Text Mining Thesis

Document Classification on EQF levels with Multilingual datasets in English

Ajda Efendi

a thesis submitted in partial fulfilment of the requirements for the degree of

MA Linguistics
(Text Mining)

Vrije Universiteit Amsterdam

Computational Lexicology and Terminology Lab
Department of Language and Communication
Faculty of Humanities

Supervised by:	Luis Guilherme
2^{nd} reader:	Isa Maks
Submitted:	October 27, 2023

Abstract

This thesis investigates the application of Natural Language Processing (NLP) techniques for document classification on the European Qualifications Framework (EQF) levels. The primary objective is to explore the feasibility of training a system using English-translated multilingual datasets from various countries to predict the Dutch EQF level (NLQF). To achieve this, two main classification methods are employed: a keyword-matching approach utilizing TF-IDF with n-grams and a machine learning method using TF-IDF with n-grams with Logistic Regression. The keyword matching approach utilizes a predefined list of phrases which are in different lengths, based on top-n salient phrases for each document class. These lists are used for the classification, while the Machine Learning method does not rely on such a list. The research aims to fill the existing gap in the literature regarding document classification on EQF levels and provide initial insights into the possibility of predicting NLQF levels based on EQF-labelled documents from diverse sources. The findings of this study have significant implications for educational institutions, policy-makers, and stakeholders involved in cross-border recognition of qualifications.

Declaration of Authorship

I, Ajda Efendi, declare that this thesis, titled *Document Classification on EQF levels with Multilingual datasets in English* and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a degree degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Date: October 27, 2023

Signed: AjdaEfendi

Acknowledgments

I am deeply grateful for the unwavering support and encouragement provided by my family throughout this challenging journey. Their belief in me and their constant presence have been my pillars of strength.

I extend my heartfelt gratitude to Prof. Dr. Mary Ann Walter, my esteemed Linguistics Professor, whose guidance and unwavering support have been instrumental in shaping my academic path. Her dedication to teaching and mentorship has been truly inspiring.
I would like to express my heartfelt gratitude to my supervisor, Dr. Luis Guilherme, for his invaluable guidance, unwavering support, and the generous amount of time he dedicated to overseeing and advising me throughout the course of this project. His expertise, constructive feedback, and encouragement have been instrumental in shaping this work. His commitment to my growth as a researcher and his genuine interest in my success have been truly motivating and inspiring. I am deeply appreciative of his mentorship and the positive impact he has had on my academic journey.

I would like to express my sincere appreciation to my friends Ellemijn Galjard, Sybren Moolhuizen and Buse Apel. Their insightful discussions, valuable feedback, and assistance have enriched the depth of my research and fostered a meaningful exchange of ideas.

My gratitude also extends to my colleagues at EDIA, whose support during my internship played an integral role in the development of my research. Their guidance, collaborative spirit, and keen interest in my work have been invaluable.

This academic journey has not only been about acquiring knowledge, but also a period of profound personal growth. The challenges I encountered have taught me resilience, determination, and the importance of continuous learning. I am grateful for the transformative impact this process has had on my own development.

In conclusion, I am honored to have had the opportunity to embark on this academic endeavor, surrounded by the support and encouragement of my loved ones, mentors, friends, and colleagues.

List of Figures

Contents

Chapter 1

Introduction

This study is conducted in the context of an internship with EDIA. This organization provides content labelling and moderation by combining Artificial Intelligence and human moderators to help companies manage their online content. With this study, EDIA aims to support institutions that provide courses/qualifications to automatically grade their documents according to the European Qualifications Framework (EQF). Therefore, EDIA aims to investigate if *it is possible to detect skill-level from the qualification-describing documents (in EQF).*
In a brief explanation of the European Qualifications Framework (EQF), it serves as a common reference framework to facilitate the comparison and recognition of qualifications across European countries (Directorate-General for Employment and Social Affairs and Inclusion (European Commission), 2023). It provides a standardized framework for understanding the knowledge, skills, and competencies associated with different educational levels.
This research centers on categorizing documents according to EQF levels, explicitly focusing on forecasting descriptions within Dutch EQF qualification documents (NLQF). The classifications are based on the English version of the collected data sets.
The primary challenge of this research stems from the shortage of existing prior studies in document classification concerning EQF levels. By addressing this void, I intend to contribute to advancing Natural Language Processing (NLP) techniques within the educational sector, ultimately facilitating the cross-border recognition of qualifications.

Initially, the study encountered the limitation of a small dataset size. For this reason, additional EQF-labelled datasets from other countries are incorporated, enriching the available data. Although EQF descriptions may exhibit variations among European countries, the fundamental premise remains consistent. While the interpretation of each EQF level might differ across nations, the core educational domain remains unchanged in all EQF-utilizing countries, such as the case of ”Primary Education” corresponding to level 1. Consequently, the dataset was expanded by integrating other EQF datasets to accommodate this variability. Given this context, the fundamental research inquiry formulated as:
”Is it possible to detect skill-level (using EQF) from the qualification-describing documents?”
Because the data consists of a collection of other EQF-using counties’ data sets, another sub-question arose, ”Is it possible to use various countries’ EQF level labelled qualification descriptions to classify Dutch EQF data set (NLQF)? To answer these questions, this research explores two main classification methods. The first approach

utilizes a keyword-matching approach enhanced by the Term Frequency-Inverse Document Frequency (TF-IDF) with n-grams. The second approach adopts a machine learning approach, employing TF-IDF with n-grams and Logistic Regression for document classification. Unlike the Keyword-Matching Approach, this method does not rely on a rule-based classification, but aims to leverage the patterns and relationships present in the data to make predictions by using a *model* and TF-IDF with n-grams. In simpler terms, two experiments are carried out: one employs a rule-based approach, while the other utilizes machine learning techniques.

The experimental setup involves developing/training the classification models on diverse EQF-labelled documents from multiple countries, utilizing English versions of the datasets. Among the datasets, there are resources in Latvian, Maltese, and Dutch languages, all of which possess corresponding English versions. Additionally, the Swedish dataset has been subjected to machine translation by EDIA. The anticipated outcomes of this research are twofold. Firstly, the results will shed light on the effectiveness of the Keyword-Matching Approach and the Machine Learning Method for document classification on EQF levels. Secondly, the research aims to offer preliminary insights into the feasibility of forecasting NLQF levels by utilizing a training set comprising EQF levelled documents sourced from various countries.
When contrasting the Keyword-Matching Approach with the Machine Learning Approach, I anticipate that the Machine Learning Approach will exhibit superior performance. This expectation arises from the fact that a machine learning model has the capacity to acquire a deeper understanding of the dataset's structure and patterns, leading to enhanced predictive capabilities.

The paper first provides background information about the task and the study. In Chapter 1, Introduction, the motivation for this study and brief information about the task and the data sets are also described, including the models and tools used for this study. The second chapter consists of previously conducted research on document classification and studies done by using TF-IDF, which gives an insight into what kind of approaches are conducted by using document classification and TF-IDF. The third chapter explains and demonstrates the task and data distribution in detail. Chapter 4 includes a detailed explanation of the methodologies used to answer the research question. In Chapter 5, the findings from these methodologies are presented, including the results of the experiments on the development dataset and Error Analysis. In the following chapter, Chapter 6, Error Analysis of the experiments are presented. Chapter 7, discussion and future work suggestions are stated. In this chapter, I delve into the discoveries and, based on these deliberations, investigate potential paths for improving, experimenting with, and advancing this specific research. Chapter 7, the Conclusion section, is a comprehensive recapitulation of the entire research project, presenting the insights accumulated throughout the study. Moreover, I discuss the results considering the research question explained earlier and discuss the outcomes by taking the previously explained expected behaviour of the models and approaches.
The code for this study is in this provided link: `https://github.com/cltl-students/Ajda_EFENDI`

Chapter 2

Related Work

This study advances the field of document classification[1] within the context of the European Qualifications Framework (EQF) through the application of TF-IDF. EQF is a hierarchical structure consisting of 8 levels, designed to classify various qualifications based on learning outcomes. Its primary function is to facilitate translation and comparison among diverse national qualifications frameworks. This framework enhances the clarity, comparability, and portability of individuals' qualifications, enabling the assessment of credentials from different countries and institutions (Union). The EQF levels serve as a standardized framework for categorizing qualifications based on their complexity and knowledge requirements. By accurately classifying documents according to their EQF levels, we can facilitate the recognition and comparison of qualifications across European countries.
As mentioned by Directorate-General for Employment and Social Affairs and Inclusion (European Commission) (2023), the European Qualifications Framework for lifelong learning (EQF) addresses the challenges posed by the diversity of European education and training systems. Differences in qualifications among countries can hinder trust in the quality and content of qualifications, impacting professional development, employment opportunities, and access to further learning. The EQF, established in 2008, serves as a common reference framework, translating qualifications into learning outcomes and enabling easy comparison among different European systems. It benefits learners, workers, employers, education providers, and more. The EQF promotes transparency, comparability, and portability of qualifications, supporting cross-border mobility and lifelong learning. It has also influenced the development of national qualifications frameworks and flexible learning paths in Europe. Furthermore, the EQF facilitates the comparison of qualifications with other countries, making the E.U. an attractive destination for talent worldwide.

In the realm of keyword extraction and document classification, significant progress has been made towards enhancing the precision and specificity of these processes. Koloski et al. (2021) introduced an innovative approach to supervised keyword extraction, specifically targeting the challenges presented by less-resourced languages like Croatian, Latvian, Estonian, and the relatively well-resourced Russian. Their work emphasized the importance of suitable training data in supervised methods and delved

[1]Document classification refers to the process of categorizing or labelling documents into predefined classes or categories based on their content, characteristics, or attributes. It is a fundamental task in natural language processing (NLP) and information retrieval.

into the intricacies of linguistic and semantic nuances.

One of their notable contributions was the proposal of a novel TF-IDF tagset matching technique, which complemented traditional approaches and aimed to improve recall in keyword extraction systems. Their work underscored the necessity of adapting to diverse linguistic contexts in media house environments.

In a separate but related study, Medved et al. (2016) expanded upon the research landscape by addressing multilingual document classification challenges, particularly in the context of English and French texts. They highlighted the significance of accurate keyword matching between these languages and leveraged top-performing TF-IDF scoring variants. Additionally, they introduced a statistical dictionary for translations, which played a crucial role in their systematic approach to document classification.

Medved et al. built upon the foundation laid by Koloski et al., emphasizing the need for precise keyword alignment between languages. Their methodology included the translation of English keywords into various French variants and the subsequent selection of the most suitable French document based on keyword intersection. This innovative approach demonstrated the importance of bridging language barriers in content analysis.

Both of these studies have contributed to advancing keyword extraction and document classification techniques. They offer valuable insights into the complexities of multilingual text analysis and underscore the refinement of keyword-based content analysis. In the subsequent sections, we will build upon these foundational works to present our novel approach in this evolving research domain.

In the field of term weighting for information retrieval and recommender systems, several approaches have been proposed. Notably, the Rocchio classification algorithm, initially introduced by J.J. Rocchio in 1971 for relevance feedback in querying full-text databases, has been adapted and extended for text and document categorization (Kowsari et al., 2019). This approach contrasts using boolean features, as Rocchio leverages TF-IDF weights to represent informative words. By constructing a prototype vector for each class using a training dataset, this algorithm effectively captures class characteristics. The prototype vector is formed by averaging the vectors of training documents belonging to the same category. During classification, the algorithm assigns a test document to the class exhibiting the highest similarity between the test document and prototype vectors. The predicted label for the test document is determined by calculating the smallest Euclidean distance between the document and the centroid of the corresponding class.

Furthermore, this approach provides the possibility to normalize centroids to unit length, enabling the label of test documents to be obtained through the identification of the class with the maximum dot product between the centroid and the document vector. However, it is essential to acknowledge that the Rocchio algorithm has limitations. These constraints include its capacity to retrieve only a limited number of relevant documents and its partial consideration of semantic factors. As such, researchers and practitioners have explored alternative classification methods, including boosting and bagging. Boosting, introduced by R.E. Schapire in 1990 to enhance the performance of weak learning algorithms, adapts the distribution of the training set based on previous classifier performance. On the other hand, bagging disregards previous classifiers, offering a distinct avenue for classification tasks. The presence of these alternative approaches contributes to the diversified landscape of document classification techniques, offering a range of strategies to achieve effective and accurate categorization outcomes.

Returning to the theme of term weighting, Marcińczuk et al. (2021) introduced a modification to traditional term-weighting schemes, including TF-IDF, BM25, and Universal Sentence Encoder (USE). Their approach incorporates the recency of a term, in addition to term frequency and document frequency, to compute relevance scores. Their modified TF-IDF and USE methodologies surpassed the performance of standard approaches across three datasets, underscoring the efficacy of considering term recency in information retrieval and recommender systems.

Marwah and Beel (2020) emphasized the importance of term weighting in various applications, such as information retrieval and recommender systems. The author discussed term weighting as a method to quantify the extent of terms in documents and the corpus. The effectiveness of term weighting was demonstrated in multiple scenarios, including text mining, text classification, document clustering, and medical science research.

Piskorski and Jacquet (2020) conducted a preliminary study comparing TF-IDF character n-grams with word embedding-based models for fine-grained document classification. The authors evaluated the performance of these two approaches and provided insights into their suitability and effectiveness in the context of fine-grained event classification.

In the realm of document classification, Taddy (2015) proposed a novel approach based on the inversion of distributed language representations. The author leveraged distributed word embeddings, such as word2vec or GloVe, to represent documents as dense vectors in a high-dimensional space. By inverting this process, the author developed a document classification algorithm that assigns labels to previously unseen documents based on their embedded representations.

The inversion of distributed language representations offers a promising avenue for document classification, as it harnesses the power of semantic information captured by word embeddings. Taddy's approach has succeeded in various domains, including sentiment analysis, topic classification, and document clustering.

In my research, I build upon the insights gained from the previously mentioned studies and intend to delve deeper into the capabilities of both the Keyword-Matching approach and the Machine Learning approach, utilizing TF-IDF for document classification.

These works contribute to the field of term weighting and provide valuable insights into enhancing the effectiveness of recommendation systems and information retrieval. Building upon these studies, my research aims to explore further and analyze various techniques for document classification on EQF (European Qualification Framework) levelled documents.

Chapter 3

Task and Data Description

3.1 Task Description

As mentioned above, the main task of this thesis is document classification on the European Qualifications Framework (EQF) levels, specifically aiming to predict the Dutch EQF level (NLQF). Now, concerning the specific task at hand it involves developing and evaluating two distinct classification methods: a Keyword Matching Approach and a Machine Learning approach. The Keyword Matching Approach utilizes the term frequency-inverse document frequency (TF-IDF) with n-grams to match documents in the test data against lists of salient phrases with different lengths per EQF level. In other words, these lists of salient phrases formed by using TF-IDF with n-grams act as a model, and they classify the documents in the test data according to the matched phrases to a particular level, which is further explained in the "Methodology" section below. On the other hand, the Machine Learning Approach employs TF-IDF with n-grams and Logistic Regression to learn patterns and relationships in the data for classification. Simply, my approach involves training both a Machine Learning and a rule-based system using a dataset where documents are labelled with specific categories. These algorithms analyze the documents using TF-IDF (Term Frequency-Inverse Document Frequency) to learn underlying patterns, characteristics, and connections within the text. With this knowledge, they can predict the category or class of new, unseen documents.

3.2 Data Description

The dataset used in this thesis consists of English versions of Latvian, Maltese and Dutch documents. Also, there is one additional Machine-Translated Swedish dataset, which is translated into English. Hence, the complete dataset comprises documents in Latvian, Maltese, Swedish, and Dutch, with their corresponding English versions employed in this research.
In the subsequent sections, these datasets will be referred to as Latvian, Maltese, Swedish, and Dutch datasets to specify the datasets under discussion. It is essential to note that although these datasets are labelled with the names of respective countries, the contents of the documents are in English rather than in the individual languages of Latvian, Maltese, Swedish, and Dutch.
It should be highlighted that each of these datasets is separate from the others as they are web-scraped individually by EDIA from the European Union. The sole dataset

for which I have a readily available link is the NLQF dataset, accessible through the following URL: https://database.nlqf.nl/search?open=all
These documents are labelled with their respective EQF levels, encompassing various qualifications, including academic degrees, vocational certifications, and professional designations. The documents cover various fields and disciplines, representing the breadth of qualifications within the EQF framework.
Although English versions of the datasets are used in this study, the wording and descriptions of the qualifications can differ per country. This allows us to investigate whether a system trained on EQF levelled documents from various countries can effectively predict the NLQF documents.
By leveraging this dataset and conducting experiments on document classification using the keyword matching approach and machine learning approach, I aim to derive insights into the performance, strengths, and limitations of these methods in the context of EQF-level classification. The results obtained from these experiments will pave the way for further research in the field of educational NLP and qualification recognition.

Organization of the dataset

The datasets include title, description of qualification and the corresponding EQF level of that particular title and description. The datasets are documents describing qualifications. In other words, a document contains the contents necessary to achieve a particular qualification, and each qualification is tied to an EQF level. To illustrate, a title called “Construction engineer” has a corresponding column called “description”, which includes descriptive explanations a construction engineer has or should have, such as “methods for supplier assessments, cost breakdowns and product calculation”.

	title	eqf_level_id	description
0	Quality assurance and testers in IT	5	• The IT industry's various areas within quality control and testing
1	Construction engineer - construction	5	. cost and production planning, production economics and profita
2	UX Designer	5	- About the connection between human behavior and experience
3	Solar technician	5	. constituent components of a solar power system . solar panel co
4	Solar technician	5	. constituent components of a solar power system . solar panel co
5	Wood sculptor/Wood carver/Wood craftsman	5	. Specialized knowledge of wood sculpting and wood carving as
6	Construction engineer railway	5	. the railway system from a technical, economic, environmental ar

Figure 3.1: A sample distribution of the training data

The distribution of instances

The total dataset set has 14725 documents. The main issue of the datasets is the imbalance in the level sizes. To illustrate, Dutch data has few instances in levels 1, 5, 6, and 7 and no instances in level 8, whereas levels 2, 3, and 4 have quite a lot of instances compared to the other levels. Similarly, in the Swedish dataset, the levels are imbalanced. While there are 3055 instances in level 5, there are just a few or no instances in other levels. Overall, in levels 1,2 and 8, all of the datasets have relatively fewer instances comparing to the other levels per dataset. The distribution of the levels are represented in Table 3.1.

The table 3.1 below displays the raw dataset instances categorized by their respective levels without undergoing any preprocessing. In the subsequent section, these datasets will be merged based on the training systems outlined for this study.

Levels	Swedish Dataset	Latvian Dataset	Maltese Dataset	Dutch Dataset	Total Dataset
1	0	4	183	34	221
2	0	29	353	248	630
3	4	239	548	323	1114
4	28	533	991	377	1929
5	3055	77	1916	17	5065
6	74	578	1880	61	2593
7	0	685	2316	3	3004
8	0	137	32	0	169

Table 3.1: Distribution of instances per level

3.2.1 Merging datasets

As mentioned above, although the datasets are separated from each other, they are all EQF-level labelled, and they consist of qualification descriptions. In order to observe the performances of the datasets on different test datasets, two different combinations of training datasets are implemented. The rationale behind this lies in my anticipation that the specific language and terminology used within each dataset may potentially impact the accuracy of predictive outcomes. In one of the training sets, there are no instances from the Dutch dataset as in the evaluation. The Dutch data is used as the test data to observe the performance of the Dutch dataset on a non-dutch training system. The other training dataset includes the Dutch dataset as well in both training and test data. Therefore, two different merging processes occurred, which are experimented on both Machine Learning and Keyword-Matching approaches. To clarify, **Swedish, Latvian and Maltese datasets are merged** to create one training dataset (which will be referred as **"non-dutch dataset"** in the following sections) to test the performance of the Dutch dataset and the other merged dataset (which will be referred as **all-datasets-combined**) includes **Dutch dataset along with Swedish, Latvian and Maltese datasets** to observe if including instances from all datasets in both training and test dataset show better or worse performances.
In summary, the "non-dutch dataset" is used to evaluate the performance of the Dutch dataset, while the "all-datasets-combined" training system aims to determine whether including instances from all datasets in both training and test datasets enhances or reduces classification performance.

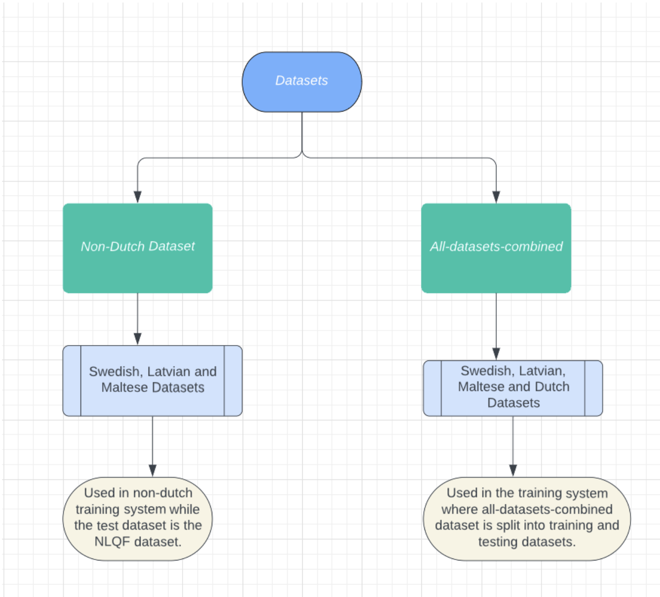

Figure 3.2: The figure above describes after merging the datasets on which systems each of them is used.

3.2.2 Splitting the data into training, testing and development datasets

The reason for creating a development dataset is to use a dataset for developing lists, testing the methodologies promised and improving the results without using the test data. For this task, the development dataset is crucial for evaluating the outcomes produced by the systems utilizing different combinations of n-gram and salient phrases lists to determine the best-performing systems and to develop them. The list comprises words/phrases extracted from TF-IDF with n-grams in the keyword-matching approach, and these words/phrases are particularly relevant to specific document classes. Given there are two experiments for each method as mentioned in the above subsection, in order to test the systems and improve them, two development data sets are formed. One of which is extracted from the "non-dutch dataset", whereas the other development dataset is extracted from the "all-datasets-combined dataset". The reason for that is for the non-dutch training system. The development dataset should not include instances similar to the Dutch dataset. Put simply, the inclusion of a "non-Dutch dataset" as training data serves the purpose of evaluating the Dutch dataset's performance on EQF descriptions from other countries. Therefore, the development dataset should exclude any Dutch (NLQF) instances. This is crucial to determine the effectiveness of different systems when trained on non-Dutch data. The reason is that the development data is utilized to identify the most effective system(s), and including NLQF instances would undermine this goal, as the system's performance would be assessed based on a dataset containing NLQF instances.

The second development set is extracted from the already-merged "all-datasets-combined" set, and this dataset will be used to develop the "all-datasets-combined training system", which includes the Dutch dataset along with the other datasets.
To sum up, one of the development datasets excludes only the Dutch to develop the non-dutch training system, and the other development dataset includes all Swedish, Lat-

vian, Maltese and Dutch datasets to develop the all-datasets-combined system. Also, it is important to note that *both of the development datasets have 50 instances per level.*

The distribution of instances per level in both systems
The following tables display the distribution of instances per level. However, before training, the documents in the training dataset are combined per level. As a result, each class in the training dataset contains only a single document, where all documents per class are merged into one compiled document. *The purpose of these tables is to showcase the size of each class.*

Level	Training	Development	Evaluation
1	137	50	34
2	332	50	248
3	741	50	323
4	1502	50	377
5	4998	50	17
6	2482	50	61
7	2951	50	3
8	119	50	0

Table 3.2: The table represents of the distribution of instances in the **non-dutch training system**.

Level	Training	Development	Evaluation
1	97	50	74
2	376	50	204
3	741	50	323
4	1308	50	571
5	3464	50	1551
6	1786	50	757
7	2075	50	879
8	60	50	59

Table 3.3: The table represents of the distribution of instances in the **all-datasets-combined training system**.

3.3 Models and Tools

3.3.1 Logistic Regression

Logistic Regression Although Logistic Regression is a popular statistical model used for binary classification problems, it models the relationship between a set of input variables (features) and an outcome variable (target) by estimating the probability of the outcome belonging to a specific class.
The logistic regression model uses the logistic function, also known as the sigmoid function, to map the linear combination of input variables to a probability value between 0 and 1 (Kanade, 2022). For this reason, Logistic Regression can learn the relation between the input and the target while it can also learn that a specific input has no

relation with another class. To elaborate, the Logistic Regression model aims to estimate the coefficients that maximize the likelihood of observing the given set of training data. With Logistic Regression, by examining the learned weights of the features, one can identify the specific terms or n-grams that have a significant influence on the classification outcomes. This transparency not only aids in understanding the underlying mechanisms of the model but also enables domain experts to validate the relevance of the selected features. Furthermore, the explainability of logistic Regression promotes trust, accountability, and regulatory compliance in document classification tasks, making it a suitable choice for research and practical applications. Furthermore, during training, the model coefficients are typically estimated using optimization algorithms such as maximum likelihood estimation (MLE) or gradient descent (Brownlee, 2019). Once the coefficients are obtained, they can be used to make predictions on new data by calculating the probability of the positive class using the logistic function.

Expected Performance

Considering the above-mentioned factors, I expect the Logistic Regression model to show better results than the keyword-matching approach. The reason for this is that a model can learn more about the mapping, whereas a keyword-matching approach relies more on the rules given to classify a given data.

3.4 TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF is a numerical statistic used to measure the importance of a term within a document or a collection of documents. As shown below, it combines the notions of term frequency (TF) and inverse document frequency (IDF) to assess the relevance of terms. TF (Term Frequency) measures the frequency of a term t in a document d (Hamdaoui, 2019).

$$TF(t,d) = \frac{\text{number of occurrences of term } t \text{ in document } d}{\text{total number of terms in document } d}$$

IDF (Inverse Document Frequency) measures the rarity of a term t across a collection of documents. It is calculated as:

$$IDF(t) = \log\left(\frac{\text{total number of documents in the collection}}{\text{number of documents containing term } t}\right)$$

The TF-IDF score for a term t in a document d is obtained by multiplying the TF and IDF values:

$$TF\text{-}IDF(t,d) = TF(t,d) \times IDF(t)$$

The intuition behind TF-IDF is that a term is considered important if it appears frequently within a document (high TF) and is rare across the entire document collection (high IDF). Terms with higher TF-IDF scores are thus assumed to be more relevant or representative of the content of the document (Simha, 2021).
TF-IDF is widely used in various natural language processing tasks such as document classification, information retrieval, and text mining. It allows for the identification

of key terms and helps to prioritize terms that contribute significantly to the meaning of a document or distinguish it from others in the collection (MarketBrew, n.d). In this study, this feature of TF-IDF is essential. The extracted phrases are salient to each level relative to the whole corpus. To elaborate, the TF-IDF scores are calculated for each word/phrase in each document. These scores represent the importance of a word/phrase in a specific document relative to the entire dataset. When classifying a new document, it calculates a weighted score for each class by considering the TF-IDF scores and matching words/phrases in the document with the top words associated with each class. The class with the highest weighted score is assigned as the predicted class. Thus, by combining the TF-IDF scores across all documents within a particular class, the system obtains the combined scores for each word/phrase in that class. Therefore, in the classification process, TF-IDF enables the system to distinguish between the classes by using those combined scores.

3.5 N-grams

N-grams, in the context of natural language processing and text analysis, are contiguous sequences of n items, typically words, extracted from a given text corpus. They serve as a fundamental unit for capturing contextual information and understanding the relationships between words within a language.
Unigrams represent individual words and offer basic frequency distribution information. For instance, in the sentence “I love to eat apples,” the unigrams would be ”I,” ”love,” ”to,” ”eat,” and ”apples.” (Zakirizvi, 2023).
Bigrams consist of pairs of consecutive words and enable the capture of contextual relationships between adjacent words. In the previous example, the bigrams would be ”I love,” ”love to,” ”to eat,” and ”eat apples.”
Trigrams, on the other hand, encompass sequences of three consecutive words, providing a richer context by considering longer phrases or expressions. In the same example, the trigrams would be ”I love to,” ”love to eat,” and ”to eat apples.”
N-grams are employed in various tasks such as language modelling, information retrieval, machine translation, and text classification. They are generated by sliding a window of size n over a given text and extracting the resulting n-gram sequences (Chandravanshi, 2021). An n-gram can vary in length, with commonly used types including unigrams (1-grams), bigrams (2-grams), trigrams (3-grams), and so forth. Each type provides a different level of granularity in analyzing text. As cited in Nithyashree (2021), n-grams are *”neighbouring sequences of items in a document”*. The author states the reason why ”we need many different types of n-grams” ”is because different types of n-grams are suitable for different types of applications” and to arrive at a confident conclusion about the most effective n-gram approach for text analysis, it is necessary to experiment with different n-gram sizes depending on the specific task at hand. By trying out various n-gram configurations and evaluating their performance, we can gain a clearer understanding of which n-gram size yields the best results for the given text analysis task. Given that, in this study, various n-gram sizes are implemented to observe the best-performing n-gram size. Using TF-IDF with n-grams is expected to increase the quality of the training because TF-IDF with n-grams detects the salient n-grams per level, which provides a more specific feature to the classification process. In other words, by using the weighted score feature of TF-IDF, which also sorts out the most salient phrases/words per class with different sizes of n-grams,

this method provides information to the system where it can learn more class-specific words/phrases to classify the documents.

Chapter 4

Methodology

The methodology consists of two classification approaches: TF-IDF Keyword-Matching Approach and Machine Learning Classification Approach (using Logistic Regression).

Preprocessing the dataset
Prior to commencing the classification procedure, the data underwent a thorough cleaning process that encompassed the removal of extraneous words, references, stop-words, and non-English sentences (notably, the Latvian dataset contained both English and Latvian sentences). Additionally, the data was tokenized and lemmatized. Subsequently, during the subsequent preprocessing phase, all datasets were standardized to achieve uniformity. This entailed aligning the datasets to a consistent format, encompassing columns for titles, descriptions, and levels.
After that, as part of the second stage of preparation, I organized the datasets in a way that made them consistent. This ensured that by the end, all the datasets shared the same structure, including having matching titles, descriptions, and level columns.

While the Keyword-Matching Approach and Machine Learning Approach are the main methods employed in this study, each of them involves two distinct experiments. Notably, there exist two different systems within each method, distinguished by their unique training and testing datasets. As a result, both the Keyword-Matching Approach and the Machine Learning Approach encompass separate experiments for these two systems.
To elaborate, the first system employs the *Non-Dutch Dataset* as described in 3.2, comprising only the Swedish, Latvian, and Maltese datasets for training. Subsequently, the NLQF dataset serves as the test data for this system.
The second training system incorporates all datasets for both training and testing purposes. The dataset, referred to as *All-Datasets-Combined* in the table, includes data from Swedish, Latvian, Maltese, and Dutch datasets. As stated earlier, this dataset is split into separate training and testing datasets. Consequently, both the training and testing datasets encompass instances from all the datasets.

To enhance the clarity of these systems and their distinctions, the following tables provide an overview of instance distribution within the training and testing sets for both approaches.

4.1 Keyword-Mathcing Approach

There are two kinds of training. One training consists of training the non-dutch dataset, developing it with the non-dutch development dataset (Development set 1) and running it on the test set. The other training is done by using the all-datasets combined dataset. This dataset is split into training and testing sets, and to develop this system, all-datasets-combined development set (development set 2) is used. Therefore, the former training is to test how well the NLQF dataset performs on the training of other countries' EQF descriptions, and the latter-mentioned training is to observe if the training shows a difference when the training and testing contains NLQF dataset along with other countries' EQF descriptions.

The training starts with applying the preprocessing mentioned above. Then, each training dataset is run on all different sizes of n-grams.

As briefly explained before, different n-gram lengths are used to extract the most salient words and phrases for each level. Given the dataset is about qualification levels, the levels generally include phrases like "Bachelor's degree", "Secondary School", and/or "Engineering". For this reason, extracting keywords as unigrams, bigrams, trigrams and combinations of unigram-bigrams, bigram-trigrams should enable me to observe which n-grams show better results for this particular task. Extracted keywords are added to a list of 20, 55, 70, and 1000 words/phrases, which consists of words/phrases of the most salient words/phrases per level. Starting from small to higher sizes of lists, performance differs. Therefore, given the size of the list can be any number and it is not useful to state each of their results in this study, the sizes that start showing big differences are selected to be tested to decide which size shows better performances. In other words, the above-mentioned sizes show significant differences between each other. However, size 10, 20, 25, and 30 did not show a big difference against each other, whereas size 20 and 55 showed considerable differences (likewise, the rest of the sizes mentioned above). Therefore, 20 and 55 are two of the parameters used to observe which size range shows better performances. This formed list is used for the classification only in the Keyword-Matching Approach.

The classification process involves utilizing a predefined list of words and phrases as a reference point. This list serves as a checklist to examine how well each document aligns with specific classes. The system computes a score based on the prevalence of these words and phrases. In essence, the words and phrases in the list hold designated saliency scores within the training system. During classification, the system evaluates how many words and phrases in each document from the training data align with those in the list. For each document we want to classify, the script calculates a weighted score for each class using a keyword-matching approach. It calculates the similarity between the TF-IDF vector of the test sentence and the average TF-IDF vector of documents in each class from the training data. It then finds the top words associated with each class and checks how many of those words are present in the test sentence. The weighted score is calculated by multiplying the similarity score by the ratio of matching words to the total top words for each class.

In cases where there's a tie in matching, the system analyzes the words and phrases with the highest assigned scores. Subsequently, it assigns the document to the corresponding

level, considering that higher scores signify heightened relevance to that specific level. To clarify, if a document corresponds to both class 1 and class 2, the system evaluates the word and phrase weights associated with each class. By prioritizing the phrases with higher weights between the classes (indicative of greater relevance), the system then allocates the document to the appropriate level.
Hence, the system establishes the predicted class by identifying the class with the most significant similarity score. In summary, the classification mechanism draws upon word and phrase lists, evaluating their presence and relevance in documents to assign the most suitable class. To illustrate, if a document matches with class 1 and class 2 at the same time, the system checks the weights of the words/phrases which match with class 1 and class 2; then by considering the higher-weighting phrases between classes (which means it is more salient to that document), it assigns the document to that level. Thus, the system determines the predicted class by selecting the class with the highest similarity score.

4.1.1 Non-Dutch training system and All-Datasets-Combined System

The above-mentioned procedure is applied to both of the training datasets. However, in the non-dutch training system, the training is done on the non-dutch dataset, and this system is developed by using the non-dutch development dataset. To provide a fair evaluation, the non-dutch training system is taken as the baseline system, and the sizes of the lists are set to be the same (20, 55, 70, 100, and 1000) in both *non-dutch* and *all-datasets-combined* systems. However, to observe which size(s) of the n-grams show(s) better performances, non-dutch development data (development set 1) is used on the non-dutch training system and all-datasets combined development data (development set 2) is used on the system where all dataset are used in the training.

4.2 Machine Learning

I also conducted an experiment using Logistic Regression, and if time allows, Support Vector Machine (SVM) model will also be implemented.

This method is developed to compare the results of the keyword approach to model classification. The motivation to compare the keyword-matching approach and machine learning approach is to observe which method performs better in this particular task. The classification process involves utilizing the descriptions of qualifications as input and predicting their corresponding EQF levels as output, similar to the Keyword-Matching approach. This classification procedure employs the same training datasets, namely the non-dutch training system and the all-datasets-combined system. To facilitate this, vectorization is executed using TF-IDF (Term Frequency-Inverse Document Frequency) with various n-gram configurations, including unigrams, bigrams, unigram-bigrams, trigrams, and bigrams-trigrams. The objective is to compare the performance of Machine Learning classification using different n-gram settings and determine which n-gram combinations are more effective in the Machine Learning Classification context. It's important to emphasize that both approaches utilize identical datasets and employ the same size of n-grams with TF-IDF. This deliberate choice aims to establish an equitable comparison between the two approaches, enabling the observation of their respective performances. The key distinction between these methods lies in the utilization of a model, specifically Logistic Regression, instead of relying solely on the list of most significant n-gram phrases (as seen in the Keyword-Matching approach) for the purpose of classification. Furthermore, it's worth noting that the system does not incorporate a predefined list. Instead, the complete output generated by TF-IDF with various n-gram sizes is utilized. In contrast to the keyword-matching approach, this method doesn't involve a matching-words phase. To elaborate, the classification process within the Machine Learning approach is facilitated by a model that possesses the capability to comprehend the sequential patterns of word combinations. Consequently, all the phrases extracted through TF-IDF, encompassing different n-gram dimensions, are inputted into the model during the classification process. The model autonomously learns the inherent patterns and subsequently classifies the test data accordingly.

Chapter 5

Results

Each of the following sections describes one particular experiment conducted on both Keyword-Matching and Machine Learning Approaches.

1- The first section (5.1) shows the results of the **Keyword-Matching Approach** including both experiments: non-dutch training dataset run on the Dutch (NLQF) dataset (test dataset) and all-datasets-combined system run on the test set).

2- The second section (5.2) shows the results of the **Machine Learning Approach** including both experiments: non-dutch training dataset run on the Dutch (NLQF) dataset (test dataset) and all-datasets-combined system run on the test set).

The results section first shows the performances of the systems on the development dataset. Then, according to the results on the development dataset, best performing systems are identified, and only the best-performing systems' results are run on test data and added to the results section.
The assessment in the following sections will be based on weighted average F1-scores, which are highlighted in bold, emphasizing the most effective F1-scores per n-gram as presented in the subsequent tables. This assessment involves calculating the F1 score for each class individually and then deriving a weighted average of these F1 scores, taking into account the instance counts specific to each class. It assigns greater significance to classes with a larger number of instances, a technique recommended in data science practices (Leung, 2022). This weighted approach is particularly valuable when dealing with datasets characterized by imbalances, like the one used in this study.

5.1 Results of Keyword-Matching Approach

5.1.1 Results of the Non-Dutch Training System

Results of the Keyword Matching Approach on the *development* dataset

This section shows the results of the non-dutch training system (see section 3.2). As mentioned above, first, the results of the system on the development dataset are presented in the table 5.1 below.

Keyword approach (parameters)	list 20	list 55	list 70	list 100	1000
unigram	P:0.504 R:0.525 F1:0.514	P:0.483 R:0.517 F1:0.499	P:0.466 R:0.502 F1:0.484	P:0.519 R:0.537 F1:0.528	P:0.607 R:0.58 **F1:0.593**
bigram	P:0.124 R:0.127 **F1:0.126**	P:0.109 R:0.11 F1:0.109	P:0.110 R:0.11 F1:0.110	P:0.126 R:0.127 **F1:0.126**	P:0.094 R:0.092 F1:0.093
trigram	P:0.148 R:0.145 **F1:0.146**	P:0.114 R:0.112 F1:0.113	P:0.112 R:0.112 F1:0.112	P:0.137 R:0.137 F1:0.137	P:0.115 R:0.112 F1:0.114
unigram+bigram	P:0.522 R:0.535 F1:0.528	P:0.553 R:0.54 F1:0.546	P:0.548 R:0.52 F1:0.533	P:0.595 R:0.55 F1:0.571	P:0.656 R:0.645 **F1:0.650**
bigram+trigram	P:0.104 R:0.107 F1:0.105	P:0.111 R:0.112 F1:0.111	P:0.138 R:0.14 **F1:0.139**	P:0.0.80 R:0.082 F1:0.081	P:0.115 R:0.115 F1:0.115

Table 5.1: The above table shows the evaluation metrics of the Keyword-Matching approach with **non-dutch** training dataset. The dataset is run on the **non-dutch development dataset.**

Results of the Keyword Matching Approach on the *test* (NLQF) dataset

The table 5.2 includes only the results of unigrams+bigrams as the best working systems according to the results of the development dataset shown in table 5.1 which shows that in all sizes of lists, unigrams+bigrams systems performed better than the other sizes of lists.
Table 5.3 represents the results of the non-dutch training system on the test dataset with the list size of 1000 because considering the results on the development set, the first two best working systems in the non-dutch training system are in the list size of 1000.

Keyword approach (parameters)	list 20	list 55	list 70	list 100	1000
unigram + bigram	P:0.32 R:0.12 **F1:0.14**	P:0.31 R:0.15 **F1:0.15**	P:0.26 R:0.13 **F1:0.14**	P:0.28 R:0.11 **F1:0.12**	P:0.20 R:0.04 **F1:0.04**

Table 5.2: The above table shows the performance of the Keyword-Matching approach with **non-dutch** training system. The dataset is run on the **test dataset which is NLQF (dutch) dataset.**

	list 1000
unigrams	P:0.31 R:0.08 **F1:0.10**
unigram+bigrams	P:0.20 R:0.04 **F1:0.04**

Table 5.3: The table shows the results of the best working non-dutch systems (according to the size of list) on **test dataset.**

In this system, list size 20 on the development dataset shows that the best working system is unigram+bigrams with 0.52 f1-score. However, when the same size of the list and the same size of n-grams are run on the test dataset (non-dutch dataset), it shows an F1-score: 0.07 f1-score. While the size of the list is 55, unigrams showed 0.499 F1 score (see Table 5.1) on the development dataset. On the test data (Dutch data), it showed 0.15 F1 score. One of the reasons for that is the development dataset has no Dutch instances. For this reason, it is expected for the system show a better performance on the development dataset than the NLQF dataset. The reason for that is the wordings and terminology are likely to be similar given the development dataset is extracted from the non-dutch dataset and includes instances from Swedish, Latvian and Maltese datasets like the training dataset.

5.1.2 Results of the All-Datasets-Combined System

Results of the Keyword Matching Approach on the development dataset

This section shows the performances of the all-datasets-combined training system (see section 3.3). The dataset is split into training and testing sets. Therefore, in both training and testing, there are instances from all datasets.

Keyword approach (parameters)	list 20	list 55	list 70	list 100	1000
unigram	P:0.529 R:0.511 F1:0.520	P:0.515 R:0.505 F1:0.510	P:0.556 R:0.53 F1:0.542	P:0.571 R:0.532 **F1:0.551**	P:0.517 R:0.543 F1:0.530
bigram	P:0.095 R:0.097 F1:0.096	P:0.108 R:0.107 F1:0.108	P:0.095 R:0.095 F1:0.095	P:0.123 R:0.122 **F1:0.122**	P:0.108 R:0.11 F1:0.109
trigram	P:0.114 R:0.117 F1:0.116	P:0.150 R:0.142 F1:0.141	P:0.117 R:0.122 F1:0.119	P:0.145 R:0.145 **F1:0.145**	P:0.119 R:0.12 F1:0.119
unigram+bigram	P:0.524 R:0.517 F1:0.521	P:0.589 R:0.545 F1:0.566	P:0.596 R:0.54 F1:0.566	P:0.589 R:0.53 F1:0.558	P:0.585 R:0.555 **F1:0.569**
bigram+trigram	P:0.135 R:0.131 F1:0.133	P:0.114 R:0.115 F1:0.114	P:0.139 R:0.14 **F1:0.139**	P:0.0.80 R:0.082 F1:0.081	P:0.114 R:0.117 F1:0.115

Table 5.4: The above table shows the evaluation metrics of the Keyword-Matching approach with **all-datasets-combined** training dataset. The dataset is run on the **all-datasets-combined development dataset.**

Results of the Keyword Matching Approach on the test dataset

As mentioned above, according to the results of the systems run on the development dataset, only the best-performing systems are run on the test dataset. Firstly, the best working n-gram size, unigram+bigram, is run on the test set, which is shown in the table 5.5. Following that, the best-performing systems, according to the size of the lists, are identified and run on the test set as well.

Keyword approach (parameters)	list 20	list 55	list 70	list 100	1000
unigram + bigram	P:0.58 R:0.56 **F1:0.57**	P:0.60 R:0.56 **F1:0.58**	P:0.63 R:0.57 **F1:0.59**	P:0.62 R:0.57 **F1:0.59**	P:0.65 R:0.59 **F1:0.59**

Table 5.5: The table above shows the performance of the Keyword-Matching approach with **all-datasets-combined** training system. The dataset is run on the **test dataset**.

	list 100	list 1000	results on test data
unigram	X		P:0.62 R:0.52 F1:0.54
unigram+bigram		X	P:0.65 R:0.59 F1:0.59

Table 5.6: The results of the all-datasets-combined system, the best working systems according to the size of the list on the test dataset. Unigrams show the results on the test data with the list size 100 and unigram+bigrams at 1000.

The table presented above (Table 5.6) displays the **best-working systems** based on the performance outcomes outlined in Section 5.4.

5.2 Results of the Machine Learning Approach

The methodology is also tested with the Machine Learning Method. The same training, testing and development datasets used in the keyword-matching approach are used in the Machine Learning Approach. The reason for that is to provide a fair comparison between the two methodologies.

Results of Machine-Learning Approach on development datasets

	unigram	**bigram**	**trigram**	**unigram +bigram**	**bigram +trigram**
non-dutch system	P:0.575 R:0.5 F1:0.535	P:0.647 R:0.55 F1:0.594	P:0.694 R:0.577 **F1:0.630**	P:0.601 R:0.505 F1:0.548	P:0.670 R:0.57 F1:0.61
all-datasets-combined system	P:0.754 R:0.7 F1:0.726	P:0.813 R:0.667 F1:0.733	P:0.830 R:0.617 F1:0.708	P:0.787 R:0.725 **F1:0.755**	P:0.824 R:0.665 F1:0.736

Table 5.7: The table above shows the experiments on the Machine Learning method. The first row shows the performance of the n-grams on the non-dutch training dataset based on the non-dutch development dataset. The second row shows the results of all datasets combined system on the second development set, extracted from the all-datasets-combined set.

Results of the Machine Learning Approach on the test sets

	results on test data
trigram (non-dutch system)	P:0.13 R:0.04 **F1:0.01**
unigram+bigram (all-datasets-combined system)	P:0.65 R:0.38 **F1:0.41**

Table 5.8: The best working systems of the Machine Learning Method on their related test datasets.

Chapter 6

Error Analysis

6.0.1 Error Analysis of the Keyword-Matching Approach

The Non-Dutch System with Unigrams in the List Size of 1000

In the Keyword-Matching Approach, one of the best-working systems was determined to be with the unigrams in the non-dutch training system. However, the weighted average for this system exhibits a rather low value of 0.13 on the test dataset. The system's analysis reveals that Class 1, 2, 3 and 4 exhibited the poorest performance when compared to the other classes in terms of the weighted average f1-score (see table 6.1 below). While it's understandable for classes 1, 2 and 3 to exhibit lower performance, given their limited representation in the training data, the performance of class 4 is unexpected, considering it has quite a larger amount of training data compared to the other classes. To illustrate, a document including these instances "advanced road transport planner", "work planning department", and "professional freight transporter" is predicted to be class 5 while its actual class is 4. The matching words of this system include words such as: "professional, advanced, management..." which are the wordings likely to be close to higher-level qualification terms. Evidently, when we examine the weighted scores, we observe that the predicted class 5 received a score of 0.005449, which is slightly higher than the score for the actual class, which stands at 0.004202.

Similarly, a class 5 document including instances as: "monitor development field", "advanced equipment technology", and "necessary performance work" is classified as class 4. The model's predicted class receives a slightly higher weighted score for class 4 (0.0075) compared to the actual class (class 5) (0.0068) in the classification. Hence, even though the weighted scores for the actual classes are relatively high, the system tends to mix up these classes due to the vocabulary similarities in their qualification descriptions.

When the confusion matrix is investigated (see Figure 6.1), it is clear that there is a mix-up with the classes which are close to each other. To elaborate, Classes 2, 3 and 4 are mostly misclassified among each other. Meaning that the system has challenges distinguishing between these classes. As mentioned above, the descriptions in classes that are in close proximity to each other, like class 4 and class 5, are more likely to exhibit similar wording. Also, The Confusion Matrix illustrates that the majority of classes face misclassifications, with a notable concentration of errors occurring between classes 4 and 5. This is particularly evident in the misclassifications where other classes are often predicted as class 5. This observation underscores the impact of the dataset's class imbalance, which leads the model to exhibit a bias towards the predominant class, which is class 5. For instance, class 1 has 137 training instances while class 5 has 4998, and as can be seen in the Confusion Matrix, Class 1 is mostly mispredicted as class 5 rather than class 1. This shows that class 1 is not well learned by the system, and in that case, class 1 is prone to be classified as class 5, which forms the majority of the training dataset.

Classification Metrics

Class	Precision	Recall	F1-Score	Support
1	0.00	0.00	0.00	34
2	0.50	0.00	0.01	248
3	0.33	0.17	0.22	323
4	0.24	0.05	0.08	377
5	0.01	0.65	0.03	17
6	0.14	0.07	0.09	61
7	0.07	0.33	0.12	3
8	0.00	0.00	0.00	0
Accuracy			0.08	1063
Macro Avg	0.16	0.16	0.16	1063
Weighted Avg	0.31	0.08	0.10	1063

Table 6.1: Classification Report of the Non-Dutch system with unigrams on the test set (list size 1000)

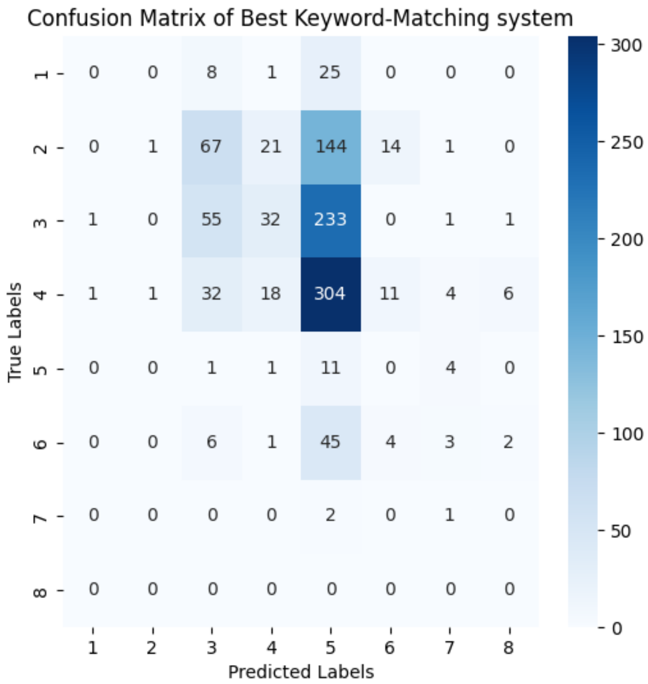

Figure 6.1: The confusion matrix of the non-dutch training system with **unigrams** in the list size of 1000.

Non-dutch System with Unigram+Bigrams in the List size of 1000)

In this experiment, the system showed a 0.650 f1-score on the development dataset, while this performance dropped to 0.04 when run on the test dataset. As can be seen, this score is lower than expected.

When observing the Classification Report below 6.2, in terms of the weighted average F1-score, classes 2, 3, and 4 exhibit the lowest performances. Similar to the previous system above, there is a bias towards class 5. It has been noticed that there is a significant amount of misclassification between classes 2, 3, and 4, particularly with class 5 (see figure 6.2). While it's understandable that classes 4 and 5 could be confused due to their terminological similarities, it's quite surprising to see such a high level of misclassification from class 2 to class 5.

To elaborate, in one of the class 2 instances: The document includes instances such as "production technology employee", "work location", "client within company outside company", "he/she generally work production hall", "workshop make product part", and its actual class is "2". However, it is predicted to be class "5".

Because the weighted score for class 5 is slightly higher, it is assigned to class 5.

The actual class' weighted score is 0.003822, and the predicted class' weighted score is 0.007710. When the matching words for the actual class are observed to be *"independently", "technical supervision", "technology" and "manufacturing"*, it is figured to be words which are likely to occur more in higher level classes such as class 5. To illustrate, in another instance where the actual class is 5, the salient words list include "management", "professional", "design project", "policy", "construction", and "specialist".

Classification Metrics

Table 6.2: Classification Metrics

Class	Precision	Recall	F1-Score	Support
1	0.01	0.01	0.01	34
2	0.33	0.00	0.01	248
3	0.28	0.08	0.12	323
4	0.10	0.01	0.01	377
5	0.02	0.82	0.03	17
6	0.04	0.02	0.02	61
7	0.10	0.33	0.15	3
8	0.00	0.00	0.00	0
Accuracy			0.04	1063
Macro Avg	0.11	0.16	0.04	1063
Weighted Avg	0.20	0.04	0.04	1063

Table 6.3: Classification report of Non-Dutch system with unigram+bigram (list size 1000)

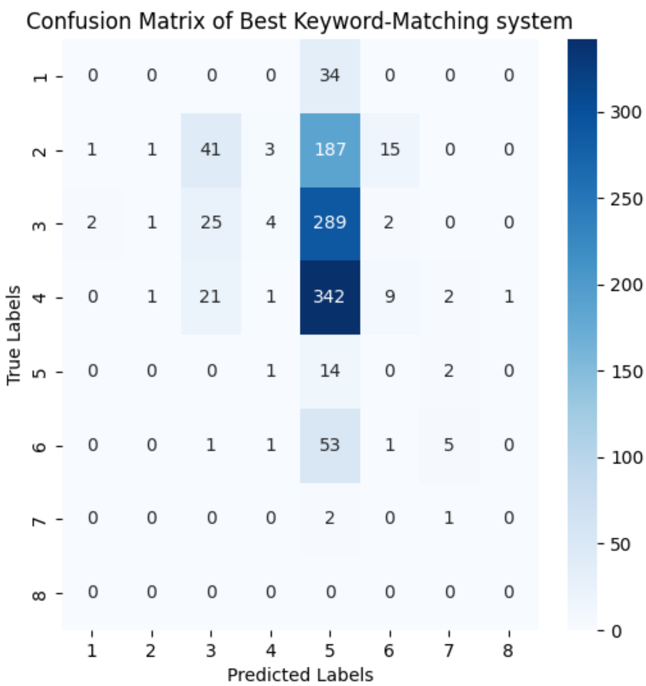

Figure 6.2: The confusion matrix of the non-dutch training system with **unigram+bigram** in the list size of 1000.

The All-Datasets-Combined system

The All-Datasets-Combined System with Unigram+Bigrams in the list Size of 1000

One of the best systems in this study is observed to be the unigram+bigrams system in the list size of 1000. This system showed considerably high performance on the test data with the f1-score of 0.59. When analyzing the Classification Report (see Table 6.4), it can be seen that weighted average f1-scores are mostly high except for class 3, and class 8. Although according to the Confusion Matrix (see Figure 6.3) class 8 is mostly predicted correctly, it can be seen that there is a slight bias towards class 8 as even though there weren't a large number of misclassifications from each class, considerable misclassifications were still observed. While further error analysis by examining keywords is needed to investigate the reason for this error, time constraints have prevented me from conducting this analysis. However, when some of the outputs of the classes are analyzed, it shows that an instance is correctly classified as class 8 has a weighted score of 0.034189, and another instance of class 4 is mispredicted to be class 8 with a weighted score of 0.000687 while its weighted score to be predicted correctly was 0.000295. Considering this, it can be deduced that a contributing factor to certain cases being mistakenly categorized as class 8 is the low weighted average scores. This implies that these documents are assigned to a class without distinctive or significant keywords associated with that class, resulting in their erroneous assignment to class 8. To illustrate, class 3 is mostly mispredicted as class 4, which is understandable given its proximity to the actual class. However, it is also frequently mispredicted as class 8. In class 3 documents, there are some subtitles as "contact provider", "information contact", "provider", and "information", and they appeared individually as "information contact", "provider", and "information" as if each of them are one document. The matching words for class 3 are "information", "provider", and "contact", which are salient words for class 8. Therefore, when those sentences are obtained as individual documents by the system, it directly assigns them as class 8 because those words/phrases are salient words of class 8. Evidently, in another instance where the actual class is 8, similar or the same terms (e.g." competence information", and" information") appeared to be repeated often, and these documents are correctly classified as class 8. Therefore, it seems that there is a dataset-related error which resulted in a misclassification of classes as class 8.

Classification Metrics

Class	Precision	Recall	F1-Score	Support
1	0.43	0.76	0.55	74
2	0.30	0.48	0.37	204
3	0.43	0.10	0.16	323
4	0.67	0.24	0.36	571
5	0.81	0.74	0.77	1551
6	0.52	0.58	0.55	757
7	0.70	0.74	0.72	879
8	0.11	0.92	0.20	59
Accuracy			0.59	4418
Macro Avg	0.50	0.57	0.46	4418
Weighted Avg	0.65	0.59	0.59	4418

Table 6.4: Classification Report of All-Datasets-Combined system with unigram+bigrams (list size of 1000)

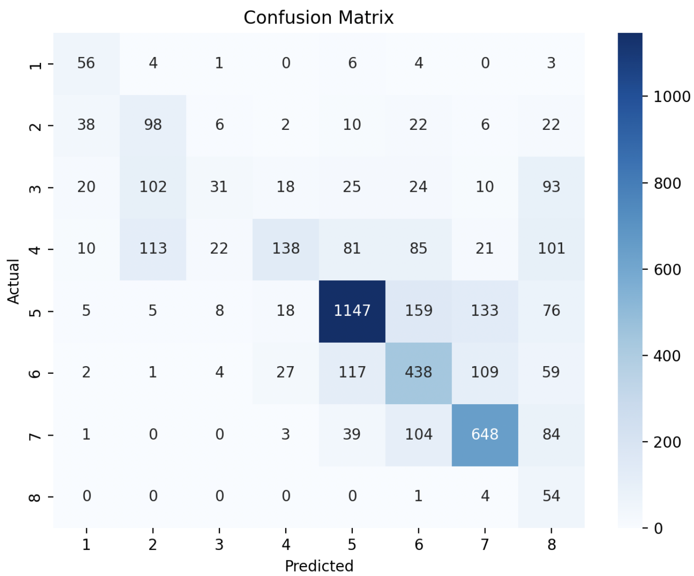

Figure 6.3: Confusion Matrix of unigram+bigram system in All-Datasets-Combined System (list size 1000)

The All-Datasets-Combined System with Unigrams in the List Size of 100

The performance of this system shows satisfying results by scoring a weighted average f1-score of 0.54. When analyzing Table 6.5, the f1-scores show high scores. Although class 5 shows some misclassifications to classes 6 and 7, out of 1551 instances, 994 of them are correctly predicted. While class 4 is mostly predicted correctly, it also shows some classes which are misclassified as class 2 and class 8. To provide an illustrative example, consider the following instances from a document: "loading unloading technique principle", "basic accounting", "necessary performance basic task professional activity level use", "international law regulation republic Latvia binding upon field logistics",

"professional term official language". In this case, the actual class assigned is class 4, a prediction accurately made by the system. This implies a close correspondence between the keywords present in the test sentence and those that typify class 4. On the other hand, an instance from class 4 is misclassified as class 8. The document includes "foreign programme awarded", "focus award contact provider information", "contact", "programme", and "information" as matching words to class 8. Also, the vocabulary variations in different datasets have affected the system's saliency list. Let us focus our attention on a particular illustrative example extracted from the training system utilizing unigrams in the list size of 100, specifically from the test data. This instance is presented in its pre-processed format.
The instance *"b.a . (hons) faculty' staple undergraduate degree. differ mainly b.a . spread equally two area study , whereas b.a . (hons) focus single area..."* is classified as **Level 1** as the matching words to the actual class are seen to be only *area, study, course*. The description itself and the actual class in the test dataset show that the document is describing level 7 qualification (Higher education); however, in the training data, "bachelor" or "bachelor degree" is explicitly written. Therefore, the system does not know "b.a" actually refers to the "bachelor degree". In short, for wordings or abbreviations as such, the system does not assign the document to level 7.

Classification Metrics

Class	Precision	Recall	F1-Score	Support
1	0.25	0.69	0.36	74
2	0.24	0.37	0.29	204
3	0.41	0.13	0.20	323
4	0.56	0.25	0.35	571
5	0.87	0.64	0.74	1551
6	0.54	0.42	0.48	757
7	0.51	0.68	0.59	879
8	0.08	0.92	0.15	59
Accuracy			0.52	4418
Macro Avg	0.43	0.51	0.39	4418
Weighted Avg	0.62	0.52	0.54	4418

Table 6.5: Classification Report of All-Datasets-Combined system with unigrams (list size 100)

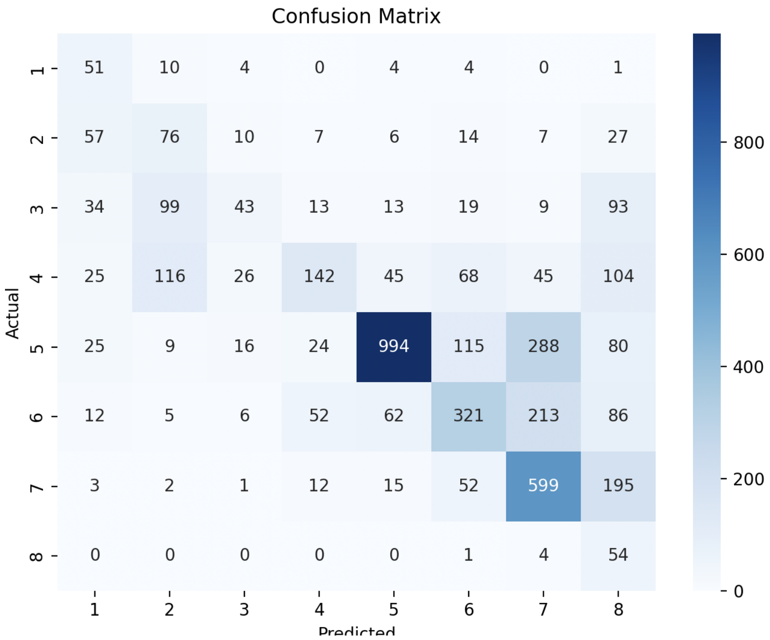

Figure 6.4: The confusion matrix of the "all-datasets-combined" system with unigram in the list size of 100.

6.0.2 Machine Learning Approach

The Non-Dutch system with trigrams

Although the non-dutch training system with the trigram system was determined to be one of the best working systems in the Machine Learning Approach, it showed quite disappointing results on the test data by showing a 0.01 weighted average f1-score. Upon examining the classification report, it becomes evident that the scores for all classes are either 0.00 (e.g., classes 2, 6, and 7) or nearly 0.00 (classes 1, 3, 4, and 5). Class 8, which lacked test data, also registers a score of 0.00. When we observe the Confusion Matrix of this system (see Figure 6.5 below), we can see that classes are mostly mispredicted to be class 1. This means that the model did not learn well about the descriptive information about the classes, and there is a bias towards class 1. Given both training dataset is imbalanced, there appears to be a generalization problem.

Class	Precision	Recall	F1-Score	Support
1	0.04	0.94	0.08	34
2	0.00	0.00	0.00	248
3	0.29	0.01	0.02	323
4	0.11	0.00	0.01	377
5	0.02	0.06	0.03	17
6	0.00	0.00	0.00	61
7	0.00	0.00	0.00	3
8	0.00	0.00	0.00	0
Accuracy			0.04	1063
Macro Avg	0.06	0.13	0.02	1063
Weighted Avg	0.13	0.04	0.01	1063

Table 6.6: Classification Report of Non-Dutch training system with trigrams in the Machine Learning method

Classification Metrics

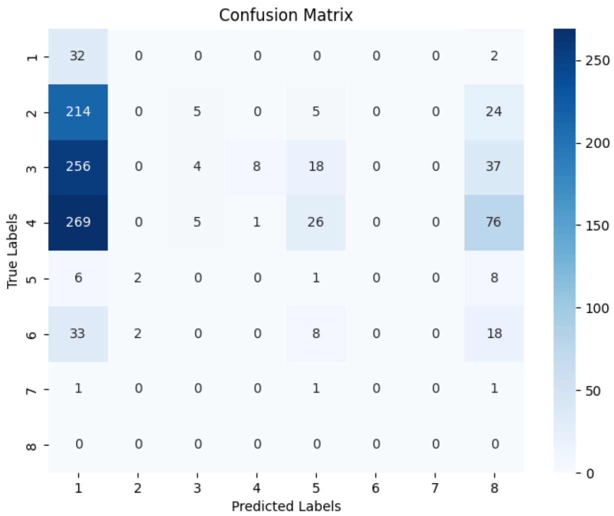

Figure 6.5: Logistic Regression on non-dutch training system run on test dataset (trigrams)

All-Datasets-Combined Training System with Unigram+Bigrams

This system shows a 0.41 weighted average f1-score, which is an acceptable score for this classification. However, the classification report (see Table 6.7) shows that the system is better at predicting class 5 with a 0.70 f1-score, whereas class 8 shows the poorest result with a 0.07 f1-score. Significantly, class 7 shows poor performance, considering the large amount of training instances provided in the dataset. When investigating the Confusion Matrix (see Figure 6.6) of this system, we can see that class 7 is mostly mispredicted as class 8. While most of the classes are accurately identified, the lower overall performance can be attributed to a significant number of misclassifications into either class 1 or class 8, for instances across all classes. These metrics indicate that the model has indeed acquired a degree of understanding of certain patterns. Nevertheless, in some instances, the model struggles to effectively capture the underlying patterns.

The Classification Report

Class	Precision	Recall	F1-Score	Support
1	0.12	0.53	0.20	74
2	0.25	0.30	0.27	204
3	0.21	0.13	0.16	323
4	0.69	0.22	0.33	571
5	0.67	0.73	0.70	1551
6	0.87	0.18	0.30	757
7	0.74	0.11	0.19	879
8	0.04	0.95	0.07	59
Accuracy			0.38	4418
Macro Avg	0.45	0.39	0.28	4418
Weighted Avg	0.65	0.38	0.41	4418

Table 6.7: Classification Report of the All-Datasets-Combined system with unigram+bigram in the Machine Learning method

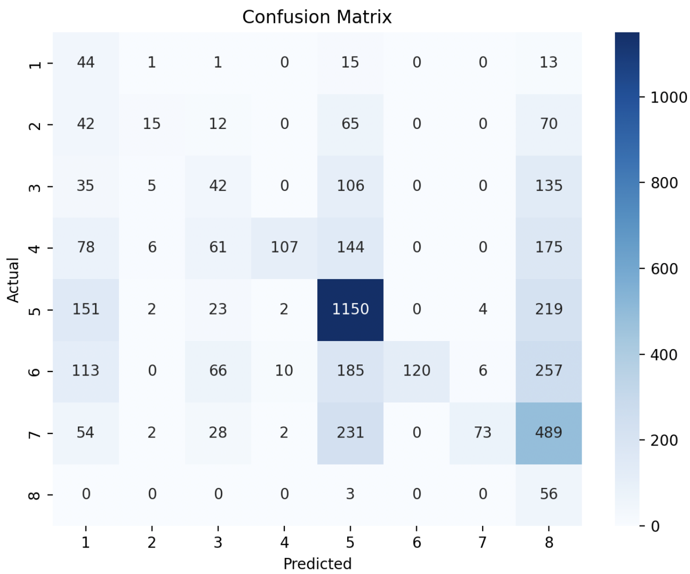

Figure 6.6: Logistic Regression on all-datasets-combined training system run on test dataset (unigram+bigrams)

Chapter 7

Discussion

Overall, the Keyword-Matching Approach outperformed the Machine Learning Method. In fact, the most effective system across all systems was found to be from the Keyword-Matching Approach, particularly when using the "all-datasets-combined" training dataset and employing unigram+bigrams system in the list size of 1000, which resulted in a weighted average F1 score of 0.59. On the other hand, while the Machine Learning System that employed the "all-datasets-combined system" yielded a substantially better F1 score of 0.41, ranking as the third best-performing system among all the options, the least effective system was identified within the Machine Learning approach as well. This particular system was trained using trigrams in the non-Dutch training dataset, resulting in a notably low F1 score of 0.01 (weighted average).
It's worth noting that the "non-Dutch training system" consistently demonstrated poor performance levels in both Machine Learning and Keyword-Matching Systems. In essence, when the non-Dutch training dataset is run on the Dutch dataset, performance significantly declines. However, when the same system is tested on the development data, which also lacks Dutch instances, it attains a more respectable F1 score of 0.5. This disparity indicates that the test data (NLQF) falls short when applied to a training dataset devoid of NLQF instances. In other words, NLQF does not exhibit strong performance when compared to other countries' EQF descriptions. As a result, it can be deduced from this observation that even though the fundamental concept of EQF descriptions remains consistent, a system lacking NLQF document data in its training cannot make accurate predictions regarding NLQF. Given the discrepancy in the matching words, along with the mismatch in vocabulary and context between the training data (comprising descriptions from multiple countries) and the test data (containing only Dutch EQF descriptions), could contribute to the lower-than-expected F1 score in the non-dutch training systems. This variation underscores the importance of dataset compatibility and consistency in classification tasks.
Considering the Error Analysis of the non-dutch training system of the Keyword-Matching Approach, one of the notable observations is the consistently low weighted scores seen across the results. These low weighted scores generally signify a limited resemblance between the words present in the test sentence and the keywords associated with each class. This indicates that the words within the test sentence do not align strongly with the prominent words that were established for each class during the training phase using the provided data. This outcome was anticipated, and it's confirmed by the low weighted scores. These scores demonstrate that the terms within the test examples from the NLQF dataset don't closely resemble the terms that were

identified as characteristic of each class in the non-Dutch training data during the training phase. This divergence in word similarity might be attributed to variations in language usage or writing style between the training and test datasets. Such differences resulted in decreased classification accuracy in the non-dutch training systems.

As for the non-dutch system in the Machine Learning Approach, an interesting finding emerges with a weighted average F1 score of 0.01. This outcome is intriguing, as it deviates from the expected notion that the machine learning approach would perform better. Notably, the imbalance in class distribution could have impacted the system's outcomes. The prominence of certain classes might have prompted the model to prioritize those more frequent classes, possibly introducing an imbalance-induced bias. The observed underfitting in this system suggests a need for dataset improvement, especially given that the machine learning system demonstrates superior results when trained on the "all-datasets-combined" system by scoring 0.41 f1-score. Given there is a bias made towards class 1, as mentioned in the Error Analysis section, it is figured that the model did not learn well about the descriptive information about the classes, and there is a bias towards class 1. Moreover, as the training dataset is imbalanced, there appears to be a generalization problem. As cited in Miller (2023), when a model is excessively trained on its training data, it loses the ability to generalize. This means that when presented with new data, the model produced incorrect predictions, rendering the model ineffective. Furthermore, some mistakes happened because the dataset had noise in some documents. There were cases where certain phrases, like "contact provider" or "information contact," were linked to particular sections in the documents, but these were often classified wrongly as class 8 in many of the systems tested in this study. Therefore, the pre-processing and data cleaning process seems to not to work well enough and this affected the systems in the Keyword-Matching-Approach.
As for the all-datasets-combined systems, the Keyword-Matching Approach, unigram+ bigrams system with the list size of 1000, shows the highest score of 0.59. As mentioned above, there are vocabulary and context differences in datasets, which caused the non-dutch training system's low performance. Considering the training and test datasets, both include instances from all datasets used in this study (Swedish, Latvian, Maltese and Dutch), the predictions were observed to be better. To elaborate, each dataset has a different context of documents, and for the same classes, their terminology may differ from one document to another. However, when this diversity is presented in both training and test datasets, the test dataset also includes that diversity of terms and phrases, which will end up having better chances of making correct predictions. Although some of the classes lacked distinct or prominent keywords typically associated with those classes, leading to their incorrect assignment to inappropriate categories, the performance of this system shows satisfactory results. However, when the unigram system with the list size of 100 is taken into consideration in the same training system, the performance slightly dropped. This is because of the abbreviations used in the dataset. As in the instance given in the Error Analysis section, test instances including "b.a" (which refers to Bachelor's Degree) are classified as "Level 1" because, in the training data, b.a is mostly written as "bachelor degree". Consequently, the system fails to recognize that "b.a" refers to the "bachelor degree." and it does not match them. The term is important for classes 7 and 8 as these levels describe undergraduate education in the documents, and the utilization of abbreviations in the documents diminished the effectiveness of the saliency list due to differences resulting from the abbreviation

usage.
Meanwhile, class 8 accurately predicts 54 instances out of 59 documents, given its keywords are representative of class 8. To elaborate, some of the keywords (salient words) for class 8 in the list are *scientific, study, academic, professional, management, business, practice, independently, theory, phd, advanced, theoretical, thesis, doctoral, ethical, develop, manage*. Considering the salient unigrams listed above, the words show that level 8 consists of words which are mostly related to creating/developing/improving and qualification descriptions for level 8 are supposed to describe independent working. In other words, in level 8 (higher education), learners are supposed to be able to work independently to create what they have learned, and they are supposed to be criticizing, evaluating, ogranizing, managing, and developing the already learned context and as those words are actually in the list show that the saliency list is accurate and the performance of the classification for class 8 being class 8 is correctly predicted.

In the Machine Learning approach with all-datasets-combined system, unigram+ bigrams showed the third-best performances among all systems in this study. Although it is expected for a model to show better results than a rule-based approach, Logistic Regression shows slightly lower performance. To clarify, Keyword-Matching-Approach with the unigram+bigram system showed a 0.59 f1-score, while the Machine Learning Approach in the same system showed a 0.41 f1-score. Thus, with a small difference, the Machine Learning Method became the third-best working system.

Overall, the unigram+bigram n-grams consistently emerged as the best-working n-grams in all the experiments in this study, as long as the dataset was all-datasets-combined. Especially in the Keyword-Matching-Approach, the unigram+bigrams showed 0.57-0.59 F1-score in all sizes of lists in the all-datasets-combined system. This enhancement in performance suggests that unigram+bigram features are effective in distinguishing subtle differences in EQF-level descriptions.

7.0.1 Future Work

The current study lays the foundation for several promising avenues of future research aimed at enhancing the performance and extending the scope of the proposed keyword matching classification system. One key aspect to explore is training the system on a more balanced dataset, encompassing both the training and testing datasets. By addressing the class imbalance within these datasets, the system's ability to accurately classify instances across various classes could be improved.
Furthermore, an intriguing direction for future investigation involves the exploration of cross-lingual classification techniques. Given the multilingual nature of the EQF descriptions in the dataset, experimenting with methods that enable the system to classify descriptions in languages beyond Dutch could potentially yield valuable insights and broaden the system's applicability. During the course of this study, the evaluation of the systems primarily centred around the NLQF dataset, while the training was conducted on a non-Dutch dataset. Additionally, the all-datasets-combined system, characterized by its imbalanced and diverse composition, was exclusively assessed using the all-datasets-combined test set. An intriguing avenue for further investigation involves carrying out experiments on the all-datasets-combined system, with a specific focus on the NLQF dataset. This approach would entail analyzing the system's performance within the context of the NLQF dataset within the broader framework of

the all-datasets-combined system. This approach holds the potential to offer deeper insights into the system's adaptability and performance characteristics in relation to this particular dataset.

In conclusion, the future work for this study should encompass efforts to enhance the system's performance through balanced dataset training, cross-lingual classification exploration, a more comprehensive analysis of the all-datasets-combined system's behaviour when applied solely to the NLQF dataset, and an in-depth investigation into the preprocessing techniques used in organizing each dataset compiled together. These directions hold the potential to further refine and expand the capabilities of the proposed keyword-matching classification system, ultimately contributing to more accurate and versatile classification outcomes.

Chapter 8

Conclusion

In this study, my primary goals were to investigate the feasibility of detecting skill-level within the European Qualifications Framework (EQF) from qualification-describing documents and to assess the applicability of using EQF level-labelled qualification descriptions from various countries to classify the Dutch EQF dataset (NLQF). To achieve these aims, I embarked on a comprehensive exploration of diverse classification approaches, with a particular emphasis on skill-level detection. Through rigorous evaluations of both rule-based and Machine Learning methods, this research has yielded valuable insights into the capabilities and limitations of each approach.

The Keyword-Matching Approach, particularly the system utilizing unigram+bigram combinations, emerged as the standout performer. This finding aligns with the observation that the salient terms in EQF descriptions play a crucial role in determining skill-level categorization. Notably, despite the imbalanced nature of the dataset and the potential variance in phrasing across countries' qualifications, the Keyword-Matching Approach exhibited strong performances in all-datasets-combined systems. Our results underscore the significance of keyword-based analysis in identifying skill-level attributes.

Additionally, we observed the impact of training data origin on classification accuracy. When training on other countries' EQF datasets and applying the models to Dutch EQF data, lower performance was evident. This points to the importance of domain-specific training data in achieving optimal classification outcomes. Furthermore, the findings illuminate the potential for a unified approach encompassing data from diverse countries' EQF descriptions to improve classification accuracy.

In conclusion, this study contributes to the field of skill-level detection by highlighting the success of the Keyword-Matching Approach, particularly with unigram+bigram combinations. We've shown the importance of considering dataset origins and the potential of adapting established algorithms for novel classification tasks. Future research should delve into refining preprocessing techniques, enhancing dataset balance, and exploring semantic analysis for further improvements in skill-level detection.

These findings underscore the significance of context-specific approaches in the realm of document classification. As qualifications continue to diversify and evolve, this study offers valuable insights for enhancing the accuracy and efficiency of skill-level determination, aiding policymakers, educators, and professionals in navigating the complex landscape of qualifications and competencies.

8.0.1 Answering the Research Question(s)

To revisit the primary research question and its associated sub-questions:
Can skill levels be identified from descriptions of qualifications within the European Qualifications Framework (EQF)?
This study demonstrates that skill-level detection can be achieved using both rule-based and machine learning methods. However, the findings indicate that the keyword-matching approach yielded superior performance compared to the machine learning approach in this context.
""Is it possible to use various countries' EQF level (qualification) descriptions to classify Dutch EQF data set (NLQF)?"
The outcomes of this study reveal that employing diverse EQF level-labelled qualification descriptions from various countries to classify the Dutch EQF dataset (NLQF) may not yield favorable results. In this study, using various countries' EQF descriptions to classify NLQF levels did not show good performances on classifying NLQF descriptions. In other words, if the classification is not introduced to any NLQF instances in the training and it has different EQF descriptions that it is trained on, it did not make quite correct inferences to classify NLQF dataset. However, it's important to acknowledge that with enhanced and balanced classification systems, the potential for success remains. Notably, this study presents initial insights into the novel task of document classification based on EQF levels. It is worth mentioning that through refining the preprocessing procedures and ensuring dataset balance, performance could be enhanced. Nevertheless, the investigation demonstrated that utilizing a training dataset devoid of NLQF instances, yet encompassing qualifications from other nations such as Sweden, Latvia, and Malta, did not yield satisfactory outcomes.

Bibliography

J. Brownlee. A gentle introduction to logistic regression with maximum likelihood estimation, 2019. URL https://machinelearningmastery.com/logistic-regression-with-maximum-likelihood-estimation/.

P. Chandravanshi. Nlp unlocked: n-grams, 2021. URL https://medium.com/@pankajchandravanshi/nlp-unlocked-n-grams-006-ceab1bc56bf4.

Directorate-General for Employment and Social Affairs and Inclusion (European Commission). Comparison report. Online, 2023. URL https://europa.eu/europass/system/files/2023-02/Comparison%20report%20final%20rev%2023-02-2023%20EN.pdf. Published: 2023.

Y. Hamdaoui. Tf-idf (term frequency-inverse document frequency) from scratch in python, 2019. URL https://towardsdatascience.com/tf-term-frequency-idf-inverse-document-frequency-from-scratch-in-python-6c2b61b78558.

V. Kanade. What is logistic regression? equation, assumptions, types, and best practices, 2022. URL https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/.

B. Koloski, S. Pollak, B. Škrlj, and M. Martinc. Extending neural keyword extraction with tf-idf tagset matching, 2021. URL https://aclanthology.org/2021.hackashop-1.4.pdf.

K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, and L. Barnes. Text classification algorithms: A survey, 2019. URL https://www.mdpi.com/2078-2489/10/4/150?utm_content=buffere0b87.

K. Leung. Micro, macro weighted averages of f1 score, clearly explained, 2022. URL https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f#:~:text=If%20you%20have%20an%20imbalanced,is%20weighted%20by%20its%20size.

M. Marcińczuk, M. Gniewkowski, T. Walkowiak, and M. Bedkowski. Text document clustering: Wordnet vs. TF-IDF vs. word embeddings. In *Proceedings of the 11th Global Wordnet Conference*, pages 207–214, University of South Africa (UNISA), Jan. 2021. Global Wordnet Association. URL https://aclanthology.org/2021.gwc-1.24.

MarketBrew. The role of tf-idf in modern search engine optimization strategies, n.d. URL https://marketbrew.ai/the-role-of-tf-idf-in-modern-search-engine-optimization-strategies.

D. Marwah and J. Beel. Term-recency for TF-IDF, BM25 and USE term weighting. In *Proceedings of the 8th International Workshop on Mining Scientific Publications*, pages 36–41, Wuhan, China, 05 Aug. 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.wosp-1.5.

M. Medved, M. Jakubíčˇek, and V. Kovárˇ. English-french document alignment based on keywords and statistical translation, 2016. URL https://aclanthology.org/W16-2374.pdf.

V. Nithyashree. What are n-grams and how to implement them in python, 2021. URL https://www.analyticsvidhya.com/blog/2021/09/what-are-n-grams-and-how-to-implement-them-in-python/.

J. Piskorski and G. Jacquet. TF-IDF character N-grams versus word embedding-based models for fine-grained event classification: A preliminary study. In *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*, pages 26–34, Marseille, France, May 2020. European Language Resources Association (ELRA). URL https://aclanthology.org/2020.aespen-1.6.

A. Simha. Understanding tf-idf, 2021. URL https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/.

M. Taddy. Document classification by inversion of distributed language representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 45–49, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2008. URL https://aclanthology.org/P15-2008.

E. Union. URL https://europa.eu/europass/en/europass-tools/european-qualifications-framework.

M. S. Zakirizvi. A comprehensive guide to language model with nlp: Python code, May 2023. URL https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-language-model-nlp-python-code/.