calgary [of Calgary](#). We used GPU-powered servers (that offers

```
1  [
2    {
3      "label": "sadness",
4      "score": 0.004
5    },
6    {
7      "label": "joy",
8      "score": 0.004
9    },
10   {
11     "label": "love",
12     "score": 0.001
13   },
14   {
15     "label": "anger",
16     "score": 0.197
17   },
18   {
19     "label": "fear",
20     "score": 0.793
21   },
22   {
23     "label": "surprise",
24     "score": 0.000
25   }
26 ]
```

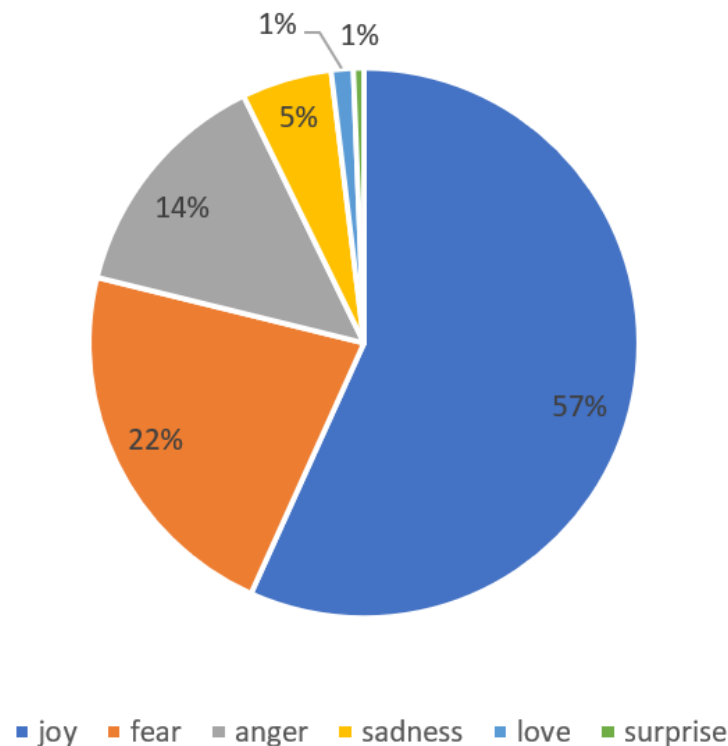
Listing 1: Output of the trained model for the sentence "I am very worried about the safety of bike lanes in this city."

16GB Tesla V100⁶ GPUs) to accelerate the training process. The evaluation results for the training process dictated the accuracy of 93.8% and F1-score of 94.06.

Classifying Emotions in a Tweet Dataset

We ran the fine-tuned BERT model on the parks tweet dataset described in Section 5.1 to classify the emotion for each tweet. Figure 5.12 describes the distribution of the predicted emotions over the parks tweet dataset 5.1:

Figure 5.12: Distribution of emotion classes over the Parks and Recreation related dataset 5.1



As could be understood from Figure 5.12, more than half (60.3%) of the tweets are predicted to convey the emotion of ‘joy’. The emotions ‘fear’

⁶<https://www.nvidia.com/en-gb/data-center/tesla-v100/>

and ‘anger’ have the next two highest portions among the tweets which are 23.5% and 14.8% respectively. ‘sadness’ (5.6%) and ‘love’ (1.4%) and ‘surprise’ (0.7%) were predicted to be the least conveying emotions among the tweets.

The Usage of Predicted Emotions in Analyzing Tweets and Requirements

Since we are mining the tweets to explore people’s concerns and requirements around a subject, the tweet emotions can explain a lot regarding the opinion they are giving or the demand they are making. Here are two important use cases for the predicted tweet emotions in analyzing the clusters created in Section 4.7.

This Sub-Section also refers to the discussion on benefits of emotional analysis in **RQ3** as mentioned in Section 1.2.

1. Find the clusters with subjects that people have expressed the most negative emotions towards them.

It could be very informative and beneficial to analyze the emotion of each tweet when extracting the requirements from tweet analysis. Because the type and the nature of the opinion or the meaning of the tweet are very related to the emotion that the tweet expresses. To back up this idea, Table 1 represents two tweets in a cluster with the subject of “bike lanes” (a cluster theme extracted in the project explained in 5.1) with two opposing classified emotions.

Even though both tweets are about the same topic, the type of first tweet which is labeled as “love” emotionally, is a wishful and thankful thinking tweet about adding bike lanes, whereas the second tweet

Tweet	Emotion	Emotion Score
Canada's cities added COVID bicycling lanes to improve access to jobs, parks, and stores, and they're liking the results. Congratulations Moncton, Kitchener, Ottawa, Montreal, Vancouver, Victoria, Toronto, Calgary, Winnipeg. https://t.co/LVLQfonCZW	love	0.99
NOT BIKE LANES!!! New Calgary wheeling lanes opposed by some southwest businesses https://t.co/VwQfVHzpdm	anger	0.99

Table 5.9: Example of two emotionally opposing tweets within the same topic.

which is labeled as “anger” emotionally, is a specific complain and request for change about the actions of some businesses towards the bike lanes. This is an example of the importance of emotion analysis in extracting requirements from tweets.

Table 5.10, Table 5.12, and Table 5.11 show the top-5 layer-1 clusters with the most number of tweets emotionally predicted to be as ‘anger, ‘fear, and ‘sadness’ respectively.

cluster	joy	sadness	love	fear	surprise	anger
1	55.5	8.1	1.5	8.1	0.7	25.7
9	42.8	7.1	3.5	21.4	0.0	25.0
11	20.0	12.6	0.0	43.5	0.0	23.7
8	64.8	4.0	1.6	6.1	1.1	22.1
7	66.7	7.7	1.4	8.9	1.3	13.9

Table 5.10: Layer-1 clusters sorted by anger

Looking at Table 5.10, we observe that the top-4 clusters with the greatest number of tweets labeled as ‘anger’ have between 22.1% and 25.7% emotionally angry tweets. Even though the majority of the

cluster	joy	sadness	love	fear	surprise	anger
4	3.6	1.2	0.0	88.8	0.0	6.4
3	20.3	6.1	0.0	61.5	0.4	11.5
11	20.0	12.6	0.0	43.6	0.0	23.7
9	42.8	7.1	3.5	21.4	0.0	25.0
13	81.8	1.2	1.2	10.2	0.6	4.9

Table 5.11: Layer-1 clusters sorted by fear

cluster	joy	sadness	love	fear	surprise	anger
5	61.8	30.3	0.0	6.3	0.0	1.6
11	20.0	12.6	0.0	43.6	0.0	23.7
1	55.5	8.1	1.5	8.2	0.8	25.8
7	66.7	7.7	1.4	8.9	1.3	13.9
9	42.8	7.1	3.6	21.4	0.0	25.0

Table 5.12: Layer-1 clusters sorted by sadness

tweets in clusters 1, 9, and 8 are joy, the portion of the angry tweets is relatively high being between 22.1% and 25.7% compared to the average emotion distribution among all tweets as shown in Figure 1 being 14.8%.

As could be seen in the Table 5.12, Table 5.10, and Table 5.11, the clusters 9, 11, 7, and 1 all fall under the top-5 clusters with highest number of negative emotions. Which means that the polarity of the tweets in these clusters is mostly negative and mostly complaining, as opposed to wishful thinking and positive opinions. This deduction puts the topics assigned to these clusters, which are respectively “pedestrian pathways safety”, “reports on busy streets”, “biking and walking lanes”, and “parks and open spaces” under-sensitive subjects of discussion and possibly should be further analyzed for more important and urgent requirements.

2. Detect the tweets that are complaining about the overall subject assigned to a cluster.

To dig deeper into the mentioned clusters in Table 5.12 and Table 5.10, which are sorted based on the emotions ‘anger’ and ‘fear’ respectively, as described in Section 4.8, the overall topic assigned to the cluster 9 is “pedestrian and biking pathway safety”. Even Though the majority of the tweets in this cluster are labeled as ‘joy’, this cluster contains 25% anger and 21.4% fear which are relatively high compared to the correlated numbers in Figure 5.11. This means that in comparison to the normal percentages the overall emotions and sentiments towards cluster 9 are mostly negative emotions. Here are the top-10 closest tweets that are labeled as ‘anger’ and ‘fear’ to the cluster centroid:

Tweet	emotion	Emotion Score
In other words, we have a pedestrian incident almost every 2 days. And yet communities like must fight for pedestrian safety measures. #yyc #yyccc #yyccwalk https://t.co/Wyec0RJNs2	anger	0.80
So the boulevard is once again mowed (as per @cityofcalgary mandate) &i another close call with a grown a*s adult riding their bike on the sidewalk (which is illegal in #yyc if youre over the age of 14) when there’s a bike lane 5m away https://t.co/5hkRge8iOW	anger	0.88

<p>"Since the incident, the city told the family that lights will be installed at the intersection."</p> <p>I hate that making a crosswalk safer only happened after an SUV driver hit and seriously injured a teen. #yycwalk https://t.co/W2y1VLQgDE</p>	fear	0.94
<p>But you know, pedestrians need to be more alert... #yycbike #yyc #WarOnCars https://t.co/fVclTuPbfM</p>	fear	0.76
<p>@projectstartrek @MTA In 2001 in Calgary on a pedestrian street - Stephen Ave - 3 bike cops were zooming around lunchtime peeps. Signs said no bikes/skateboards. I yelled "slow down!" They circled me. Cop behind me kept saying "reciprocity" like it was a Buddhist mantra. I asked him to spell it.</p>	fear	0.65
<p>This is a tip of the hat to the conversation that we need to have in #yyc about the cost of supporting our sprawling road grid.</p> <p>The cost of pedestrian/bike/transit infrastructure pales in comparison to the cost of roads and yet we are reluctant to make those investments. https://t.co/jdbaX6WLb8</p>	fear	0.99

https://t.co/xyVzpn4qaE Oh goody theNIGHT-MARE continues Drunk people on scooters Friday nights Can't wait to get assaulted on my way home again by drunk youth ...yippie ...can't wait	fear	0.77
Just had a meeting with the city to discuss safe cycling connectivity, in the Riley communities. Was told one site that helps them make infrastructure budget decisions is https://t.co/CYUAM67QEe #yycbike if you have a dangerous encounter please post it on this site.	anger	0.99
Issued a ticket to a driver who stopped in a bike lane this morning. It doesn't matter how long you're going to be, you can't stop in the bike lane, just the same as you can't stop in the middle of the road. #yyc #yyctrffic https://t.co/dKYDaYYa4s	anger	0.88

Table 5.13: Top-10 closest tweets to cluster 9 centroid labeled as 'fear'.

As you could see in the top-10 close tweets to the cluster centroid that is labeled as being either 'anger' or 'fear', almost all of them address a severe problem, which is the risk of bike and scooter traffic compromising pedestrian safety. As could be seen from Table 5.13, the important tweets concerning the issues around the cluster's overall subject can be automatically detected using tweet emotion analysis.

Chapter 6

Discussion and Threats to Validity

6.1 Discussion

Here we discuss how the development of DeKoReMi adds value to the body of knowledge, focusing on the gap between the industry and the academic research in requirements engineering and decision support systems. We then compare the proposed methodology pipeline with the existing related studies as explained in Chapter 3 and discuss how this research could complement the already conducted studies in this criteria. And lastly, we elaborate on how this research is focused on delivering needs in the industry via academic research and conducting empirical studies.

Validation of Results

This research aims to provide enough information to the industry from any existing textual content (in this case, Twitter) to guide them during their decision-support system to make data-supported decisions. Thus, the results are consumed by the industry (in this case, the city of Calgary and

Suncor Energy company), and the value of the final results could be evaluated only after applying the extracted knowledge in the decision-making process and getting feedback on how beneficial the provided information is to the domain experts. Even though this is a time-consuming and difficult evaluation process, as we have successfully applied the methodology pipeline on multiple real-life projects using "action research" methodology, we have concrete feedback (Sections 5.2 and 5.1) from the industrial collaborators and domain experts on how the methodology performs and how they would further apply it on their internal textual data.

Moreover, one of the main sub-steps of the DeKoReMi methodology pipeline is to cluster tweets and assign themes of discussion to them. The sheer task of assigning themes to the clusters is empirically evaluated and optimized during the empirical experiment conducted and explained in Section 5.4, evaluating the absolute quality of an unsupervised task like text clustering is tricky and multiple studies have worked on this and concluded that there are no single best metric to evaluate the validity of the cluster indices (Maulik and Bandyopadhyay (2002), Arbelaitz et al. (2013), Dimitriadou et al. (2002)). However, we have used internal indices to sort and compare the cluster over others in Section 4.7 when selecting clusters for further requirements analysis.

Qualitative evaluation method

Despite the requirements and theme extraction part of the methodology that could be and has been manually evaluated in Sub-Section 5.1 by three field experts, the final results of the methodology is pure knowledge and could not easily be evaluated quantitatively. This is because the effect of the

knowledge could be only measured after applying and using the extracted knowledge in the decision-making process and measuring the degree of the improvements that the provided knowledge has made from the methodology, which is why the overall evaluation of the DeKoReMi methodology at this stage has been qualitatively evaluated the field experts from Suncor Energy.

After delivering the final extracted knowledge to the field experts from Suncor Energy, they examined the provided results and evaluated the quality of the outcome (requirements and supporting knowledge) and its potential influence in their decision-making process. The qualitative evaluation was delivered to us in a dedicated meeting, including nine field experts from Suncor. The summary and notable parts of their feedback and evaluation are brought in Section 5.2.

Comparing DeKoReMi with the Related Works

We have delivered a review of the most related studies to different steps of the proposed methodology pipeline in Chapter 3. A short comparison and classification of the added values and different features of the related works are listed in Table 3.1. The listed features are the benefits that each study brings to the process of decision support systems. Now that we have fully introduced DeKoReMi, Table 6.1 adds the proposed methodology into the feature classification Table 3.1.

Table 6.1 compares the added values of the DeKoReMi methodology pipeline to the applications of NLP on Requirements Engineering in the Decision Support Systems (DSS) with the reviewed related studies. As could

Features	Studies
dynamic topics (or requirements)	DeKoReMi , Rosa et al. (2011) , Lossio-Ventura et al. (2021) , Rejito et al. (2021)
providing knowledge	DeKoReMi , Kitamura et al. (2007)
empirical evaluation	DeKoReMi , Hong and Davison (2010) , Haque et al. (2019)
semantic analysis	DeKoReMi ,
multi purpose or domain	DeKoReMi , Rosa et al. (2011) , Lossio-Ventura et al. (2021)
automated process	DeKoReMi , Jafari et al. (2021) , Haris et al. (2020)
used deep learning	DeKoReMi , Ito and Chakraborty (2020)
requirements (or topic) validation	DeKoReMi , Rosa et al. (2011)
method for topic assignment to text	DeKoReMi , Lossio-Ventura et al. (2021)

Table 6.1: The list of features and values offered by the reviewed studies along with **DeKoReMi**

be understood from Table 6.1, DeKoReMi offers all the features and added values of the reviewed literature and related studies. The two added features on top of the related studies which is very important but has been paid attention very few studies and in only one of the reviewed studies provide are "deep learning usage" and "semantic analysis" (which is addressed in none of the reviewed related works). The former means that in this study, we move beyond using statistical models like TF-IDF in the topic modeling process using clustering ideology and use Transformers and in particular BERT (in Section 4.6), which is (in the base model) a deep 12 layered neural network with 110+ million training parameters to be trained [Devlin et al. \(2018c\)](#). And the latter conveys that the deep learning method used in the methodology is pre-trained on over 3.3 billion words and models the se-

mantics of the sentences as well [Reimers and Gurevych \(2019\)](#). Whereas most of the used methods for tweet embedding (or representing tweets in terms of vectors) such as commonly employed TF-IDF (used in [Haque et al. \(2019\)](#), [Rosa et al. \(2011\)](#), [Rejito et al. \(2021\)](#)) which is a merely statistical model or (a relatively simpler neural network method) Word2Vec (used in [Lossio-Ventura et al. \(2021\)](#)) that does not model and convey the deep semantic meaning of the text when converting tweets into vectors. The empirical studies explained in Sections [5.4](#) and [5.1](#) also use emotional analysis which focuses on the semantic expression of the tweets. DeKoReMi also offers extraction of dynamically created requirements, unlike the mere text classification methods which use pre-defined classes, such as [Haque et al. \(2019\)](#) and [Jafari et al. \(2021\)](#). Apart from the requirements extraction, DeKoReMi also provides related knowledge in terms of Subjective and Objective analysis around the extracted topics and requirements in Section [4.9](#).

Action Research and Collaboration with the Industry

As described in detail in Chapter [5](#), this thesis employs "action research" for developing the DeKoReMi pipeline over the course of three industrial-academic projects collaborating with the city of Calgary and Suncor Energy company. Which means during the development of DeKoReMi, we heavily interacted with the industry collaborators in a parallel feedback loop between the research and action in the projects to adjust the methodology and its outcomes based on real-life requirements from the industry. Also, the final objective and subjective analyses templates proposed in Section

4.9 were formed with direct consultation with domain experts from Suncor Energy during the final implementation of the project in Section 5.2. Of course, the evaluation of the final analyses is qualitative and should be done by the field experts in action during the decision-making process. The final feedback and evaluation of the results are given by 8 domain experts from Suncor Energy and are mentioned in Section 5.2.

DeKoReMi in decision-making process

Decision-making is not merely the act of prioritizing one decision above others. A decision-making process consists of three main stages, which are "intelligence", "design", and "choice" [Pomerol and Adam \(2004\)](#). During the "intelligence" stage, textual analysis introduces problem areas and provides a cognitive representation of the decision scenarios. The scenarios could be either improving and perfecting already existing decision problems or forming and explaining new decision areas. The "design" stage is the generation and compilation of knowledge assets (such as limitations, objectives, explanatory gold nuggets of information, and stakeholders) extracted from the text. The "design" stage aims to provide complementary and supporting knowledge around the extracted decision problems in the "intelligence" stage. And lastly, the "choice" stage is to prioritize the possible options and recommend the final outcome to the decision-making process.

DeKoReMi plays an integral role in the first two stages, being "intelligence" and "design". In the "intelligence" stage, DeKoReMi uses deep natural language models to extract the different themes of discussion and requirements, revealing the new and existing problem areas from text. The

extracted problem areas are further transformed into decision problems. And in the "design" stage, the required supporting knowledge and gold nuggets of information around the elicited decision problems is gathered, consolidated, and provided. The domain experts will further employ the provided decision scenarios and supporting knowledge assets extracted in the "intelligence" and "design" stages to prioritize the possible solutions and make data-supported decisions.

6.2 Threats to Validity

In the course of this thesis, we have proposed a requirements and knowledge extraction methodology pipeline and conducted several empirical experiments to fine-tune the methodology. However, it is crucial to identify any possible threats to the validity of the research, methodology, and results. In this research, we discuss the known limitations and threats to the validity of the research materials and our efforts to mitigate them or propose ideas to fine-tune the research in possible future efforts.

Validity of the DeKoReMi Methodology and the Results

Validity of the Clusters and Assigned Themes

During the implementation of the DeKoReMi there is an important step, which is clustering semantically similar tweets into the same groups; there is no absolute evaluation of the quality of the clusters (in terms of internal indices) to confirm the effectiveness of the used clustering methodology. Although it is confirmed by several studies ([Maulik and Bandyopadhyay](#)

(2002), [Arbelaitz et al. \(2013\)](#), [Dimitriadou et al. \(2002\)](#)), there is no single best method to evaluate the quality of a clustering task (especially when there are more metrics like the semantic similarity in textual clustering as opposed to mere numeric clustering), and often the evaluation results of different methods may become contradictory (as seen in the results of [Lossio-Ventura et al. \(2021\)](#)). Thus, it makes it difficult to provide a concrete evaluation of clustering results. However, the combination of Sentence-BERT with K-Means has been tested in [Ito and Chakraborty \(2020\)](#) and judging by the used evaluation method (using cosine similarity and euclidean distance) they concluded that Sentence-BERT outperforms BERT and Word2Vec in terms of resulting in more numerically similar clusters. Which is a very similar approach to the clustering method used in DeKoReMi with the difference that DeKoReMi uses the improved Sentence-BERT as used in [Ito and Chakraborty \(2020\)](#) combined with multi-layer fine-grained clustering as described in Section 4.7. Moreover, we provide an empirical evaluation of the semantic closeness and tweet membership quality of the clusters in Section 5.1, which is done by the field experts from the city of Calgary. Although the number of validated clusters could definitely be more to support the results, but since we aimed to evaluate the results by the field experts from the city of Calgary, the provided time devotion by the city of Calgary members was very limited.

Validity of the Final Analyses

As discussed in the Discussion Section 6.1, the evaluation of provided ending analyses as described in Section 4.9 is a complicated task. The provided knowledge regarding the extracted themes of discussion and requirements is

delivered in the form of subjective and objective analyses, which are formed with consultation with the domain experts from the industry to adjust them, aiming to be applicable in real-life Decision Support Systems. Thus, the evaluation of the quality and the effectiveness of the outcome should be done by the domain experts while applying them during their decision-making process. It is only obvious that the feedback received from the such evaluation cannot be presented in numbers and only can be evaluated subjectively by the domain experts in action. During the collaboration with Suncor Energy, we have fully implemented the DeKoReMi methodology, highly collaborating with their domain experts to fine-tune the method and the results. And a final observation and qualitative evaluation of the results are presented in Section 5.2. The evaluation of the semantic closeness and density of the clusters are also empirically evaluated in Section 5.1 by the field experts from the city of Calgary.

Of course, there is more room to make further evaluations of the effectiveness of the results, and to do so, we should re-implement the methodology in more real-life industrial projects. Due to time and money limitations, we could not perform the methodology on more projects as collaborating with industrial counterparts are extremely time-consuming and could not be performed numerous times during a Master's degree.

Validity of Empirical Experiments

Validity of the Tweet Actionability Empirical Experiment

During the empirical experiment conducted and explained in Section 5.3 we manually annotated 2300 tweets in terms of being actionable or non-

actionable by four summer interns. There is a subjectivity threat to the validity of the labels assigned by the four annotators. To address this issue, each tweet has been annotated by two annotators, and a tweet can be considered actionable only if both annotators have found the tweet to be actionable. Also, they all were explained by the domain experts the definition and examples of attainability to make sure everyone have the same understanding of the subject under study and the definition of actionability within the subject. Also, there was a limitation in the number of tweets, otherwise, it would be beneficial to have more tweets annotated for evaluation and supporting the hypothesis.

Validity of the Automatic Theme Assignment Empirical Experiment

The second empirical experiment conducted during the development of DeKoReMi was detecting the optimal method for the objective assignment of themes to the tweet clusters that are closest to human perception. As mentioned in Section 5.4 we described the experiment design using which we have set scores to the four auto theme assignment methods proposed in Section 4.8. Similar to the previous sub-section, there is a subjectivity threat to the validation of the result as we are comparing the objective results with human perception, which is by nature subjective. To mitigate this threat, we used four annotators during the experiment and aggregated all the opinions to alleviate personal subjectivity and retrieve the general opinion (from multiple people) which is closer to the definition of human perception.

Similarly, there is of course the limitation of the number of annotators we could use and the time they could have devoted to the research. If there were no limitations on that criteria, we would suggest gathering more annotators

to eliminate subjectivity and use larger subsets of tweets for reading as a representation of the whole cluster.

Conclusions

In this section, the identified threats to the validity of the proposed methodology and the conducted empirical experiments were discussed. In general, as the nature of this study is more empirical and investigative, and the results were partially generated by unsupervised machine learning models, the perfect evaluation of the results requires many hours of annotation from many annotators to eliminate all the threats to the validity. Although, using the limited resources we've had, we tried to mitigate this threat by hiring four interns and heavily interacting with our industry collaborators, and asking domain experts for feedback and annotation to validate the results and conduct the experiments.

Chapter 7

Conclusions and Future Work

7.1 Summary

Decision Support Systems (DSS) are designed to help decision-makers through a systematic pipeline of data analysis to make the right decisions using the proper data. Therefore, data gathering and analysis is one of the integral steps in the DSS pipeline. A very informative type of data used in the DSS to elicit insights for supporting decisions is textual content. However, knowledge extraction from textual data is not an easy task because firstly, there is more amount of textual data generated than could ever be properly analyzed, and secondly, textual data is mostly written in unstructured natural language, which could be interpreted in many ways conveying deeper meanings like sarcasm or feelings rather than the absolute translation of the words. One of the most widely used sources of textual content in the industry, politics, and academia is Twitter, where people express themselves freely and post over 500 million tweets every day about various subjects. However, the task of extracting useful information from tweets is very diffi-

cult, specifically for making pivotal decisions in the industry. This is due to the nature of tweets being public opinions written in unstructured, possibly grammatically incorrect and conversational natural language that also contains multimedia content to complement the tweet.

Aiming to fill this gap, we propose the "Deep Knowledge and Requirements Miner" (DeKoReMi) methodology pipeline that employs state-of-the-art Natural Language Processing (NLP) and Deep Learning techniques to fetch and analyze large amounts of textual data from social media, specifically Twitter and elicit valuable gold nuggets of knowledge from them to support the decisions made in the DSS pipeline. DeKoReMi has been developed using "Action Research" methodology during the course of three industrial-academic projects collaborating with the city of Calgary and Suncor Energy. Therefore, the methodology is built to extract the required knowledge from textual content tailored to industrial decision-making needs. DeKoReMi employs different NLP tasks in its pipeline such as text classification, sentence embedding, tweet clustering, and emotional analysis to extract different themes of discussion among the tweets, elicit the requirements from the extracted themes, and provide knowledge around the elicited requirements to the domain experts to support guide them through their decision making process. The DeKoReMi methodology pipeline has been proven to be effective both in providing pivotal and valuable information to the domain experts and decision-makers during the implementation of the projects.

To evaluate and improve two of the important steps in the methodology, we have conducted two extra empirical experiments. The first empirical experiment is conducted to investigate the influence of the expressed emotions in the text on the possibility of taking action based on the text. And the

second empirical experiment is to identify the best objective and automatic theme assignment method among the proposed methods to generate the closest themes to human perception.

The developed methodology pipeline has numerous applications in the industry, academia, and politics where there is a dire need for analyzing unstructured text and extracting the different segments of information that the text represents, eliciting the requirements, and providing the gold nuggets of knowledge that support each requirement.

7.2 Future Work

Automating the Extraction of Actions from the Requirements

One of the empirical experiments conducted in this thesis is investigating the effect of emotional analysis on the possibility of stemming actions from tweets. This study was done on sheer text and emotions and could be expanded in two ways. The first expansion idea is to investigate other attributes of text than emotions, like the structure of the sentences, the semantic meaning, the subject class, or the nature of the text (such as being news, personal opinion, scientific fact, etc.) and find patterns between these attributes and the actionability of the text. Doing so, a model could be developed that inputs text and automatically results in action, similar to a decision-maker. The second expansion idea is to move beyond sheer text and use requirements, and the corresponding provided knowledge by the proposed methodology - DeKoReMi - and automate the process of eliciting actions from requirements. If these ideas are implemented together, the

upgraded methodology has the ability to partially replace a human decision-maker and consume massive textual data and generate actions based on the consumed text.

Automated Problem Area Assignment

The DeKoReMi methodology pipeline is designed to elicit requirements and the corresponding knowledge. However, the act of mapping the requirements to the problem areas should be done by domain experts. This idea is the attempt to partially replace the human in the loop and automatically map the requirements to the problem areas, and further assign the tasks to the corresponding departments to address the requirement. If this idea is combined with the previous future work explained in Section 7.2 it could result in a powerful decision-making tool for organizations with multiple sub-departments, such as the municipal organizations and the governments.

Analyzing Tweet Streams

This study was done using gathered data from Twitter in specific periods of time. The idea is to train models based on the provided analysis over the static dataset gathered from Twitter, and classify tweet streams into different areas, such as the discussion theme, requirement type, actionability, and problem area. This idea is very helpful for the applications that require real-time analysis of public opinion, such as the politics and the government requirements extraction in different times, such as the COVID-19 pandemic.

Customer satisfaction assessment

There is a high demand for assessing customer satisfaction in most industries, and the majority of the feedback for many products is received in a textual format. The textual feedback could be received in many forms in different products, such as textual reviews for applications and software-based products, tweet reviews for a wider variety of industries, and direct customer feedback gathered through surveys. All of these require textual analysis to assess the level of customer satisfaction to react upon. The idea is to combine DeKoReMi and emotional analysis to measure customer satisfaction using the text-based reviews received via various forms of feedback and present a deeper understanding and knowledge around different aspects of the satisfaction of the customers from the developed product.

Delivering a software product (web application) that fully delivers the DeKoReMi features employing visualization

Even though the methodology is mostly automatic, meaning that most of the tasks are done using NLP and ML models, the combination and execution of the methods are still manual, as each step has been developed individually. The idea is to combine all of the steps and integrate them into a web application that automates the whole process and employs various visualization techniques to demonstrate the resulting outcome. Using visualization, the customers (field experts from industries) could have a better understanding of the provided themes of discussion, extracted requirements, and supporting knowledge around the requirements.

Bibliography

The PROMISE Repository of Software Engineering Databases. School of Information Technology and Engineering, University of Ottawa, Canada, 2015. URL <http://promise.site.uottawa.ca/SERepository>.

Zahra Shakeri Hossein Abad, Vincenzo Gervasi, Didar Zowghi, and Ken Barker. Elica: An automated tool for dynamic extraction of requirements relevant information. In *2018 5th International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, pages 8–14. IEEE, 2018.

Alfred V Aho. Algorithms for finding patterns in strings, handbook of theoretical computer science (vol. a): algorithms and complexity, 1991.

Mehdi Allahyari, Seyedamin Pouriyeh, Mehdi Assefi, Saeid Safaei, Elizabeth D Trippe, Juan B Gutierrez, and Krys Kochut. Text summarization techniques: a brief survey. *arXiv preprint arXiv:1707.02268*, 2017.

David Alvarez-Melis and Martin Saveski. Topic modeling in twitter: Aggregating tweets by conversations. In *Tenth international AAAI conference on web and social media*, 2016.

Olatz Arbelaitz, Ibai Gurrutxaga, Javier Muguerza, Jesús M Pérez, and Iñigo

- Perona. An extensive comparative study of cluster validity indices. *Pattern recognition*, 46(1):243–256, 2013.
- Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.
- DM Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation: the journal of machine learning research, v. 3. 2003.
- Adailton Ferreira de Araújo and Ricardo Marcondes Marcacini. Re-bert: automatic extraction of software requirements from app reviews using bert language model. In *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, pages 1321–1327, 2021.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018a. URL <http://arxiv.org/abs/1810.04805>.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018b. URL <http://arxiv.org/abs/1810.04805>.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018c.

- Evgenia Dimitriadou, Sara Dolničar, and Andreas Weingessel. An examination of indexes for determining the number of clusters in binary data sets. *Psychometrika*, 67(1):137–159, 2002.
- Google. Understanding searches better than ever before. URL <https://blog.google/products/search/search-language-understanding-bert/>.
- Maarten Grootendorst. Keybert: Minimal keyword extraction with bert., 2020. URL <https://doi.org/10.5281/zenodo.4461265>.
- Monika Gupta and Parul Gupta. Research and implementation of event extraction from twitter using lda and scoring function. *International Journal of Information Technology*, 11(2):365–371, 2019.
- Md Ariful Haque, Md Abdur Rahman, and Md Saeed Siddik. Non-functional requirements classification with feature extraction and machine learning: An empirical study. In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pages 1–5. IEEE, 2019.
- M Syauqi Haris, Tri Astoto Kurniawan, and Fatwa Ramdani. Automated features extraction from software requirements specification (srs) documents as the basis of software product line (spl) engineering. *Journal of Information Technology and Computer Science*, 5(3):279–292, 2020.
- Fahad ul Hassan and Tuyen Le. Automated requirements identification from construction contract documents using natural language processing. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, 12(2):04520009, 2020.

- Aron Henriksson and Jelena Zdravkovic. A data-driven framework for automated requirements elicitation from heterogeneous digital sources. In *IFIP Working Conference on The Practice of Enterprise Modeling*, pages 351–365. Springer, 2020.
- A. Hernandez-Suarez, G. Sanchez-Perez, K. Toscano-Medina, V. Martinez-Hernandez, V. Sanchez, and H. Perez-Meana. A web scraping methodology for bypassing twitter api restrictions, 2018. URL <https://arxiv.org/abs/1803.09875>.
- Liangjie Hong and Brian D Davison. Empirical study of topic modeling in twitter. In *Proceedings of the first workshop on social media analytics*, pages 80–88, 2010.
- huggingface. Sentence transformer pre-trained model: all-mpnet-base-v2. <https://huggingface.co/sentence-transformers/all-mpnet-base-v2>, 2021.
- ES Hur, Thomas Cassidy, and BG Thomas. Seeding sustainability through social innovation in fashion design, proceedings of the crafting the future. In *The Crafting the Future: the 10th European Academy of Design Conference*. The European Academy of Design, 2013.
- Hidetoshi Ito and Basabi Chakraborty. Social media mining with dynamic clustering: a case study by covid-19 tweets. In *2020 11th International Conference on Awareness Science and Technology (iCAST)*, pages 1–6. IEEE, 2020.
- Parinaz Jafari, Malak Al Hattab, Emad Mohamed, and Simaan AbouRizk. Automated extraction and time-cost prediction of contractual report-

- ing requirements in construction using natural language processing and simulation. *Applied Sciences*, 11(13):6188, 2021.
- Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 1972.
- Natthaphon Kengphanphanit and Pornsiri Muenchaisri. Automatic requirements elicitation from social media (aresm). In *Proceedings of the 2020 International Conference on Computer Communication and Information Systems*, pages 57–62, 2020.
- Motohiro Kitamura, Ryo Hasegawa, Haruhiko Kaiya, and Motoshi Saeki. A supporting tool for requirements elicitation using a domain ontology. In *Software and data technologies*, pages 128–140. Springer, 2007.
- Jay Kumar, Junming Shao, Salah Uddin, and Wazir Ali. An online semantic-enhanced dirichlet model for short text stream clustering. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 766–776, 2020.
- M Sunil Kumar, A Harika, C Sushama, and P Neelima. Automated extraction of non-functional requirements from text files: A supervised learning approach. *Handbook of Intelligent Computing and Optimization for Sustainable Development*, pages 149–170, 2022.
- Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014.

- Juan Antonio Lossio-Ventura, Sergio Gonzales, Juandiego Morzan, Hugo Alatrasta-Salas, Tina Hernandez-Boussard, and Jiang Bian. Evaluation of clustering and topic modeling methods over health-related tweets and emails. *Artificial Intelligence in Medicine*, 117:102096, 2021.
- Walid Maalej, Maleknaz Nayeibi, and Guenther Ruhe. Data-driven requirements engineering-an update. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 289–290. IEEE, 2019.
- Bell Manrique-Losada, Carlos M Zapata-Jaramillo, and Diego A Burgos. Re-expressing business processes information from corporate documents into controlled language. In *International Conference on Applications of Natural Language to Information Systems*, pages 376–383. Springer, 2016.
- Mohammad Navid Masahati. Twitter scraper. <https://github.com/mammalofski/Twitter-Scraper>, 2021.
- Ujjwal Maulik and Sanghamitra Bandyopadhyay. Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on pattern analysis and machine intelligence*, 24(12):1650–1654, 2002.
- Edi Surya Negara, Dendi Triadi, and Ria Andryani. Topic modelling twitter data with latent dirichlet allocation method. In *2019 International Conference on Electrical Engineering and Computer Science (ICECOS)*, pages 386–390. IEEE, 2019.
- University of Calgary. High performance computing (hpc). URL <https://it.ucalgary.ca/research-computing-services/our-resources/high-performance-computing-hpc>.

- Jean-Charles Pomerol and Frederic Adam. Practical decision making—from the legacy of herbert simon to decision support systems. In *Actes de la Conférence Internationale IFIP TC8/WG8*, volume 3, pages 647–657, 2004.
- Bing Qi, Aaron Costin, and Mengda Jia. A framework with efficient extraction and analysis of twitter data for evaluating public opinions on transportation services. *Travel behaviour and society*, 21:10–23, 2020.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.
- Peter Reason and Hilary Bradbury. *Handbook of action research: Participative inquiry and practice*. sage, 2001.
- Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.
- J Rejito, A Atthariq, and AS Abdullah. Application of text mining employing k-means algorithms for clustering tweets of tokopedia. In *Journal of Physics: Conference Series*, volume 1722, page 012019. IOP Publishing, 2021.
- Kevin Dela Rosa, Rushin Shah, Bo Lin, Anatole Gershman, and Robert Frederking. Topical clustering of tweets. *Proceedings of the ACM SIGIR: SWSM*, 63, 2011.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

- Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium, October–November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1404. URL <https://www.aclweb.org/anthology/D18-1404>.
- Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *China national conference on Chinese computational linguistics*, pages 194–206. Springer, 2019.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- Zhiquan Ye, Yuxia Geng, Jiaoyan Chen, Jingmin Chen, Xiaoxiao Xu, Suhang Zheng, Feng Wang, Jun Zhang, and Huajun Chen. Zero-shot text classification via reinforced self-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3014–3024, 2020.
- Jianhua Yin and Jianyong Wang. A dirichlet multinomial mixture model-based approach for short text clustering. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 233–242, 2014.
- Wen Zhang, Taketoshi Yoshida, and Xijin Tang. A comparative study of tf* idf, lsi and multi-words for text classification. *Expert systems with applications*, 38(3):2758–2765, 2011.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.