Addis Ababa University

College of Natural Sciences

Department of Computer Science

Amharic Question Classification System Using Deep Learning Approach

Saron Habtamu Zelelew

A Thesis Submitted to the Department of Computer Science in Partial Fulfillment for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia

April 2021

Addis Ababa University

College of Natural Sciences

Department of Computer Science

Saron Habtamu Zelelew

Advisor: Yaregal Assabie

This is to certify that the thesis prepared by Saron Habtamu, titled: *Amharic Question Classification System Using Deep Learning Approach* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

	Name	Signature	Date
Advisor:			
Examiner:			
Examiner:			

Abstract

Questions are used in different applications such as Question Answering (QA), Dialog System (DS), and Information Retrieval (IR). However, some questions might be too complex to be analyzed and processed. As a result, systems are expected to have a good feature extraction and analysis mechanism to linguistically understand these questions. The retrieval of wrong answers, inaccuracy of IR, and crowding the search space with irrelevant candidate answers are some of the challenges that are caused due to the inability to appropriately process and analyze questions. Question Classification (QC) aims to solve this issue by extracting the relevant features from the questions and by assigning them to the correct class category.

Even though QC has been studied for various languages, it was hardly studied for the Amharic language. This research studies Amharic QC focusing on designing hierarchical question taxonomy, preparing Amharic question dataset by labeling the sample questions into their respective classes, and implementing Amharic QC (AQC) model using Convolutional Neural Network (CNN) which is part of the DL approach.

The AQC uses a multilabel question taxonomy that integrates coarse and fine grain categories. This multilabel class helps us to be more accurate in retrieving answers compared to the flat taxonomy. We constructed the taxonomy by analyzing our AQ dataset and also adopting the standard taxonomies that were previously studied. We have prepared the AQs in three forms: Surface, Stemmed, and Lemmatised forms. We train and test these datasets using a word vectorizer trained on surface words noticing that most interrogative words appear to be similar even when they are stemmed and lemmatized. As a result, we have achieved 97% and 90% training and validation accuracy for Surface AQs. Scoring 40% for the stemmed AQs. However, the word2vec model could not represent the lemmatized AQs appropriately. As a result, no results were obtained during training. we also tried to extract features from AQs by using different filters separately. This gave us an accuracy of 86% while requiring an increasing number of training epochs.

Keywords: - Amharic Question Classification, Deep Learning, CNN, Fine grain, Coarse grain Hierarchical Taxonomies, Word2vec.

Dedication

To My Family

Almaz and Habtamu

Acknowledgments

I would like to give my biggest gratitude to almighty God, who has paved the way and made me reach this point of success.

I would also like to thank my advisor Dr. Yaregal Assabie for investing his precious time and knowledge without hesitating. Your encouragement and guidance have led me to pursue my academic skills as well as knowledge and also made the work to be practically successful.

Obviously, it wouldn’t be easy without my family who have helped me throughout my entire journey and beyond. Thank you for your priceless support and effort which has made me achieve this glorious opportunity that I have ever dreamed of.

Medhanit Getachew, Elham, thank you for your humbleness in sharing your thoughts and all your concerns for this work. Classmates and instructors, it has been a great time and support in filling the gaps I have faced during my Postgraduate year.

Table of Contents

List of Tables

List of Figures

List of Algorithms

List of Acronyms

ABBC	Attention-Based BiGRU-CNN network
AQC	Amharic Question Classification
CNN	Convolutional Neural Network
DL	Deep Learning
DS	Dialog System
FCL	Fully Connected Layer
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model
IE	Information Extraction
LSTM	Long Short-Term Memory
MLP	Multi-Layered Perception
NLP	Natural Language Processing
QA	Question Answering
QC	Question Classification
RNN	Artificial Recurrent Neural Network
SVM	Support Vector Machine
TREC	Text Retrieval Conference
UTQD	University of Tehran Question Dataset

Chapter 1: Introduction

1.1 Background

Natural language is used to interchange ideas and establish communication between one another in a written or spoken form. Natural Language Processing (NLP) involves Artificial Intelligence and Linguistics to make computers understand the statements or words written in human language. NLP draws from various knowledge areas: The sound of words (phonology), word formation (morphology), sentence structure (syntax), and the ontology of words (Semantics) are analyzed using NLP [1].

NLP gained much attention for representing and analyzing human language computationally and bridge the communication gap between people and computers [2]. When we consider dialogues or human-computer interaction, we have to make the system have lots of knowledge and strong inference mechanisms such as logical inference and common-sense reasoning. Question Answering (QA), Dialog System (DS), and Search Engines bridge the interaction gaps by analyzing queries and retrieving answers.

DS deals with responding to customers' questions and facilitates transactions by providing an automated response in guiding users [3]. Question answering has also been used to develop DS and chatbots to maintain a human-like conversation with machines. QA is the extraction and retrieval of candidate answers automatically.

As search engines began to incorporate search over data into their algorithms, results have evolved accordingly. The search pages include knowledge panels, answers to questions, and more [4]. To create relevant content, determining the type of question the audience is asking and how to supply the content for each type of question is very crucial.

There are several types of questions. The rhetorical question type makes a point rather than demanding an answer. For instance, What's the problem with you these days? is a rhetorical question. A question tag is a grammatical structure that has a declarative or an imperative statement that is turned into a question by adding an interrogative fragment. For example, you know Ethiopia, don't you?

The Non-interrogative grammatical structures do not use a question mark as the rest of the question types to refer to questions. This tends to be difficult for machines that try to identify questions using question marks as an identifier. For example, "tell me your name" is an imperative sentence asking the subject its name.

Confirmation (closed-ended) questions require answers in the form of a Yes or No answer type. This shows the requirement of systems to have strong inference mechanisms, a higher level of knowledge acquisition and retrieval techniques, world knowledge, and common-sense reasoning to analyze questions.

Factoid questions include WH- questions, which consist of Hypothetical [What] and Causal [how and why] question types respectively [5]. Open-ended questions include advantage or disadvantage, cause and effect, comparison (evaluative), example, explanation, identification, opinion, rationale, and significance. The non-factoid question consists of a list, definition, acronym, and how type of questions. There are also question types like cross-lingual, counterpart, famous, stand for, synonym, name-a, name-of, and true or false.

Understanding questions by figuring out their complexities is one of the main tasks in question answering. This requires analyzing questions, which incorporates Question Classification, Query Generation, and Query Expansion. Question classification is a way of assigning questions to the appropriate class category after understanding their semantic meaning. Question classification plays an important role in applying detecting rules, categorization, and identifying the critical elements of a question [6].

The class category that represents an answer type is known as a taxonomy. QC uses these taxonomies to assign questions to a category that best fits the answer type of a query. Taxonomy can be flat or hierarchical (multi-label). A flat-type question taxonomy consists of only the coarse grain class. Flat question taxonomies also assign the question to only one class category. The multi-label classes are used to classify questions by assigning more than one class label. More than a two-label class is acquired by involving both coarse grain and fine grain categories. The coarse grain categories are more generic that consist of fine grain categories in them. For instance, Location. Fine-grain categories are more specific than the coarse grain category. They are the subfields of the main category like State, Country, City, and Provinces for the generic coarse grain class “Location” taken as an example. A multilabel or hierarchical classification is important to get a more specific and accurate retrieval system than the flat taxonomy.

Question Classification (QC) has been implemented using different approaches. Rule-based and Machine learning are among the well-known approaches [7]. The authors explained the approaches to question classification in three main groups: rule-based, machine learning, and language modeling-based. In the machine learning approach, expert knowledge is replaced by a sufficiently large dataset that gives information after being labeled or tagged to retrieve data easily.

Deep learning (DL) is a dominant Artificial intelligence (AI) approach, especially for classification purposes. DL can learn from a large set of data in a supervised manner. DL outperforms other models by analyzing and extracting features using learnable weights that act as a human brain to easily understand the input data.

1.2 Motivation

Amharic is the working language of the Federal Democratic Republic of Ethiopia. It is one of the most spoken Semitic languages next to Arabic [9]. The language is also used in several states or regions within the federal system, including Amhara and the multi-ethnic Southern Nations, Nationalities, and Peoples Region [10]. As the number of speakers grows, the number of electronic transactions and data usage also increases. Hence, it appears to be an increasing amount of demand for an automated system to exploit these data appropriately.

To exploit data especially in information retrieval, it is necessary to understand questions thoroughly. Amharic has a wide range of interrogative words as listed in Table 6.1. This language is not limited to using interrogative words to form questions. It has a distinct way of grammatical construction, character (Fidel) representation, and statement formation [10]. If we see the word "ሄደ?" it is written as "did he go?" in English. Who is/are the world champion? can be translated as "የዓለም ሻምፒዮን ማን ነው?", "የዓለም ሻምፒዮን ማነው?" or "የዓለም ሻምፒዮን ማናቸው?", "የዓለም ሻምፒዮን ማን ናቸው?" as well as "የዓለም ሻምፒዮን ማነኙ? ", this shows that Amharic uses different word formation, grammatical arrangement, and type of question particles to ask a question having a similar semantic meaning. As a result, it is inspiring to work more on this area to solve issues that happen in the question answering and retrieval systems.

Researchers [9, 17, 18] have worked on QC for Amharic questions under different applications like QA. However, the studies are not generic enough to analyze terms other than the interrogative words that are used to form questions. Most researches on Amharic QC are also made in a closed domain, or by just considering factoid or non-factoid question types using rule-based or machine learning approaches like SVM.

However, these approaches can be difficult to work on a large amount of data and to effectively extract features. Thus, we intend to work on a generic question classification using a large set of AQs using the DL approach. This approach easily extracts features by processing a large amount of data.

1.3 Statement of the Problem

The emergence of Artificial Intelligence and Information Technology has contributed to the development of question analysis as well as QC. Researchers [11], have studied highly informative phrases or words within a question, explored question taxonomies according to the semantic features, and tried to solve the lack of information that caused misclassifying instances of the minority class by proposing a multi-label QC.

QC is language-dependent. Each language uses its unique letter, words, grammatical structure, question formation, and pattern. Researchers have developed QC for various languages. Mohammad Razzaghnoori *et al*. [7] developed Persian QC. Yihe Yang et al. [12] developed Chinese QC using deep learning. Alami Hamza *et al.* [13] developed Arabic question classification using SVM. Syed Mehedi *et al*. [14] developed a question classification for Bangladesh using SVM with a hybrid feature extraction method. [15, 16] developed English QC using a deep learning approach.

However, The question construction and answering techniques in Amharic are different from English and other languages. the QC developed for other languages cannot be applied to Amharic.

Various authors [6, 9] have attempted to work on Amharic Question Classification under QA and Dialog system. However, the study of Amharic question classification as a part of QA or DS has limited the researchers' focus to be on a restricted domain, dataset, and to consider only predetermined question types. If we see some of the works on QC under QA, Definition, Description and Biographical question types were studied by Tilahun Abdisa [6], factoid questions by using Ethiopian History dataset (closed domain) was studied by Medhanit Getachew [9] and the study of List type questions for Ethiopian tourism was made by Brook Eshetu [15].

The Question Classifications researched under QA for Amharic are either Pattern Based rule-based or using machine learning approaches such as SVM [16]. The more data there is, the more complex it is for the machines to train. In the meantime, DL has provided better performance for classification problems. The DL algorithm can capture complex phenomena: If fed the right amount of data, which tends to be a lot of data, deep learning algorithms can learn the most complex patterns in the training data. Thus, we propose a deep learning approach for the Amharic Question Classification (AQC model).

1.4 Objectives

General Objective

The general objective of this research is to design a generic Amharic Question Classification model using a Deep Learning approach.

Specific Objectives

The specific objectives of this research are

- Review literature and related work on grammatical structures of Amharic questions, QC, and the methods of classifying questions.
- Collect Amharic question dataset.
- Identify and redefine Amharic question taxonomies at a hierarchical level.
- Design a generic AQC model using a CNN a DL approach.
- AQC system prototype development.
- Test the performance of the system.
- Evaluate the model using AQs in a Surface, Stemmed, and Lemmatized form.

1.5 Methods

To conduct this research, we will review the nature of Questions specifically the grammatical structure of Amharic questions, the interrogative words, and the taxonomies we intend to use for categorizing (labeling) as well as classifying questions. The strategies and approaches used for vectorizing data, extracting features, and classifying questions will also be assessed to analyze the gaps and improvements in QC to identify problems.

After we identify problems, we will design and develop an AQC prototype model that will address the gaps. This process initially involves collecting a large dataset of Amharic questions from Amharic electronic documents, websites (forums and news sites), Ethiopian National Exams, and QA communication media like Tameson and Bahrdar FM to get an archived generic Amharic question dataset. We will also reach out to researchers that have previously collected questions for their work such as Tilahun Abdisa [6] and Medhanit Getachew's [9].

Since we will study the Amharic QC using Surface, Stemmed, and Lemmatized datasets, we will take the surface questions as they are and preprocess 2000 sample questions into a stemmed and lemmatized form by using a semi-automated system developed by [17]. After having these datasets (Surface, Stemmed, and Lemmatized AQs), we normalize, remove special characters, and tokenize them to feed the structured data into the AQC model.

The proposed DL approach, i.e., CNN will be applied to extract features, train the model and classify questions after representing the dataset into a vectorized form. We will use python and Keras functional API to develop the AQC prototype. The model will be evaluated using the Confusion Matrix system that evaluates a model that has a hierarchical classification system. The implementation and evaluation will be discussed more in Chapter 5.

1.6 Scope and Limitations

The AQC model focuses on classifying Amharic questions since we are more familiar with the language. We study a generic type of questions that are not restricted to a specific domain. As a result, we define taxonomies on a high level as 'Number: date' which considers just the date and not just the specific domains like History or a person's Biography (birthdate). This indicates that the model mainly considers the taxonomies we specified and the datasets it is trained on. Hence, the AQC will examine any domain related to the hierarchical question taxonomies i.e., the coarse grain and fine grain classes defined in Figure 4.2.

We use the word2vec model that is trained on Amharic surface words. Using word2vec incorporates a wide range of semantically related words compared to other alternatives where it is expected to encode every character with 0s and 1s (One-hot encoding.). Hence, we analyze questions that are represented in a vectorized form by a word2vec model represented sparsely.

1.7 Application of Results

The Amharic Question classification system can be a great help in areas that demand processing inquiries and user requests. QA, IE, Search engine, and DS can leverage this system. Dialog systems can process questions that come from users and create an interaction with the computer system by identifying what the user needs. Search Engines can focus on the core request once it gets the question class. An automated help desk provides online help in the absence of an expert. It is similar to Customer service, chatbots, and other bidirectional communications among the user and the machines.

1.8 Organization of the Rest of the Thesis

The rest of this thesis is organized as follows. Chapter 2 presents the literature we reviewed on the fields related to QC. Chapter 3 reviews the researches done on QC for different languages. Chapter 4 explains the proposed AQC model and its components. The evaluation and implementation of the AQC prototype are included in Chapter 5. We summarize and present the future work in Chapter 6.

Chapter 2: Literature Review

This Chapter focuses on explaining QC, the taxonomies used for classifying questions, and the general architecture of QC. The classification of questions alongside the tools and strategies used to accomplish each task will also be covered. Lastly, some of the DL approaches used to classify questions such as CNN, LSTM, and RNN will be discussed as follows.

2.1 Question Classification

We employ questions to find information and fill the gaps that result from a lack of knowledge. Automated systems also utilize questions to retrieve, extract information, and provide services by being integrated into a Human-Computer interaction system such as DS. These systems process questions to understand their semantic meaning by using question analysis. Question analysis consists of QC, query generation, and query expansion [6].

QC is a process by which a system analyses a question by understanding the syntactical and contextual basis. It is also known as Question Categorization because they categorize questions based on the labeled categories to which they belong [18]. Classifying questions is the task of assigning a label to an input question [7]. This is especially essential in automatic question answering, document retrieval, and information extraction systems by learning from several labeled or tagged question samples.

IR (retrieving documents) and IE (retrieving the contents in a document) integrate QA to retrieve or extract the relevant answer to the user. Information can be retrieved by taking questions and a sentence/paragraph which has the expected answer(s) in it. IR tackles the problem of document retrieval based on the closeness of the document and the questions submitted to the IR system. IE on the other hand involves deep analysis of queries (i.e., user questions) to understand the user's intention for precisely indicating to extract correct answers (sentences or passages) [10]. Hence, The advancement of this and E-Learning calls for the need for QC that improves QA by reducing the search space of potential answers by discovering answers more efficiently and accurately [8]. While research into QC is fairly mature, a multilabel (hierarchical) QC is not always used for answering questions [19]. Answer selection and QC are interdependent things that demand a higher accuracy and flexibility in understanding the questions considering two or more candidate classes by creating a relationship between the taxonomies, semantic classes, and the questions [18].

2.2 Question Taxonomy

Question Classes are a strong signal to Deep Learning models. The specific system of classes used by a QC is known as a Taxonomy [19]. Question taxonomies can be applied in public services as well as e-government agencies that have several users. These users can be managed using user-friendly interactive systems like DS, and Help Desk, as well as QA that are expected to display the relevant response to the inquiries by categorizing questions into the appropriate taxonomies such as Agriculture, Product, Health, Language, Location, and Event after analyzing the question according to the trained QC model. This helps the system retrieve accurate data for users.

Question taxonomies can be divided into Flat and Hierarchical. Flat taxonomies have only one level of category. On the other hand, Hierarchical taxonomies consist of super- and sub-categories [20]. Questions are classified into one or more classes that incorporate the answer type. QC uses these classes to categorize queries after analyzing them. A set of question categories of possible answer types is called a taxonomy [20]. Answer type taxonomies can be divided into flat and hierarchical. Flat taxonomies have only one level of categories and are therefore not taxonomies in the true sense of the word. Hierarchical or multi-label taxonomies consist of super- and sub-categories. The hierarchy contains 6 coarse(super- or main) classes (Abbreviation, Entity, Description, Human, Location, and Numeric Value) and 50 fine classes [21].

One difficulty QC task is, there is no a completely clear boundary between classes which creates an ambiguity to classify questions. Question taxonomies have a semantically related word list. Features will be extracted for the specific class if a word in a question belongs to the list. For example, What do goats eat? Can belong to Food considering the word eat, Plant (taking the words goat and eat), or Animal by analyzing the word goat [21].

Search engines receive keywords and return some relevant and irrelevant data to the users. While QA systems were designed to get a natural language question and retrieve more probable and appropriate answers. An appropriate answer has the characteristics of being concise, comprehensible, and correct [7]. A common solution to constructing answer type or taxonomies is to manually extract a subset of WordNet [20]. Another approach is to manually analyze a specific corpus, i.e., a collection of texts or questions, and infer a taxonomy from it [21].

2.3 The General Architecture of Question Classification

Question classification consists of Question Pre-processing, Feature extraction (Question Processing), and Categorization as pictorially presented in Figure 2.1 Question preprocessing is the initial step after a dataset collection. It is where the data can be normalized, stemmed, tokenized to make the machine understand its inputs after they are put in a vectorized form. A question analyzer is a sub-component of question processing that processes and analyzes the question type to produce a set of keywords to help the system predict the expected answer type [9]. Figure 2.1 shows the key difference between the previous and the current knowledge-based designs and how the whole process is simplified in the Machine Learning approach.

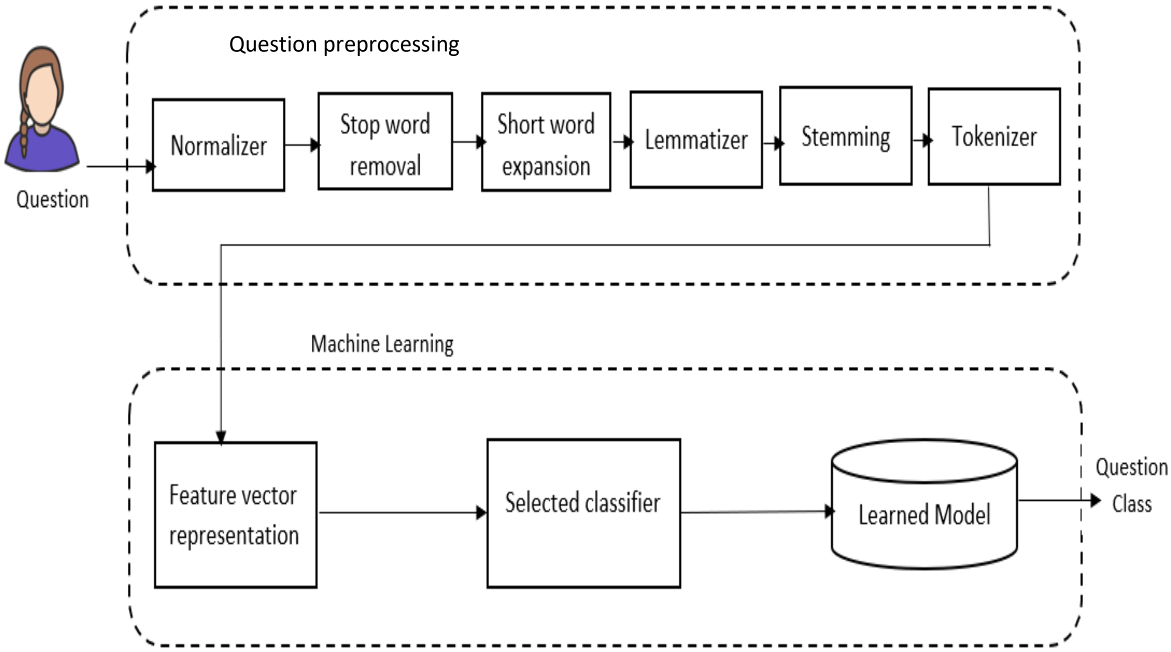

Figure 2.1: The General Architecture of QC

2.3.1 Question Preprocessing

Question preprocessing is the first step after collecting data. It is the way of preparing the data into a more analyzable and comprehensive way of presenting it to the machine learning process. Normalization, Stopword removal, Short word expansion, Lemmatization, Tokenization, and Stemming are some of the preprocessing steps in NLP.

Normalization

Amharic uses a writing system called Fidel or Abugida adapted from the Ge'ez language [10]. Amharic Fidels have more than 380 Unicode representations (U+1200-U+137F) which is an Ethiopic Unicode Standard). In Amharic, one word can be expressed using different characters that have the same sound but different structural appearance like ሀ/ሐ/ኀ, ጸ/ፀ, ሠ/ሰ, አ/ዐ,… ሆ/ሖ/ኆ, ጸ/ፆ, ሦ/ሶ, and ኦ/ዖ. A word could have different forms due to the addition of prefixes, infixes, and postfixes, these affixes could change the role of the word to be noun, verb, adjective, adverb [22].

Stop Word Removal

Most of the time, some of the stop words in a question or question particles are more important in determining the correct content that might contain the answer. For example, in the question "ኢትዮጵያ ውስጥ ስንት የመንግስት ዩኒቨርሲቲዎች አሉ?" (how many public universities are available in Ethiopia?)" systems might consider the three words (ኢትዮጵያ (Ethiopia), መንግስት (Government) and ዩኒቨርሲቲ (University)) that might help in detecting the correct document where ስንት ነው (how many) is a very important word removed as stopwords. For the question "የአፍሪቃ ዋንጫ መቸ ይጀመራል? (When will African Football cup start?)", 2000-2002, መስከረም-ጳጉሜ (September-August) that can be inferred from the interrogative word መቸ (when) [10, 24]. So, these terms are very practical in IR and other applications that involve questions for searching their answer type.

Short Word Expansion

Short words in Amharic usually use "." and "/" while writing words in short form. "ሕ/ሰብ", "ዓ.ም" represent "ሕብረተሰብ" and "ዓመተ ምህረት" respectively. The short word expansion is needed because the word could appear in a short form within the question and fully written in the answer source document or vice versa. So the short word expansion solves this problem by expanding the words using a set of predefined short words [6].

Stop Word Removal

Documents and questions could contain stop words like "ነው", "ናቸው", "እና", etc. [6]. They are the most frequent words that appear in many sentences. The terminology for these frequent words is known as stop words. These are considered to be impractical to distinguish one sentence from another. On the other hand, these words are treated as being very useful in classifying questions.

Stemming and Lemmatization

Stemming is an NLP technique that is used with the assumption that morphological variants represent similar meanings. It maps several words into one base form that can be used as a term. This means that it increases similarities between documents and queries because they have additional common terms after stemming. It is language-dependent and should be tailored for each language since languages have a varying degree of differences in their morphological properties [17].

In English, the number of possible inflected and derived forms is relatively small, and variation is mostly restricted to the attachment of suffixes. Compounds that are not yet lexicalized are in most cases written as separate words. Stemming, i.e., the removal of suffixes using a list of possible suffixes, is therefore considered a practical way to map related word forms to a common stem [23]. A simple stemming procedure gives the same stem for various forms of the same word while lemmatization keeps them separate [24].

Tokenization

The question processing begins by performing tokenization to extract individual terms. This is done by using different demarcations such as white spaces to chop up the input text which will be ready for the next process. For example, Where do you live? Can be tokenized as ‘where’ ‘do’ ‘you’ and ‘live’ by taking the words separately. This process will help the retrieval process to get keywords easily [9].

2.3.2 Question Processing

Question processing most often consists of the construction of question representation, derivation of expected answer type, and keyword extraction [20]. Question processing helps the machine to easily represent words numerically, comprehend questions by extracting features, and make accurate categorizations. We can extract features using lexical, syntactical, and semantic linguistic information after representing the inputs into a vectorized form.

Feature Vector Representation

Numerical representations are a prerequisite for most machine learning models or algorithms that learn to approximate functions that map inputs to outputs. This is alternatively referred to as “word embeddings”, “encoding”, and “vectorizing”. convert symbolic representations (i.e., words, emojis, categorical items, dates, times, other features) into meaningful numbers or real numbers that capture underlying semantic relations between the symbols [25].

Pretrained word embeddings are word representations that capture the syntactic-semantic meanings by getting trained on a dataset [25]. Syntactic analysis analyses how words are grouped into phrases, what words modify other words, and what words are of central importance to the sentence. The semantic analysis involves the extraction of context-independent of a sentence's meaning, including the semantic roles of entities mentioned in the sentence, and quantification information.

There are different types of trained word embeddings. Pretrained word embeddings are models that are trained on a large dataset. Pretrained word embeddings capture the semantic and syntactic meaning of a word. SpaCy, FastText, GloVe, Flair**,** Char2Vec, BERT, are some of the pre-trained word embeddings [25].

Word2vec is a language modeling that uses a neural network to learn associations in a large corpus of data such as a text. Once the word modeling is trained, the model can detect semantically related words or suggest additional words for a partial sentence. As the name implies, word2vec represents each distinct word with the corresponding list of numbers called vectors. The vectors use a simple mathematical function or a cosine similarity between the vectors to indicate the level of semantic similarity between the words represented by those mathematical functions or vectors [26].

FastText is a word2vec model that has a pre-trained resource for more than 154 worldwide languages including Amharic trained on Common Crawl and Wikipedia using fastText. These models were trained using a Continuous Bag of words with position-weights, in dimension 300, with character n-grams of length 5, a window of size 5, and 10 negatives [27]. CBOW (Continuous Bag of words) is a model that tends to predict the probability of a target word given a context. A context may be a single word or a group of words [28].

The GloVe represents words in a vector form by learning features in an unsupervised way. It trains the system on word-to-word co-occurrence statistics in a corpus. The resulting co-occurrence statistics representations showcase linear substructures of the word vector space. The word representation also takes the input and compresses it into a low-dimensional vector. It has two parts: an encoder and a decoder. The encoder is responsible for producing the low-dimensional embedding that helps the machine to understand a raw text. The decoder inverts the vectorized representation that is the computation of the network and reconstructs the original input. The decoder is different from a feed-forward network (FFN) because the mentioned network maps the embedding to an arbitrary label.

Feature Extraction

What is a feature? In machine learning, features are individual attributes extracted from an input. They are used by feature extractors to get descriptive attributes. Models use such features to make predictions.

In machine learning, new features can be obtained from old features [29]. Based on linguistic information, these features can be grouped into three different: lexical, syntactical, and semantic features. The Lexical features are extracted by using n-gram combinations such as Unigram, Bigram, Trigram, Quadgram, and Pent-gram. These n-grams extract n features from their inputs. For example, you are the assistant, aren't you?. Can be extracted by combining the combinations of the first and second word i.e. "aren't you" if the system considers a Bigram.

The syntactical feature is extracted by analyzing the grammatical structure of the sentence. There are two kinds of syntactic features used for question classification. These are tagged unigrams and headwords [29]. Tagging is labeling words in the questions grammatically like Noun, Verb, and Subject. Headword is taking keywords in data that can be used as an indicative word for classifying questions. For example, Labeling the data as LOCATION: City Where do you live? The system can learn by identifying headwords like "live" can and train to indicate Location.

Semantic features are useful in the case of sparse data like Word2vec [29]. These semantic features are extracted after representing similar words with hypernyms to get a complete meaning out of the extracted features at a different level. Sparse data representation or vectorization is a convenient way to easily understand a wide range of different words with similar contextual meaning with less computation for extracting features.

2.3.3 Question Classification

Question Classification is categorizing questions into one or more classes. The classes are determined based on the taxonomies defined for the datasets the system trains on. The performance of the QC mainly relies on the taxonomy of the answer types the questions are to be classified, the correctness of labels for a corpus of questions, and an algorithm that the classifier learns to make the actual predictions [20].

2.4 Deep Learning and Artificial Neural Networks

Artificial Intelligence (AI) is a general field that encompasses machine learning and deep learning. A machine-learning is trained rather than explicitly programmed [30]. Deep Learning (DL) is part of machine learning. The central problem in machine learning and DL is to meaningfully learn useful representations of the input data and transform the data into representations that get us closer to the expected output.

DL uses the successive layers to learn representations from data and extract meaningful representations [30]. Deep Learning is also known as layered or hierarchical representation learning which is an evolution of machine learning that creates increasingly complex hierarchical models intended to mimic thought processes in the human brain more than simple machine learning models [39].

Modern deep learning unlike other machine learning often involves a number of its successive layers to automatically learn representations from the exposure of sample datasets. These layered representations are learned through models known as neural networks [30]. When a set of processing units are assembled in a closely interconnected network, they offer a surprisingly rich structure exhibiting some features of the biological neural network. Such a structure is called an artificial neural network (ANN) [34].

The learning in DL happens by storing the specification of what a layer does to its input data in the layer's weights, which in essence are numerical representations of the input data. Learning in DL refers to finding a set of values for the weights of all layers in a network that will correctly map example inputs to their associated targets [30]. The Deep Neural Networks do input-to-target representation mapping through a deep sequence of simple data transformations or layers and that these data transformations are learned by exposure to examples.

To maintain the output of a neural network, we need to be able to determine how far our predicted output is from the expected target value. Determining the gap of the actual and predicted value is the job of the loss function of the network, also known as the objective function. This function computes captures how well the network has done by getting a distance score of predictions and the true target of the sample data.

DL back propagates to minimize the achieved loss using the score as a feedback signal to adjust the value of the weights in a direction that will lower the loss score. This adjustment is the job of the optimizer, which implements the central algorithm in deep learning what is called the Backpropagation algorithm [30].

The learning units in the neural network are called neurons. Each neuron receives several inputs, takes a weighted sum, passes it through an activation function, and responds with an output [31]. DL also uses these neurons to extract features by giving weight values to the input data. These weighted inputs are, summed together to produce the logit or a function that represents probability values from 0 to 1, and negative infinity to infinity [32]. In many cases, the logit also includes a bias, which is a constant value used to adjust the output along with the weighted sum of the inputs to the neuron. The logit is then passed through a function to produce the output. This output can also be transmitted to other neurons [33].

One of the main advantages of the algorithms implemented in NN is, they show good performance due to their multidimensional nature and much communication among elements on the classification of data with a lot of features [34]. Even though their main drawbacks include a high cost, CPU utilization, and high physical memory. Regardless of these drawbacks, Neural networks are used in a classification system. They determine the error of the network and then adjust the network to minimize the faults [35]. By gathering knowledge from experience, they avoid the need for human operators to formally specify all of the knowledge that the computer needs. AI Deep Learning has many layers with the learned features built on top of each other [35].

DL uses weights to learn features. These weights are initially assigned a random value to implement a series of random transformations. The output at the beginning is mostly far from what it should ideally be, and the loss score is accordingly very high. However, the network progressively adjusts its value with the training sample it acquires. As the weights are adjusted in the correct direction as the loss score decreases. A trained network is a network with a minimal loss for which the outputs are as close as they can be to the targets [30].

Training a DL neural network revolves around the following objects: Layers, which are combined into a network or model, input data and corresponding targets, the loss function, which defines the feedback signal used for learning, and the optimizer, which determines how learning proceeds.

2.4.1 Layers

A layer is the highest-level building block in DL. A layer usually receives a weighted input and transforms them with a set of non-linear functions. The transformed non-linear functions are then passed as an output to the next layer. A layer usually contains an activation function, pooling, and a convolution. All the layers that lie in between the first layer of the neuron's input layer and the last layer of neurons or the output layer in a network are called hidden layers [36].

Layers can be used for different purposes –fully connected or dense layers often process simple vector data, stored in two-dimensional tensors as (samples, features) [30]. It also is a data-processing module that takes one or more tensors as an input and outputs one or more tensors [30]. A tensor is a container for usually numeric data or it is a multidimensional vector input matrix. Various layers are appropriate for different data processing as well as tensor formats.

By clipping compatible layers together, we can build DL models to form useful data-transformation pipelines. More often, hidden layers learn compressed representations of the original input with fewer neurons than the input layer. In addition to this, hidden layers make the network learn complex relationships by employing the neurons to learn nonlinearity [33]. Nonlinearity is a neural network that can successfully approximate functions that do not follow linearity. Nonlinearity can successfully let us predict the class of a function divided by a nonlinear decision boundary [40].

2.4.2 Activation Functions

These functions are designed to highlight important data values from their input [37]. It is a function that is used to get the output of the node or neuron. It is also known as Transfer Function. Activation Functions can be divided into two: Linear Activation Function and Non-linear Activation Functions [38]. However, three major types of neurons are used in practice that introduces nonlinearities in their computations. These are Sigmoid, Tanh and ReLu.

Sigmoid

This is especially used for models where we have to predict the probability only between the range of 0 and 1 [33]. The logistic sigmoid function can cause a neural network to get stuck at the training time. Therefore, it is mostly used in the output layer or incase of binary classification [39].

Hyperbolic Tangent Function (Tanh)

Tanh is a shifted version from the sigmoid function where its range is between -1 and 1 [33]. This function makes the training for the next layer easier and faster by making the data more centered to have a zero mean by utilizing the activations that come out of the hidden layers.

One of the downsides of both Sigmoid and Tanh is if our weighted sum input is either very large or very small, then the gradient also called derivative or slope of this function becomes very small and ends up being very close to zero which slows down gradient descent [40].

ReLu

ReLu is faster to compute than the rest of the activation functions. As a result, it is usually the default choice of activation functions. The gradient descent in ReLu does not get stuck. One disadvantage of this activation function is that the derivative is equal to zero when the weighted sum input is negative. The problem is known as the dying Relu. If the weights in the network always lead to negative inputs into a Relu neuron, that neuron would not effectively contribute to the network training [40].

Softmax

The softmax layer in DL is used for a multi-class classification purpose. It uses a probability distribution by making the sum of all the outputs to be close to or equal to one. A strong prediction is a single entry with a weighed vector close to one. While the remaining entries close to 0 are seen as weak prediction would have multiple possible labels [40].

This layer gives a better idea of how confident we are in our predictions. Unlike in other kinds of layers, the output of a neuron in a softmax layer depends on the outputs of all the other neurons in its layer.

2.4.3 Loss functions and Optimizers

To tackle the problem of training multilayer neural networks instead of just single neurons, backpropagation was pioneered by [41]. Backpropagation uses gradient descent for error reduction, by adjusting the weights based on the partial derivative of the error concerning each weight [42]. The quantity that will be minimized during training is known as Loss Function (objective function). It represents a measure of success for the task at hand.

Optimizers improve the learning process by determining how the network gets updated based on the loss function. AdaGrad, RMSProp, and Adam are the three most popular adaptive learning rate algorithms.

AdaGrad

AdaGrad optimizer adjusts the learning rate for each parameter. While AdaGrad works well for simple convex functions, it is not designed to navigate the complex error surfaces of deep networks [30].

RMSProp

RMSprop is a gradient-based optimization technique used in training neural networks. This optimizer uses a window of fixed size over the gradients computed at each step rather than use the full set of gradients [30].

Adam

Adam optimizer is known for its corrective measures and its ability to combine zero initialization bias which is a weakness of RMSProp as well as the core concepts behind RMSProp with momentum more effectively [30].

2.5 Deep Learning Approaches in Question Classification

QC uses well-known DL approaches to easily acquire knowledge through supervised, unsupervised, and semi-supervised approaches. QC models use these approaches as necessary to train their system. More about DL for classifying questions will be explained below.

In Supervised Machine Learning, the program is “trained” on a pre-defined set of “training samples”, Supervised learning is fairly common in classification problems. It provides discriminative power by identifying patterns after being trained on the sample data. The supervised approach always learns features from a labeled dataset either in a direct or indirect form [43]. This approach like any other machine learning has a training and testing phase. In the training phase, the machine learns from a labeled sample data by adjusting the learning algorithm until it reaches the goal to predict and classify the target data without the knowledge of its label. Such a performance threshold is measured using a function called objective function or loss function. In the testing phase, the learning algorithm prediction accuracy is tested using unlabeled data. The loss function helps to minimize the prediction error [44].

Unsupervised Learning intends to learn features capturing the correlation of the given data for pattern analysis when there is no information about target class labels is available [43]. It is a much harder mechanism because the computer learns to accomplish tasks that we tell it to do. It has two approaches. The first approach is to teach the machine using some sort of reward system to indicate its correctness. However, this type of training usually faces a decision problem because it only aims to make decisions that maximize rewards rather than coming up with an accurate classification system. Clustering is the second type of unsupervised learning that intends to find similarities in the training data [35].

The semi-supervised approach has the properties of both supervised and unsupervised learning approaches [45]. The training dataset used in semi-supervised learning mostly contains a small amount of labeled data and a large amount of unlabeled data [46]. RNN, Feed-forward network, LSTM, and CNN are some of the DL models that are used in question classification by involving supervised, unsupervised, or semi-supervised learning approaches.

Feed-forward Network

The Feed-forward network (FFNets) has achieved high accuracy on many classification benchmarks [47]. The neural network in FFNets perceives the pattern of each category from their input. The input passes through each layer until it reaches the output layer. The classification happens as the model identifies the output category with a maximum value. Models based on the feed-forward network view a text as a bag of words. They are among the simplest deep learning models for text representation. But the main drawbacks include huge computational power for training, exploding gradient, no backpropagation to correct errors, and vanishing gradient problem as the number of layers increase [48].

RNN-Based Models

A recurrent neural network (RNN) is derived from FFNets. RNN based models view questions as a sequence of words and intend to capture word dependencies and text structures for classification purposes. Even though RNN models are used in NLP, to identify the sequential characteristics of input data in addition to predicting the next most likely scenario, they do not work well and often underperform because they suffer from recalling the extracted features of the sequence of words.

Among different kinds of RNNs, Long Short-Term Memory (LSTM) is the most popular architecture, designed to capture long-term dependencies like sentences [47]. LSTM has a memory cell that enables adaptive memory usage offering a way to capture relations among words without missing information. LSTM was proposed to overcome the recall issue in RNN to apply them for longer texts. This model achieves promising results on language modeling, sentiment analysis, and Natural Language Inference (NLI) [49].

Convolutional Neural Networks

Convolutional Neural Networks (CNN) is the major building block used in DL that process complex forms of data such as multiple arrays represented by a word vector. Convolutional neural networks can automatically learn a large number of filters in parallel [50].

CNN has three layers to extract features and learn from the input data. These are the Convolutional layer, Max pooling layer, and Fully connected (dense) layer. Convolution in a convolutional layer apply a filter to an input word matrix and result in a map of activations called a feature map. The feature map contains the locations and strength of a detected feature in an input.

Fundamentally, the convolutional layer is inspired by the dense (fully connected) layer. While the fully connected layer takes a one-dimensional vector as an input and outputs a new vector using matrix multiplication. The convolutional layer receives inputs up to three channels [51]. The basic computation in CNN is to convolve a window function applied to the input word matrix. Each index in the output is obtained by performing the values of the kernel element-wise by combining n-grams or two to five words at a time. with the corresponding values in the input where the kernel is sliding on, then summing up the extracted features. The full convolution is done by repeating the operation by sliding the kernel over the input [51].

If we compare RNN with CNN, RNNs learn to recognize patterns across time to analyze words sequentially. It works well for NLP tasks such as sentence analysis where the comprehension of long-range semantics is required. Whereas, CNN trains to recognize patterns in a sparse distribution across space [52]. They perform precisely where detecting local and position-invariant patterns is important. These patterns might be phrases that express a particular sentiment as “not pleasant” or topics like “Falling kingdom”.

CNN is also starting to be a more popular model in classifying questions [47]. The convolutional nets are made up of neurons, weights, and biases. The weights are shared among neurons. The sharing of weights ends up reducing the overall number of trainable weights [53]. Here, each neuron receives a certain vectorized data, takes a weighted sum, and passes it through an activation function. Each neuron has an output after analyzing the vectorized data with the weighted sum. The activation function ReLu exists between the convolution and the pooling [54].

The pooling layer operates on each feature map independently [54]. This layer performs a classification task based on the features extracted by the previous layer using filters. The pooling layer provides a fixed size output matrix by applying a filter size of its own to reduce features and extract the most valuable information. This allows us to have a variable size input and kernels, but to always obtain the same output dimensions to feed into our model [55].

The pooling operation is specified, rather than learned. There are two common pooling operations: Average Pooling and Maximum Pooling. We might also use Global Pooling to extract features. However, it is not as common as the average and max pooling operations. The Maximum Pooling (max pooling) computes the maximum value of the feature map. Whereas the average pooling considers extracting features by considering all of the elements in the feature map.

The pooling layer identifies features that change position during feature extraction, by using a local translation invariance functionality to the system [56]. Invariance to translation means the positional change in elements while we extract features. In all cases, pooling helps to make the representation approximately invariant to small translations of the input [56].

After several convolutional and pooling operations, the feature map size is reduced and more complex features are extracted. Eventually, with a small extracted feature map, the contents are reduced into a one-dimensional vector and fed into a fully-connected for some additional processing [49].

Fully connected neural networks (FCNNs) are a dense layer where all the nodes, or neurons, in one layer, are connected to the neurons in the next layer [57]. FCNN is also highly trainable by combining weighted parameters with the connected input neurons [58]. The convolutional and pooling layers use the ReLu activation function to highlight features with valuable information. Fully connected (FC) layers usually leverage from a softmax layer to classify inputs appropriately.

The softmax layer classifies data by transferring the vectorized neurons into a probability from 0 to 1 [40]. The last layer of the CNN is also seen as a loss function layer that is used to specify how the network training penalizes the deviation between the predicted and true labels [49].

CNN is different from the feed-forward neural network because CNN convolves over the input layer to compute the output using nonlinear activation functions like ReLU or Tanh applied to the features. In a traditional feedforward neural network, we connect each input neuron to each output neuron in the next layer. That is also called a fully connected layer, or affine layer [55].

Chapter 3: Related Work

The study of QC took place before the resurgence of Machine Learning. For instance. Hermjakob in 2001 used an extensive QC system consisting of 115 elementary question classes in their work [19]. This Chapter will go into detail about how QC progressed over time for different languages with their taxonomies.

3.1 Question Taxonomies and Analysis

The basic foundation in QC was laid by [21]. The authors designed a generic question taxonomy using the TREC dataset, which is domain-independent and a widely used dataset. The question hierarchy designed using this dataset contains 6 coarse classes (Abbreviation, Entity, Description, Human, Location, and Numeric value) and 50 fine-grained classes as shown in Table 3.1 [21]. Each coarse class contains a non-overlapping set of fine classes. The motivation behind adding a level of coarse classes is that of compatibility with previous work's definitions, and comprehensibility. The publishers also hoped that a hierarchical classifier would have a performance advantage over a multi-class classifier. One difficulty in the question classification task is that there is no completely clear boundary between classes. Therefore, the classification of a specific question can be quite ambiguous. For instance, What is the PH scale? could be a numeric value or a definition. It is hard to categorize those questions into one single class. Mistakes will likely be introduced in the downstream process if we do so. To avoid this problem, the authors assign a single question to one of 50 possible classes. This method is better than mapping to a flat taxonomy type [20].

Table 3.1: Question Taxonomy

Coarse grain	Fine-grain classes
Description	definition, description, manner, reason
Entity	animal, body, color, creation, currency, disease, event, food, instrument, language, letter, other, plant, product, religion, sport, substance, symbol, technique, term, vehicle, word
Human	description, group, individual, title
Location	city, country, mountain, other, state
Number	code, count, date, distance, money, order, other, percent, percent, period, speed, temperature, size, weight

Factoid questions are the most challenging type of questions to classify. Various approaches have been proposed to enhance the identification and the classification of factoid questions; The most popular classification taxonomy of factoid ('wh-') questions is Li and Roth's categories [20, 61]. "What's the world's longest suspension bridge?" is categorized under "Location" while we believe that it is more appropriate to consider a bridge an entity. hierarchizing based on the accuracy of meaning is very handy during these kinds of issues. The author [19] stated that prioritization is important as this particular taxonomy does not allow a question to be a part of two classes The author also found the need for a new class that constitutes either "Human: Individuals" or "Human: Groups" such as companies, teams, and universities. This specific requirement is a direct result of the restriction that a question must be classified without prior knowledge of the answer. For example, the question "Who won the Nobel Peace Price in 2012?" is impossible to classify without knowing if the answer was an organization or an individual, even if we were to ignore the possibility of multiple individuals.

3.2 QC for Different Languages

We will review studies of QC for different languages by mentioning the strategies and the methodologies used. The strength and weaknesses of each study will also be discussed as follows.

3.2.1 QC for the English Language

The work in [19] shows the need to work on QC using fine grain taxonomies to reduce the search space of potential answers and discover more accurate answers. The main focus was to identify Entity classes and enhance the answer selection by redefining question taxonomies designed by [21]. The taxonomies were defined after recognizing categorizations that would make Entity class identification very hard. For example, the question "What's the world's longest suspension bridge?" is categorized under "Location" while it is believed that it is more appropriate to consider a bridge as an entity. It is also indicated that the class Human needs a more clear boundary between Human: individual and Human: group as a result, these two classes were redefined as Human: IndividualOrGroup class category. The Fine-grain classes Year, Volume (Size), Volume (Liquid), Time, Numeric range were also included under the Number coarse grain to make classification more specific.

The work in classifying questions primarily involves three tasks [19]. These include a) extracting the Question's Syntactic Map or structure b) identifying the headword of the noun phrase in the question, while handling Entity Identification and phrase detection, and c) using rules to map words at different positions in the Syntactic Map to question classes using a hierarchical structure. The paper also explains mitigating question ambiguities by modifying the system to make it return a second possible class when there happens a misclassification due to ambiguous words. TREC dataset and DL approach were involved to train the model while using GloVe word embeddings to encode and decode the data. Training the model using a variety of fine-grain data has made DL models gain more from increased data rather than homogeneous data.

Generally, the work has made more efforts in classifying fine-grain questions to identify Entity classes to have the best answer selection mechanism. Even though the redefined taxonomies by the authors were important, the paper doesn't show a clear evaluation result. It is also ambiguous to have a Human: IndividualOrGroup class as each individual and group class have their distinct property.

QC was stated by [59] as mapping questions into one of the k classes where k represents a list of question taxonomies. These question taxonomies were taken from [21] because of the overall coverage of question types that can be used in different question answering platforms. The publicly available datasets provided by the University of Southern California, University of Illinois Urbana-Champaign, and TREC were used by manually labeling all of the question dataset. Five thousand five hundred sample questions were used for training and 500 of them for testing the system. The experiment in the paper shows that SVM achieved higher results compared to other machine learning classifiers like Decision Tree and Naïve-Bayes algorithms.

The occurrence of words more frequently is given high value than the less occurring words. But here, the authors take these rare words in the training dataset to possess as much semantic information as the rest of the more frequently occurring words and achieved 85.5%.

3.2.2 QC for the Persian Language

Question Classification for the Persian question answering system was proposed by [7] using a machine learning approach. Three methods were proposed by the researcher for feature extraction. The First method uses clustering algorithms to partition vocabulary into clusters and acquires feature vectors corresponding to each question using clustering information. Extracting features from questions using recurrent neural networks and feedforward neural networks are proposed as a second approach, which has the advantage of learning faster with less need for data. The questions are converted to a feature vector, which is obtained by the Word2vec method and weighted by tf-idf coefficients. The third approach considered Recurrent Neural Networks with a limited amount of data and showed the experiment. The authors used three datasets extracted from Wikipedia to evaluate the system's performance. The extracted datasets contained 5452 questions which were translated from an English question dataset, and the third included 5000 questions. Using these datasets the system was trained using Multi-Layer Perceptron, SVM, and RNN and gained an accuracy of 72.46 %, 72%, and 81.77% respectively. The major strength of these Persian QC is that it has considered question datasets that are even hard to be answered even by humans. Even though the Persian QC system involved three types of feature extraction methods, it has not considered preserving the word orders for training the system.

3.2.3 QC for the Chinese Language

Attention-based LSTM for Chinese question classification was developed by [12]. The implementation used deep learning-based methods to be able to mine deep information of a text. The model combines the characteristics and advantages of the convolutional neural network, attention mechanism, and recurrent neural network to not only extract the features of Chinese questions effectively but also learn the context information of questions to solve the problem that the Text-CNN model can lose position feature. The experiment was done by using Fudan University Chinese question classification data set that includes 10,243 questions from thirteen question categories. Using these datasets the attention-based CNN model has achieved 79% accuracy. After a comparative study, Attention-Based LSTM did not show significantly better performance, but slightly better than LSTM. The authors concluded that the Attention-Based LSTM model will be more effective in long texts, while Chinese questions are all short. Even though implementing the model preserving the positions of the words is very important, the experiments conducted for these models did not use the same data set for training the system by using CNN, LTSM, and the Attention-based model. This might affect the study since the performance of the system is also highly dependent on the dataset. Thus, the variation of the dataset also affects the result.

The work in [60] developed question classification for the Chinese Language by preparing a question corpus that contains 4394 Chinese questions and follows the two-layered question taxonomy. The taxonomy has 6 coarse categories and 65 fine categories. The authors removed stop-words from all question instances in the training set, and added up the frequency of each word, such as noun, verb, adverb, adjective, from each question category. The authors gave main focus on extracting features of the questions by using Chinese word segmentation, keyword extraction by removing stop-words from all question instances in the training set, and adding the frequency of each word, such as noun, verb, adverb, and adjective. Bag of Words helped the authors to extract features by grouping words in a question sentence to generate more meaningful chunks. As a result, 97.9% was achieved after training the extracted features using an SVM classifier. Here, the main focus was made on feature extraction. However, if we see DL models, it might not be necessary to go through all these long processes to extract features to get the semantic meaning of questions.

3.2.4 QC for the Arabic Language

Arabic Question Classification using the SVM method based on new taxonomy and continuously distributed representation of words was implemented by [13]. The continuously distributed representation of words was used to capture the semantic and syntactic relations between words in questions. The contribution of the paper is twofold. First, building a taxonomy for open-domain Arabic questions to capture syntactic and semantic relations between words by referring to the question taxonomies designed by [21]. Second, implement an efficient method for classifying Arabic questions. The work includes retrieving passages from documents by processing the input questions, retrieving a passage, and processing the answers. Here, we can see that QC is important in retrieving contents from documents by classifying the input questions to retrieve the relevant documents and contents from the retrieved files. To evaluate the effectiveness of the proposed method, the authors collected 1302 Arabic questions from data-sets TREC and Moroccan school books and achieved 90% accuracy. Even though the accuracy is well achieved, the system was built only for WH questions which is a restricted domain.

3.2.5 QC for the Bengali language

The Bengali QC system [14] was implemented using SVM and a Hybrid Feature Extraction method for the Bengali language. The researchers gathered 1375 questions from the Bangladesh subject and 1118 questions. As the authors mentioned, the Bengali language is not so rich to perform machine learning classification tasks but yet achieved an accuracy of 89% by considering Wh- questions, the length of the questions, and their position that might affect the system.

As Bengali question classification is at the early stage of development, a single-layer taxonomy that consists of only nine course-grained classes was used. The researcher indicated that it is possible to increase this accuracy with a larger dataset and applying a neural network algorithm.

3.2.6 Question Classification for Amharic

Question classification for Amharic was implemented by [9] under the Amharic QA system. Factoid and list type questions were studied by using SVM and HMM. The corpus used for the study is only concerned about Ethiopian history as the research focuses on a specific domain or closed domain question answering. The dataset was first preprocessed by using character normalization, short word expansion, tokenization, stop word removal, and text cleaning. To make the system understand the preprocessed questions question analysis, classification, question type identification, and query generation were involved under the question processing component. After training the system, 80% and 88% accuracy was achieved in retrieving the answers using HMM and SVM respectively. The accuracy of the system was competitive. However, considering the wider taxonomies in this domain area would have been possible instead of applying the standard question taxonomies to classify the questions.

Non-factoid question types were also studied by [6] using SVM and rule-based classification approaches for classifying and retrieving data. The design involves document preprocessing, and question analysis to retrieve documents. The preprocessing includes character normalization, short word expansion, stop word removal, stemming, morphological analysis, and indexing. The Question analysis includes question classification, query generation, and query expansion. The question classifier identifies three types of non-factoid questions Biography, Definition, and Description as shown in Table 3.2 [6]. The rule-based algorithm determines the question type by using the interrogative terms of the question and class indicative terms which can be used as a clue to indicate the class of the question type. The SVM machine learning approach predicts the type of questions based on the learned model. The performance of the SVM-based question type classifier is 83.3% and that of the rule-based question type classifier is 98.3%. It is a remarkable result having used a rule-based and machine learning approach. However, there needs to be a deeper analysis of the question types.

Table 3.2: Definition, Description, and Biography non-factoid question types

Question particles	Question types
ምን ማለት ነው፣ ምንድን ነው፣ ምንድን ናት፣ ምንድን ናቸው	Definition
ምንድን ነው፣ ምን ... አለው/አላት/አለቸው፣ ምን እንደሆነ አብራሩ/አብራሪ/ አብራራ/ግለጽ/ግለጭ/ግለጹ፣ ለምን ይጠቅማል/ ትጠቅማለች	Description
ማነው፣ ማናት፣ and ማናቸው	Biography

Amharic question classification system was also used under the “Teteyeq” [16] QA system. One thousand fifty Factoid questions were used to train and evaluate the system using a rule-based approach. Normalization, tokenization, stemming, stop-word removal was applied to preprocess questions. the question processing component has the subcomponents question classification, expected answer type determination, and query generation. The query generation is used to process the question to help generate a well-structured query to find out the type of question it belongs to after analyzing questions to determine the category the question belongs to.

Table 3.3 [16] shows the factoid questions used by the author. These questions could be further classified into different question types based on the question particles. It is also possible to expand the question taxonomies into more specific ones to get a more relevant answer retrieval system. If we see Numer, ስንት ስንቱ ከስንት ለስንት ምን ያህል can have taxonomies like Age (እድሜ), Count (ቁጥር), Date (ቀን), Distance (ርቀት), Money (ገንዘብ), Percent (ፐርሰንት), Time (ጊዜ), Weight (ክብደት) and so on. It is also suggestible to use more question datasets to implement the system. Regardless of this, the author has scored 89.9% accuracy on correctly retrieving the answers after giving more focus on categorizing questions.

Table 3.3: Amharic factoid Question types

Question particles	Question types
ማን ለማን ማነው የማን በማን	Person name, place
የት በየት ወዴት የየት እስከየት	Place
መቼ በመቼ እስከመቼ ለመቼ	Time
ስንት ስንቱ ከስንት ለስንት ምን ያህል	Numeric, time

3.3 Summary

From the related work we have reviewed, it is shown that most authors [19] use question categories designed by [21]. Even though this taxonomy is used in most studies, it is considered to be far from a standard question taxonomy since there are classes not included in the hierarchical question taxonomy. The researchers [21], studied the multiclass taxonomy on the dataset published in the year 2002. Hence we need a more updated standard question taxonomy that involves a wide range of question datasets.

The question taxonomy [21] was also adopted by authors [7] to use it to classify Persian questions. The author used one of the DL approaches that is LSTM by additionally using SVM for the QC. However, converting the question data into a vectorized form is a long process that involves combining both a word2vec and a TF-IDF algorithm. While converting the questions into a word2vec model to extract features, it is necessary to consider the positions of words that might affect the semantic meaning. Even though the work [19] focuses on increasing classifying questions accurately, the positions of the questions that might affect the semantical meaning were not considered.

Nevertheless, we have seen QC for various languages, we couldn’t find an independently studied QC implemented for Amharic as a broad subject. The studies were on a restricted domain that is not generic. Nevertheless, QC is studied in Document retrieval [16], and QA [9]. Studying Amharic questions under QA and other applications, the authors have given much attention to extracting features from questions by using rule-based and bag of words based semantic as well as syntactic analysis to extract features as presented in Table 3.4. Even though we have to study feature extraction methods to apply to the AQC model, most DL approaches i.e. CNN is used for processing questions by extracting the necessary contents that are useful for training and classifying the questions. In light of this, we propose a generalized AQC model that is trained on a more generic Amharic question dataset and multiclass taxonomies.

Table 3.4: Related work on Amharic Questions

Proposed System	Language	Dataset	Question Type & Result		Author & Date
SVM and Heuristic rule-based approach	Amharic QA	Documents from Ethiopian tourism	Listing (non-factoid) Result 55%	Closed domain	Brook Eshetu [17] (In 2013)
SVM HMM	Amharic QA	Ethiopian History books and websites	Factoid and Listing (Non-factoid) SVM 73% & HMM 65%	Closed domain	Medhanit Getachew [9] (In 2019)
Java and gazetteer (instead of named entity recognizer)	Amharic QA	Web & Ethiopian news papers	Factoid	Closed domain	Seid Muhie [18] (In 2009)
Rule-based approach, HMM	DS for Amharic	156 address records from websites		Closed domain	Fitsum Seyoum (In 2015)

Chapter 4: Design of Amharic Question Classification

This Chapter covers the design and prototype development of the AQC model using the CNN DL approach. Question Preprocessing, Question Processing, and Categorization (QC) are the major components in the AQC system architecture that are utilized to accomplish the Amharic QC. We will discuss the AQC system architecture along with the system interaction in the training and testing phase.

4.1 System Architecture

AQC possesses a similar system architecture as every other QC system in the machine learning approach. The common components in various Question Classifications are Question Preprocessing, Question Processing, and Question Classification. The only difference with the DL approach is in the word representation and how the system extracts as well as learns features. Figure 4.1 shows the overall system architecture of the AQC model that involves the mentioned components.

Question Preprocessing is the first step in the AQC model. This component helps us get more structured data by involving Normalization, Special Character Removal, and Tokenization to the raw data. The next step in training the AQC model is to let the system understand the preprocessed input data by representing them in a vectorized form. We use Word Modeling in the training component to convert the preprocessed structured data into a vectorized form.

The Word Modeling takes a preprocessed Amharic Question Corpus and extracts a semantic relation of words using a sparse distribution. The Word Modeling also saves the semantically related vectorized words in the Word2vec Model for later use in the Testing component. This word vectorizer gives a richer semantic meaning to make the machine understand the input questions well.

Question processing is where we make the machine understand and learn from the input questions using a supervised CNN approach. The CNN extract features from the Amharic Question Corpus after representing them in a word matrix. The learned features are then saved in a Question Classification Model to evaluate the performance of the AQC model by using new data that were never used in the AQC model before.

The Testing component shares similar architecture as the training component by using the preprocessing and vectorization of words similarly. The Question Classification can automatically categorize questions without necessarily extracting or learning the features again. Figure 4.1 shows how we evaluate the system by loading the AQC learned model using the Question Classification Model.

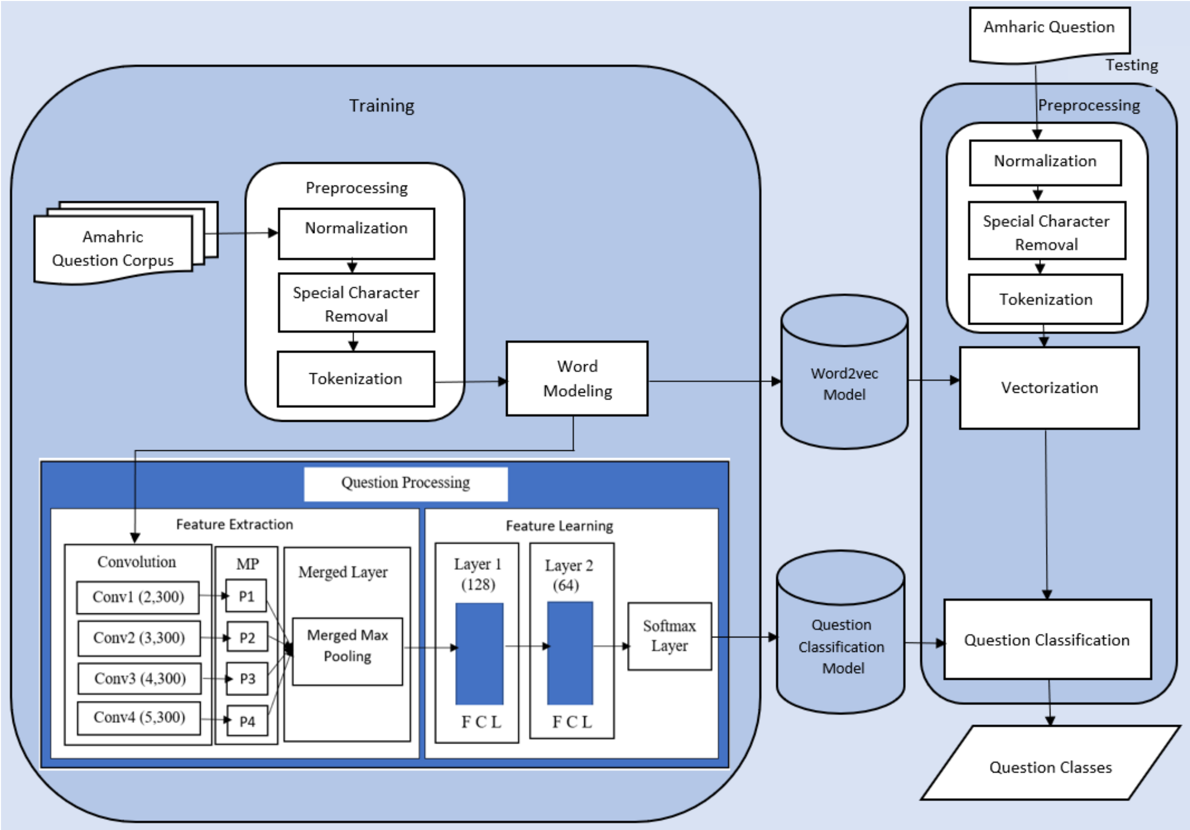

Figure 4.1: Amharic Question Classification AQC Model

4.2 Training

The Training component is part of the AQC system architecture that is used to train the AQC model using a labeled Amharic Question Corpus. Various tasks are performed to train the system: Question Preprocessing, Question Processing, and Question Classification.

Question Processing is the major component in training the AQC model using the CNN approach. Question Processing is all about representing the vectorized words into a fixed size word matrix, and train the AQC model to generate a learned knowledge base i.e Question Classification Model as shown in Figure 4.1.

The AQC model learns from the labeled input data by managing its loss value. Backpropagation plays the biggest role in minimizing the loss value and maximize the training accuracy by letting the system learn the most valuable features with a minimum loss value. The AQC model obtains a loss value by looking at the difference between the predicted value and actual labeled data when classifying questions.

This section will go into detail about training the AQC model by discussing each component: the Preprocessing, the Word Modeling, and the Question Processing.

4.2.1 Preprocessing

Preprocessing is a process of structuring the raw data using different data cleaning and formatting techniques to make them suitable for the learning process. The AQC model uses the Amharic Question Corpus to classify questions by initially preprocessing them. The Preprocessing component plays a vital role in letting the AQC model learn from well-structured data.

The AQC model generally acquires 8000 Amharic Questions to train and evaluate the system. Five thousand of them are labeled and stored in the Amharic Question Corpus. These questions are prepared in three different forms: Surface, Stemmed, and Lemmatized form. Even though stemming and lemmatization are usually included in the preprocessing component, we let each dataset maintain their form by stemming and lemmatizing questions independently using a semi-automated system. This helps us use a common preprocessing component for each of them.

Annex A shows sample questions taken from the Amharic Question Corpus from the three different forms: Abbreviation: expansion " "ኢ.ፌ.ዴ.ሪ." ተዘርዝሮ ሲጻፍ እንዴት ነው?" is the surface form, Abbreviation: expansion "ዒ.ፌ.ዴ.ሪ." ተ ዘርዝር አ ሲ ጻፍ እንዴት ነው?" is the stemmed form, and Abbreviation: expansion ""ኢ.ፌ.ዴ.ሪ." ተ ዝ-ር-ዝ-ር አ ሲ ጽ-ህ-ፍ እንዴት ነው?" is the lemmatized form. Even though the questions have three different forms, we can use a common preprocessing component to clean the data from special characters like the quotation mark (" "), normalize characters (ኢ/ዒ) by merging similar characters into a common form and breaking down each question into a token to analyze them semantically.

If the questions were to be stemmed and lemmatized using the AQC Preprocessing component, we would not have used a common preprocessor for each of the different question datasets. Instead, would stem and lemmatize questions separately by designing three different Preprocessing components. To overcome this issue, we stemmed and lemmatized some of the Amharic Question Corpus using a semi-automated preprocessor (stemmer and lemmatizer) outside the AQC model.

Normalization: Amharic alphabets, like “ዔ/ኤ”, “ሃኅኃሐሓኻ/ሀ”, “ሑኁዅ/ሁ”, “ኂሒኺ/ሂ”, “ጸ/ፀ”, “ጾ/ፆ”, “ጹ/ፁ”, “ጺ/ፂ”, “ጻ/ፃ”, “ጼ/ፄ”, “ጽ/ፅ”, “ሠ/ሰ”, “ሡ/ሱ”, “ሢ/ሲ”, “ሣ/ሳ”, “ሤ/ሴ”, “ሥ/ስ”, “ሦ/ሶ”, and “ዓኣዐ/አ” appear interchangeably in different words without changing their meaning. These characters have the same pronunciation or phonetics, having similar meanings but different representations. If we see the word, “አመት ምህረት”, it can be written as “ዐመት ምህረት” and “ዓመት ምሕረት” by using characters አ/ ዐ / ዓ and ሀ/ሐ in different words correspondingly.

Therefore, we normalize the alphabets that are used interchangeably using Algorithm 4.1. We also removed unused extra spaces using the rstrip() function.

Algorithm 4.1: Normalization

```
Input: Filename
Output: Outputfile
FindReplace<-[('ዔ','ኤ'),('[ሃኅኃሐሓኻ]','ሀ'),('[ሑኁዅ]','ሁ'),
('[ኂሒኺ]',’ሂ'),('ጸ','ፀ'),('[ጾ]','ፆ'),('[ጹ]','ፁ'),('[ጺ]','ፂ'),('[
ጻ]','ፃ'),('[ጼ]','ፄ'),(  [ጽ]','ፅ'),( '[ሠ]','ሰ'),( '[ሡ]','ሱ'),('[ሢ
]','ሲ'),('[ሣ]','ሳ'),('[ሤ]','ሴ'),(     '[ሥ]','ስ'),(     '[ሦ]','ሶ'),
( '[ዓኣዐ]','አ')]
Readfile=Filename.readline()
    for CharPair in FindReplace
      Outputfile<-
Readfile.(charpair(FinReplace[0],FinReplace[1]))
     end for
         return Outputfile
```

Special Character Removal: Characters other than alphabets are removed from AQs. These non-alphanumeric characters are often found in punctuation marks like adding quotations to refer to other people’s speech, as well as symbolizing currencies. These non-alphanumeric characters add no value to the data other than causing noise to the system.

To get rid of these noises such as unwanted characters, we use a regular expression (regex). Regex is usually used in strings to "find" or "find and replace" characters as shown in Algorithm 4.2.

Algorithm 4.2: Special character removal

```
Input: Filename
Output: Outputfile
SpecialCharacter<-!@$%^&*()*&^]+, ''
Readfile=Filename.readline()
    for (i=1, i<length(Filename), i++)
       if (Filename[i]== SpecialCharacter):
          Outputfile<-Remove(Filename[i])
       end if
    end for
   return Outputfile
```

Tokenization: Tokenization helps us to analyze the semantic meaning of each word in the AQs at a token level. We achieved this by using white spaces and the split() function as we have presented in Algorithm 4.3. This function could also consider colons or "ሁለት ነጥብ" (:) for this purpose. However, our dataset (the AQs) does not utilize these punctuation marks to separate the words.

Algorithm 4.3: Tokenization

```
Input: Textfile
Initiate i to 0
     Textfile=Textfile.split into newline
     Tokenizedarray <-[]
        for i to len(Textfile) do
        first_element = Textfile[i].splitaftereveryspace
      tokenizedarray.append(filename[i])
        increment i
     end for
return tokenizedarray
```

4.2.2 Word Modeling

The Word Modeling in the AQC system architecture is a process used to represent words in a vectorized sparse distribution after understanding their semantic relation. These words are extracted from the preprocessed Amharic Question Corpus.

We use the Skip-gram word modeling tool to train the system to identify the semantically related words by using the tokenized input words. We present each word using two parameters (focus word, context word). Each word in the question dataset is assigned a unique number in the word vocabulary we create. We use one-hot encoding to transform these numeric data into 0s and 1s. The Skip-gram model helps our system learn the relation between similar words by predicting the target or focus word from the rest of the words that are taken as a context.

For example: “Entity: letter የላንቃ ተናባቢ ድምጾች የምንላቸው የትኞቹ ናቸው?” we make the machine understand the relation of each word by creating all of the possible focus words and the context words out of the given question. (የላንቃ, ተናባቢ), (የላንቃ, ድምጾች), (ተናባቢ, ድምጾች), (ድምጾች, የምንላቸው), (የምንላቸው, የትኞቹ), (የትኞቹ, ናቸው),(ተናባቢ, የላንቃ), (ድምጾች, የላንቃ),(ድምጾች, ተናባቢ), (የምንላቸው, ድምጾች), (የትኞቹ, የምንላቸው), and so on. We then create the word vocabulary by assigning each word in our dataset a unique number. i.e., ‘የላንቃ: 0’, ‘ተናባቢ:1’, ‘ድምጾች:2’, ‘የምንላቸው: 3’, and ‘የትኞቹ: 4’. The words are distributed across the column using one-hot encoding to make a 0s and 1s representation. The Skip-gram makes use of the parameters (focus word, context word) and the represented words using the binary number to create a vectorized sparse distribution.

The vectorized words are then fed into the CNN Question Processing component for further processing to extract and learn features. The rest of the trained vectorized words are stored in the Word2vec Model to automatically vectorize questions without modeling other similar words again.

4.2.3 Question Processing

Question Processing represents the vectorized words into a 32*300-dimensional word matrix, extracts features (keyword) and, derives the expected question type. This component is generally very useful to let the AQC model learn from the vectorized Amharic Question Corpus by involving the mentioned operations in Question Processing.

The question processor outputs trained models after learning features. The saved trained AQC models are then used in the Testing component to evaluate the Amharic Questions.

The overall CNN architecture in the Question Processing component consists of two main tasks. The first one is the feature extractor, which uses the convolutional as well as the max-pooling layer to make the CNN understand questions and the classifier which is the softmax layer that lets the system learn the features by referring to the predicted and actual labeled data.

Feature Extraction

This is a mechanism where we identify valuable features (elements) from the questions. The feature extractor uses the convolutional layer to slide across the word matrix, to extract features from various combinations of words.

The questions in our Amharic Question Corpus have a maximum length of thirty-two. As a result, we have a fixed input word matrix size to be 32 which is the maximum sequence of words in the questions, and 300 is the dimension of the vector that is sparsely distributed close to the input words by their semantic similarity. The convolutional layer represents the vectorized words retrieved from the Word Modeling into a 32*300-dimensional vectorized word matrix.

The height or region size of our filter considers bi-gram up to a pent-gram (2-5 words) at a time. This means the machine learns using an n-gram: bi-gram, tri-grams, quad- grams, and pent-gram filters during convolution. These n-grams are determined by the height of the kernel. If the height quals to 2 then the kernel is learning a bigram word matrix. Bi-gram filter learns two tokens at a time like “ማን ነው” “የት ነው” and “ስንት ነው”. As the height of the kernel increases, the network will be able to learn phrases of various lengths that can help them effectively classify a given question.

If we consider n-grams for the phrase “ውጤታማነትህን ሳላደንቅ ማለፍ አልችልም ወይም ይከብደኛል”, It should be classified as a positive word. If we break down the phrase into a bigram “ውጤታማነትህን ሳላደንቅ ”, This should be classified as a negative phrase. However, if we consider quad-grams, “ውጤታማነትህን ሳላደንቅ ማለፍ አልችልም ” each will have a different classification category. It again learns it as a positive category. Thus, the network will be able to learn from various filters that are activated according to the phrases found in the input questions.

The convolutional layer is in charge of the operational combination of the input matrix as well as the trainable weight matrix (kernel or filter). This layer extracts features by analyzing the semantical relation from the amalgamation of words that give more meaning. For each word in the embedding vector, a single convolutional layer with a four number of filters is applied. Table 4.1 shows an instance of what the word matrix looks like for AQs in Surface, Lemmatized, and Stemmed forms. As the word matrix shows, the Word Modeling has represented these words having been trained on the surface words since the words have maintained their similarity even after being stemmed or lemmatized.

Table 4.1: Word matrix of a word2vec model

Amharic Questions			Representing words in 300 dimensional semantically related words		
Surface form	Stemmed	Lemmatized			
የካብኔ	ካብኔ	ካብኔ	0.08	0.065	0.113
ትርጉም	ትርጉም	ትርግም	-	-	-
የተጠቀሰው	ጠቀሰ	ጥቅስ	-0.33	-0.25	0.07
አዋጅ	አዋጅ	አዋጅ	0.04	-0.28	-0.29
ቁጥር	ቁጥር	ቁጥር	-0.15	-0.112	0.14
ስንት	ስንት	ስንት	-0.04	-0.19	-0.13
ላይ	ላይ	ላይ	-0.65	-0.54	0.47
ነው	ነው	ነው	-0.16	-0.20	-0.19

The word "ትርጉም" in the second row of Table 4.1 shows a slight difference from the surface and stemmed words by changing all the characters into a Sadis form including the letter "ጉ" ("ትርግም"). Even though the word modeling can represent the surface and stemmed words similarly relative to the lemmatized words, it would not get meaning from the lemmatized word such as "ትርግም".

As Pedro [37] stated, Kernels record the accurate position of features. As the filter kernel convolves across the word matrix, each position causes activation of the neuron usually a non-linear activation function called Relu to form an output feature map. A feature is extracted when the convolutional filter convolves across the entire input layer or the vectorized word matrix by extracting features from the n-gram words. The convolution operation lets the machine analyze the semantic meaning of questions by combining each word at a different level (bigram to pent-gram). Feature extraction is applied using an activation function Relu to obtain feature vectors. These activation functions improve the training of the neural network by activating the most accurate values while preserving the completeness of the information during feature extraction.

The kernel learns by first taking the learnable weights randomly and by updating their weight value by referring to their loss value to minimize the loss. The loss value is determined by looking at the gap between the predicted output and the actual labeled sample data.

While convolving across the word matrix padding is added for some scenarios. Padding could either be the "SAME" or "VALID". When the input size equals "SAME", it requires the filter window to slip outside the input map by adding zeros on the right and left sides of the word matrix. It helps not to lose information. "VALID" padding is when the Filter window stays inside the input map, which means when no padding occurs. We chose our padding to be "VALID" because it still covers the information we need without adding extra computation.

The max-pooling (MP) in Figure 4.1 extracts the most obvious or the highly weighted values from the extracted feature maps resulted from the convolutional layer. The pooling layer takes the maximum value acquired while convolving the input word matrix. The maximum value represents terms that are activated by having the most valuable information from the rest of the extracted features.

If we see the question "የካብኔ ትርጉም የተደነገገው አዋጅ ቁጥር ስንት ላይ ነው?", the convolutional layer first takes an input as a vectorized word matrix and outputs the learnable feature maps by representing the combination of words by a specific vector. Table 4.2 shows "የካብኔ, ትርጉም, የተደነገገው" represented by a vector 0.18. This vector can represent the word "meaning" in the sparsely distributed Word2vec model which represents the trigrams by a common similar term represented by the vector. "አዋጅ ቁጥር ስንት" represents questions by having a maximum feature combination of its category Number. The tokens, "ቁጥር" and "ስንት" especially appear more frequently in the Number category in questions like "የነባር ይዞታ አስተዳደር በተመለከተ አንቀጽ ቁጥር ስንት ላይ ተደንግጓል?" or "Number:article የህዝብ ቆጠራ ኮሚሽን ተጠሪነቱ ለማን እንደሆነ የተገለጸው በስንተኛው አንቀጽ ላይ ነው?". i.e., Cat sitting and Dog resting, both represent animal resting regardless of the difference of the pets. Thus, the Word Modeling learns weighted filters that represent various terms by a common vectorized word.

Hence, we can easily get the meaning of the questions by taking the three tokens without further processing the rest of the words. “Which part of the body gets easily affected by diseases? “, “What is the most colored part of the eye?” or “ቀለምማ የሆነው የዓይንናችን ክፍል ምን ይባላል? የትኛው የውስጥ ክፍላችን ቶሎ በበሽታ ይጠቃል?”, “አየር መታገድ የሚጀምረው አየር የትኞቹን የሰውነታችንን አልፎ ማንቁርት ሲደርስ ነው?” All of them express a body part by just combining the word “የሰውነት ክፍል” + the interrogative word and by learning the common tokens in the dataset for Entity: body category.

Table 4.2: Feature extraction using a Trigram filter

Trigrams	Vector
የካብኔ, ትርጉም, የተደነገገው	0.18
ትርጉም የተደነገገው አዋጅ	-0.25
የተደነገገው አዋጅ ቁጥር	-0.112
አዋጅ ቁጥር ስንት	0.14
ቁጥር ስንት ላይ	-0.13
ስንት ላይ ነው?	-0.19

The feature extraction helps the system to easily assign questions into their appropriate category without taking the whole value in our feature map. This lets the AQC model reduce the less valuable features and speed up their computation. We then merge the max-pooling layer to form the FC layer. The merged max-pooling layer combines all the maximum values of each n-gram filter that convolve across the word matrix using the concatenation function.

Feature Learning

The CNN learns the extracted features obtained by convolving and pooling the maximum feature values. Learning the extracted features is known as feature learning. The CNN learns features by applying weights to each input and by connecting all neurons in the system. The neurons that incorporate input features with the respective weights are connected using the FC dense layer. The more fully connected layer, the more the CNN layer tends to work better [62]. As a result, we used two fully connected layers (FC) in the AQC model.

The first FC layer receives the feature vectors extracted by the pooling layer, and the second FC layer is used for classification by looking at the classification labels that are labeled in the softmax layer. The FC layers have a dropout of 0.5 and 0.7 to avoid overfitting while we train the model. We chose these drop-out values, having the understanding that the model learns well as the dropout value gets closer to 1 or between 0.5 and 0.8. The model learns relevant features by dropping the least valuable extracted features. Dropout is useful in regularization by taking the relevant features.

The weights in the CNN training component are initially randomly assigned since the AQC was not priorly trained on the system. However, as the training component continues to learn from the extracted features, it starts to save the feature patterns with the activated neurons in the FC layer. Hence, saving these patterns is important to identify features based without necessarily going through the question processing component i.e., feature extraction and feature learning.

4.3 The Testing Component

The Testing component is somehow similar to the Training component except that it uses a learned Question Classifier Model to test the Amharic Questions. This component tests unlabeled questions that were never fed into the system before to evaluate the performance of the AQC model. The Question Classification Model gets saved and updated every time the Training component passes through some training. The Testing component as shown in Figure 4.1, retrieves the saved Question Classification Model to test questions without necessarily going through feature extraction or the learning process.

The trained model can automatically predict the vectorized input questions by identifying their pattern using the learned Question Classification Model. However, We can only test the stemmed or surface Amharic questions using a trained AQC model on a stemmed or surface AQs. We use the trained model with the corresponding Amharic questions that the model was priorly trained on.

However, since we do not use a Softmax layer to store a numeric value for the class categories, we assign a numeric value for each taxonomy in the testing component. The learned model compares the predicted class with the learned features and displays the evaluation as shown in Figure 4.3.

```
['አንደኛ' 'ተወዳጅ' 'የሆነው' 'የስፖርት' 'ጨዋታ' 'ምንድን' 'ነው?']
Predicted: ENTY Confidence : 98.45%
Predicted Entity Sub Categ: sport Confidence : 90.824%
['ላይ' 'በጣም' 'ፈጣን' 'የተባለው' 'ስፖርታዊ' 'ጨዋታ' 'ምንድን' 'ነው?']
Predicted: ENTY Confidence : 98.308%
Predicted Entity Sub Categ: sport Confidence : 75.314%

['ለመቆጣጠር' 'ጥቅም' 'ላይ' 'የሚውል' 'ኬሚካል' 'ምን' 'ይባላል?']
Predicted: ENTY Confidence : 97.526%
Predicted Entity Sub Categ: body Confidence : 51.467%
['ሆርሊክ' 'ምን' 'ዓይነት' 'ምርት' 'አገኘ?']
Predicted: ENTY Confidence : 99.475%
Predicted Entity Sub Categ: product Confidence : 51.104%

['ሌላ' 'ቃሉ' 'ምንድን' 'ነው?']
Predicted: ENTY Confidence : 98.505%
Predicted Entity Sub Categ: letter Confidence : 81.298%
['መከፈቻ' 'የሚስጥር' 'ቃል' 'ምንድን' 'ነው?']
Predicted: ENTY Confidence : 64.61%
Predicted Entity Sub Categ: letter Confidence : 74.978%
```

Figure 4.2: Test component in AQC model

4.3.1 Preprocessing

The preprocessing component receives the Amharic Questions as input. The questions here are also in three forms sharing a similar process i.e., Normalization, Special character removal, and Tokenization as the training preprocessing component. Each of the preprocessing components are discussed in Section 4.2.1.

4.3.2 Vectorization

After preprocessing the Amharic questions, we vectorize each question using the knowledge base i.e., the Word2vec Model. Even though we have applied one-hot encoding in the Word Modeling, it might not be applicable in getting the semantical relation of words from the given questions. The trained Word2vec Model solves this issue by training on a large amount of data to identify the semantically related words using the Word Modeling.

The Word2vec Model stores the learned vectorized questions in a 300-dimensional word vector. Its purpose is to mainly store the modeled words with their semantically related vectorized words. The Vectorization in the Testing component of the AQC system architecture converts the preprocessed raw data into a vectorized form by using the learned Word2vec knowledge base Model. We get the word matrix from the stored vectorized trained data using a lookup table functionality. For instance, The question “ከየት መጣህ?” is mapped to the vectorized words using the lookup table that search for the words “ከየት” and

“መጣህ?” with close sparse distribution as “ከየት" =(0.2, -0.4, 0.7, ...) and “የት" = (0.0, 0.6, -0.1,..). Each word is mapped to a vector with multi-dimensional related vectorized words. This makes the Testing process easier by getting the tokenized input words alongside the related vectors.

On the other hand, words with a different meaning are represented with a vectorization far from the target word. The Word2vec Model is trained on words having similar words across the 300 dimensions. Even though there are 50, 100, 200, and 300-dimensional word embeddings, we use a 300-dimensional word vector to get a more accurate representation of the words. The represented words in the Word2vec model are loaded into our AQC model to represent the input questions with a 300-dimensional word2vec word matrix using the codecs tool.

Researchers [6, 10] used to rely on preprocessing (stem or lemmatize) data to train their system using the preprocessed base words. The intention is to make machines understand different words that vary due to affixes, quantifiers or tenses by extracting the base equivalent term. Word vectorization has brought a better solution in representing these similar words using a sparse distribution or representing similar words closer to each other instead of depending on the base words only. As a result, we do not necessarily preprocess (stem or lemmatize) our Amharic questions since the word vectorizer already lets the AQC model understand all the semantically related words.

We intend to see the performance of the Word2vec Model that was priorly trained on the surface Amharic Question Corpus by considering words that stay similar even when they are preprocessed. This is to see how applicable the Word2vec Model is for the stemmed and lemmatized words as well. If we see the interrogative words “ምን”, “ምንድን”, “በምን”, “የምን”, “ምንድናቸው”, “ከምን”, “ምናቸው”, “የትኞቹ”, “የየትኛው”, “በየትኞቹ”, “የትኛውን”,”ተዘርዝሮ”, “ሲዘረዘር”, “መቼ”, “በስንት”, “ስንቱ”, “በስንተኛው”, and “ከስንተኛው”, they are stemmed and lemmatized as “ማን”, “ምን”, “የት”, and “ስንት”. This indicates that the Word2vec Model understands these words after lemmatization and stemming. The surface word “መ/ቤቶች ተዘርዝሮ ሲጻፍ እንዴት ነው?” is stemmed or lemmatized as “መ/ቤት ዘርዝር ጻፍ እንዴት ነው?” Or “መ/ቤት ዝርዝር ጽህፍ እንዴት ነው?”. Even though we get some difference in the form of the words like “ዘርዝር”, “ዝርዝር”, “ጻፍ” and “ሲጻፍ” respectively, they continue to make use of the Word2vec Model in representing the words.

4.3.3 Question Classification

The Question Classification is like the intersection between the Question Classification Model and the Vectorized Amharic Questions that we get from the Testing component. The Question Classification Model stores the learned features using the labeled Amharic Question Corpus. The model is a set of saved trained models after training on coarse and fine grain categories. We load the saved knowledge base models using the Question Classification to evaluate the Question Classification Model.

The Amharic Questions are classified at a hierarchical level. The taxonomies were redefined by adopting the multilabel categories designed by [21] and also analyzing various taxonomies studied by [20]. The hierarchical taxonomy consists of coarse grain and fine grain taxonomies.

The Question Classes in a coarse grain category include generic classes like Abbreviation, Description, Entity, Human, Location, and Number. The fine-grain taxonomies under the coarse grain subcategory contain the classes listed in Figure 4.2. The AQC model is a multi-class, single-label prediction using a categorical_crossentropy as the loss function and softmax as the final activation function. The hierarchical level classification makes the retrieval system to be more specific and accurate.

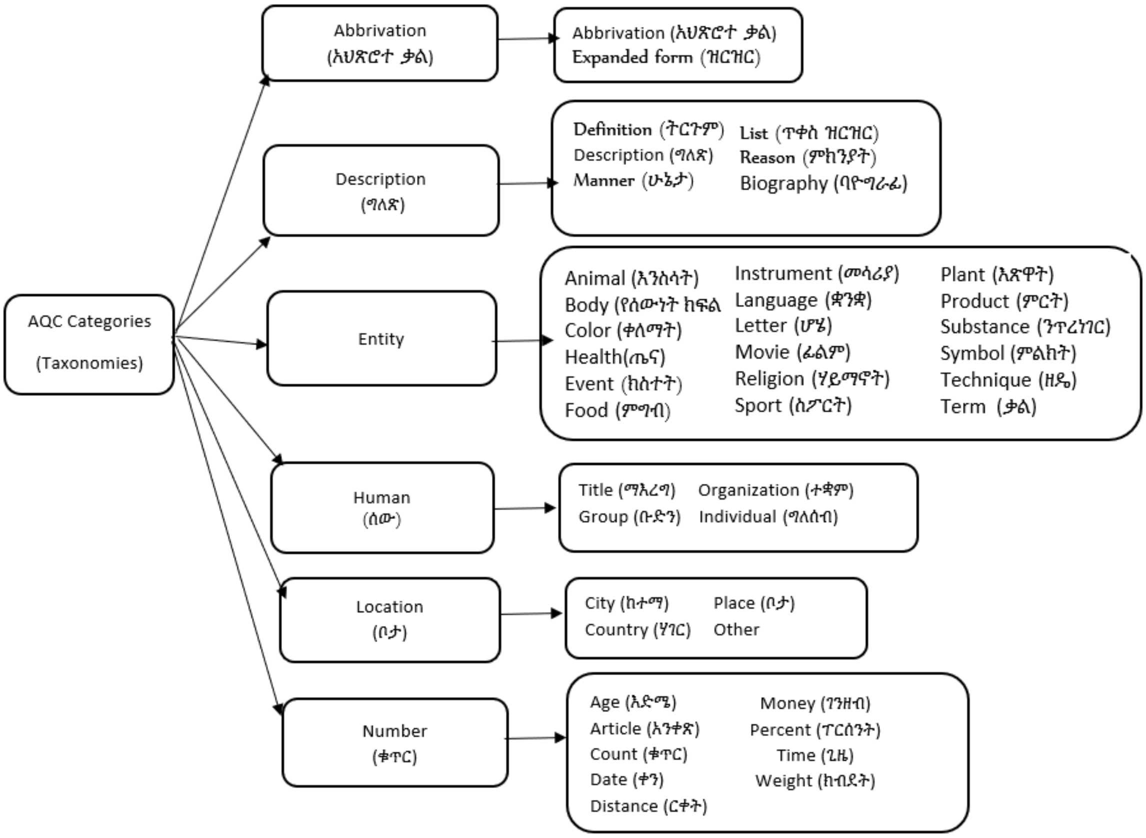

Figure 4.3: AQC hierarchical taxonomies

The softmax layer classifies questions by converting the vectors (weighted sum) extracted by the previous layers (convolutional layer and fully connected layer) into a probability vector that adds up to produce a number close to one.

Summary

The Design of the AQC model was discussed by explaining the materials as well as methods used for the prototype. The AQC model has three components the preprocessing, question processing, and classification. The preprocessing component takes raw data to preprocess them using Normalization, Special character removal, and Tokenization. The Amharic questions should always go through the preprocessing component to structure the questions and prepare them for the next process.

The AQ dataset was organized in three forms: Surface, Stemmed, and Lemmatized. To preserve the form of the three-question datasets, we have not included the lemmatization and stemming processes within the preprocessing component. The main intention here is that, to study each dataset using the AQC model as well as the trained word2vec model and also see their performance. Even though the word2vec model was trained using the surface words, we have noticed that the stemmed and lemmatized words share common interrogative words as shown in Table 5.2. Hence, we intend to see the possibilities of training and testing the trained (stemmed and lemmatized) datasets using the word2vec model that was priorly trained using the surface words.

The AQs were labeled hierarchically. We redefined these hierarchical question taxonomy as shown in Figure 4.6, after studying the most used Question taxonomy [21] and the answer types of our Amharic question dataset. We can use the AQC model for retrieving an open domain or generic data by incorporating AQC as part of any Amharic question analyzer within the question category scope.

The AQC model trains in a supervised way by using a tagged dataset. This model was developed using a CNN DL approach. The CNN different layers for various tasks. It uses a convolutional layer, max pool layer, and the classification or softmax layer for extracting features and training the system. Researchers [12] use the Flattening layer after the max-pooling layer to convert the pooled layers into a single 1D layer. What makes the AQC model different from the rest is, it trains the system without the Flattening layer by using the Global max-pool layer. This pooling layer extracts one feature that has the maximum value from the whole extracted features of each n-gram model.

Chapter 5: Implementation and Evaluation

This Chapter presents the development processes of the AQC model with the tools and techniques used for the implementation of the prototype. We also have included the experimental results gained from validating and testing the model.

5.1 Dataset Preparation

Machine Learning approaches train themselves from their inputs and performs other related tasks without the need to learn everything. They improve their performance by learning from their sample input data. Thus, data has a huge role in the machine learning process. Stanford Question Answering Dataset (SQuAD) and Google AI have prepared well-structured question datasets open to their users. We on the other hand could not find a well-organized Amharic question dataset for this work.

Regardless of this issue, we have collected 8000 generic Amharic questions from Websites, Amharic books, Ethiopian National Exams, Magazines, and communication media like Tameson, Bahrdar FM Question and Answer radio program are also some of the important sources for our work. Even though we tried to use the Spiderling corpus tool to obtain questions from websites to build a question corpora, the installation packages and dependencies were a challenge that took time and effort.

Among the data we collected, 5000 question samples were used for training by labeling them into their respective taxonomies. Each taxonomy has its question set such as Abbreviation 100, Description1000, Entity1600, Human 1000, Location 600, and Number 1100. Even though we have mainly used six of the multilabel taxonomies for this work, we have also tried to study Verasity questions using a 300 question dataset. From the 5000 training samples, 20% of them were used for validating the model. While 3000 sample data was used for testing the model during training and the other half for testing the already trained model.

Even though stemming and lemmatization are mostly included in the preprocessing component, we have independently preprocessed them by using a semi-automated system developed by Tilahun Yeshambel [17].

The semi-automated preprocessor stems and lemmatizes words that were previously fed into the system. Abbreviations, interrogative words, and stop words were some of the words that were not included in the semi-automated system. As a result, we had to manually stem and lemmatize words that could not be altered by the system. Some of these words are “መ/ቤት_ኦች, በ_አህጽር_ኦት, ተንትን_ኢ, ተንትን_ኡ, and ሲ_ዘረዘር”.

We use stemming, to remove the affix and prefixes of the Amharic questions by analyzing the derivationally related words that have similar meanings. As Melese Tamiru [61] stated, stemming removes affixes that have changed their form by applying the tense, quantifier, gender, and case of a word. Moreover, in the case of removing suffixes with vowels, the last character of the word after the removal of the suffix is changed to “sades” (the sixth order of a character). Stemming usually changes the forms of Amharic verbs. A word will be stemmed only if it is not a Noun (a person or an entity name) [6]. For example, in the question, “ልጅ ኢያሱ ማን ናቸው?”, “ተክለ ጻድቅ መኩርያ ማን ኸ ው?” and “ኢ.ፌ.ድ.ሪ.” ተ ዘርዝር ኦ ሲ ጻ-ህ-ፍ እንዴት ነው? Nouns like “ልጅ ኢያሱ”, “ተክለ ጻድቅ”, and “ኢ.ፌ.ድ.ሪ.” stays the same.

A root or lemmatized form is usually a sequence of three consonants known as radicals. The lemmatization extracts the base words by doing a morphological analysis. The question “የአለም አቀፍ የገንዘብ ድርጅት በአህጽሮት ሲጻፍ እንዴት ነው?” is lemmatized as “አለም አቀፍ ገንዘብ ድርጅት አህጽር ጽህፍ እንዴት ነው?” extracting a root word gives the base words relative to the stemmed ones. If we see “ስብር” is the root or lemma of “ስበር, ስብር, ሰበር, አሰበር, ተሰበር, and ሰባበር” while “ሰበር, ሰብር, ስበር/ስበሪ, ሰብሮ/ሰብራ, and መስበር” is what we get when we stem the word “ሰበረ” due to affixes.

The surface words as “አይ ኤም ኤፍ የምን አህጽሮተ ቃል ነው?” , “የሊባኖስ ባንዲራ የምን ዓይነት ዛፍ ምስልን ይዟል?” are easily undersood by the AQC model. Since our word2vec model represents them along with their semantically related words. As a result, we are not expected to stem or lemmatize them.

To use these datasets for classification, we need to label them according to their answer type. Hence, we redefined these taxonomies based on the most used hierarchical taxonomy [21] because of the large training set the taxonomies were prepared on and which is seen as most suited for domain-independent QA [19]. Our taxonomy was redefined by adding Veracity(Yes/No & True/False) and Desc (List & Reason) and Number (Age & Article) classes. Based on this, the taxonomies prepared for AQC include 6 Coarse_grain and 43 Fine_grain classes. Abbreviation 2, Entity 18, Description 6, Human 4, Location 4, Numeric 9 as listed in Annex B. That is, the training folder contains 6 different text files for each broader class containing the question samples. For example, abbr.txt has questions about abbreviations, desc.txt has questions about descriptions. We acquired the subclasses of our taxonomy by observing the characteristics of the Amharic question types with their answer type.

Labeling or Tagging: The taxonomies we prepared were used to tag or label our training samples. There are some techniques for "Automatic labeling". One of them is the "active learning" approach. The premise is that once we hand-label a few hundred points, a model powered by embeddings will be pretty accurate when it thinks it is accurate and can be trusted to label new points by itself.

Thus, we have labeled some sample data manually and applied active learning to automatically label the rest of our dataset.

E.g. Location: city የአዲስ አበባ ዋና ከተማ ማን ነው? (Labeling)

However, it is recommended to manually label some of the automatically labeled samples if the questions are ambiguous. The challenge we faced when we try to automatically label the questions was dealing with some interrogative words that are used in different question categories. Finding the word "ምን" in Abbreviation Description and Entity categories can be a good example of why we might face issues during active learning for automatic labeling. As a result, we manually labeled the questions to be more accurate.

5.2 Tools and Experiment setup

We used the Python programming language for implementing the AQC prototype. Python accompanies many libraries for developing complex scientific, numeric applications and is designed with features to facilitate data analysis and visualization. We also used Keras to build and train our model utilizing Tensorflow as a backend. Numpy library helped us to compute the numerical operations of the neural network. The Unicodes in the trained word2vec model were encoded and decoded using the Codecs tool.

There are two ways to build models using Keras. One is sequential the other is functional. Sequential is not flexible which means it can not create a model that: shares layers, have multiple inputs have multiple outputs while the functional API is the opposite [37]. So we implemented our model using the functional API to maintain its flexibility.

Machine learning requires a developmental platform with good computational power. GPUs, CPUs, RAM as well as storage capacity. Fortunately, Googlecolab (colab.research.google.com/) provides these resources for free with an estimated usage time of GPU, i.e., allocation per user is restricted to a maximum of 12 hours at a time. The

GPUs available in Colab often include Nvidia K80s, T4s, P4s, and P100s. However, to choose the type of GPU, we are required to use the pro version. Using more resources like datasets in Colab, are more likely to run into usage limits and have their access to GPUs and TPUs temporarily restricted.

Our training has a minimum of 7- 14 epochs depending on the question category we train. For example, the “ENTITY”, “DESCRIPTION” classes go 13 iterations and we limit additional iterations to avoid overfitting. An iteration takes an average of 20 minutes. We save the models every time we train them.

5.3 Evaluation Metrics

We used a confusion matrix for measuring the AQC model. It is constructed by comparing the predicted class with the actual class [63].

Accuracy: The proportion of predictions that were correct. It is generally converted to a percentage where 100% is a perfect classifier. For a balanced dataset, an accuracy of where is the number of classes is a random classifier. An accuracy of 0% is a perfectly wrong classifier.

Hamming Loss: If we score 0, this indicates that we have a perfect classifier. A score is a random classifier for a balanced dataset, and 1.0 is a perfectly incorrect classifier.

Precision: This is the proportion of a class that was predicted to be in a given class and are actually in that class. A value of 1 means that the classifier was able to perfectly predict, for that class. A value of 0 means that the classifier was never correct, for that class.

Recall: This is the proportion of the True class predictions that were correctly predicted over the number of True predictions (correct or incorrect). A recall of 1 is a perfect recall, 0 is a “bad” recall. Figure 5.1 shows an example of how we used the Confusion matrix in our AQC model.

```
[[ 5  2  1  0  0  0  0]
 [ 0 51  5  0  0  1  0]
 [ 0  4 40  0  0  0  0]
 [ 0  1  1 13  0  0  0]
 [ 0  0  0  0  7  1  0]
 [ 0  2  0  0  0 31  0]
 [ 0  2  1  0  0  0  0]]
```

Figure 5.1: Confusion matrix- taken from the loss and accuracy of AQC model

Each row of the matrix represents the true label and each column of the matrix represents the prediction of the. The diagonal elements in the confusion matrix represent the number of accurate classifications for each class, while off-diagonal elements represent misclassifications classifier as we have presented it in Figure 5.3. The equations for accuracy and loss are shown below.

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Yi \cap Zi|}{|Yi \cup Zi|} \qquad \text{HammingLoss} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Yi \Delta Zi|}{|L|}$$

Where Δ stands for the symmetric difference of two sets.

Y- true label, Z- Predicted class, and L the full set of labels used in the dataset.

5.4 Training and Experiment Results

We trained the AQC model using 5000 labeled Amharic questions. Twenty percent of the training set was used to validate our model. We tested the model with 3000 test sets.1500 test sets were applied to evaluate the AQC model during training. While the rest 1500 test sets evaluate the trained AQC model.

When evaluating the trained AQC model, it should output the coarse and fine grain categories along with a confidence value. This shows the total number of questions, number of correct predictions for primary categories, number of correct predictions of secondary categories, and a confusion matrix for the primary categories.

The training process uses Adam optimizer to manage the weights when learning by referring to the loss function. The gap between the predicted and actual training data is the value assigned for the loss function. We use categorical_crossentropy to get the difference between the labeled and predicted value. This function is usually applied for multiclass classification models.

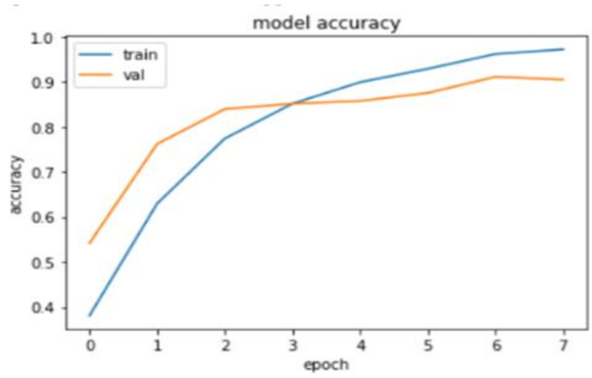

Figure 5.2: Accuracy and loss of AQC

Figure 5.5 shows the accuracy and loss score of the AQC on training as well as validation experiment. The evaluation of the coarse grain classification showed a promising result by scoring 90% accuracy on the validation and remarking a training loss of 13% with good training accuracy of 97%. The validation set is acquired by splitting 20% of the training to improve its accuracy by looking at the validation and test results of the AQC model during training.

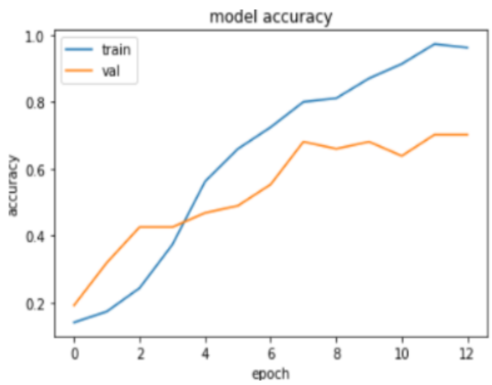

Figure 5.3: Training and Validation Result of Entity Fine-grain Category

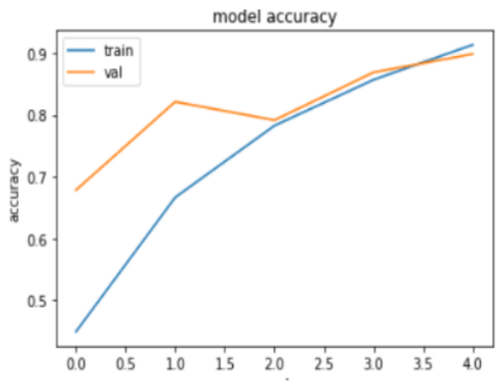

Figure 5.4:Training and Validation Result of Number Fine-grain Category

Figure 5.3 and 5.4 shows the performance of the AQC model while training and validating some of the fine-grain classes Entity and Number. Even though the AQC model has trained well both for Entity and Number fine grains, the validation for Entity fine-grain class showed the need for performance improvements by scoring 65% validation accuracy.

In this experiment, we also have tried to categorize Veracity type of questions, i.e., Yes/No and True/False question types. However, they happen to have a great loss of value as a result of their nature of similarity between each other (Yes/No or True/False).

Yes/No question types usually use question indicators like አይደል? ነው? አለ? የለም?. While True or False doesn't usually use question marks but the nature of them is like a statement type. For example: እዚህ አገር ለፓርኪንሰን በሽታ ህክምና የሚሰጥ ራሱን የቻለ ስልጠና አለ?

The above example can be seen in two ways. One is like a statement type that needs a true-false answer while avoiding the question mark and on the other hand considering the whole sentence we can come up with a Yes/No answer. So we can see that this is the reason why most taxonomies do not include these types of classes [20].

In this research, we tried to study some scenarios by changing the kernel and our Architecture.

Scenario 1: We changed the n-gram filters into bi-gram and tri-gram filters by removing the other two n-grams kernels. In the second training phase, we tried to remove the bi-gram and tri-gram filters by substituting them with the quad- as well as pent-grams. Training the system using both conditions displayed similar results as shown in Table 5.1.

Table 5.1: Scenario one experiment by changing filter size

Kernel filters	Training scores
Bi-gram and Trigram filters	Training_acc 0.92 Val_acc 0.86
Quad-gram and Pent-grams	Training_acc 0.91 Val_acc 0.86

Scenario 2: The AQC model uses four max-pooling layers of different sizes to take the maximum outputs of the activation function. In this experiment, we modified our architecture by merging the convolutional layers into the fully connected layer removing the max pool layer as shown in Figure 5.5

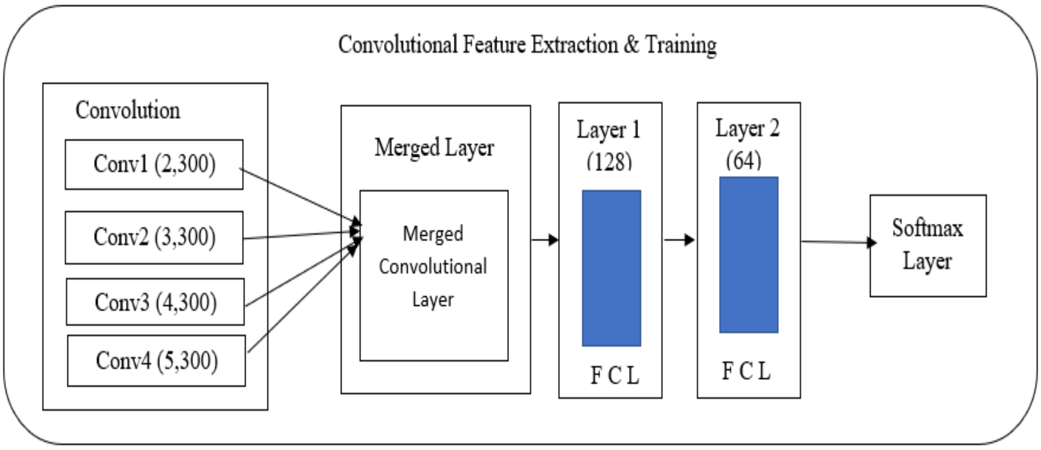

Figure 5.5: The AQC Training component without the max-pooling layer

Modifying the architecture without the max pool layer has increased the number of the training epochs and reduced the performance of the AQC model. It also requires the convolutional output (Feature maps) to be of the same size. By taking one convolutional layer, i.e., a bigram we achieved training and validation of 0.86 and 0.79 accuracies with 12 epochs.

Scenario 3: Simulation without the FCN.

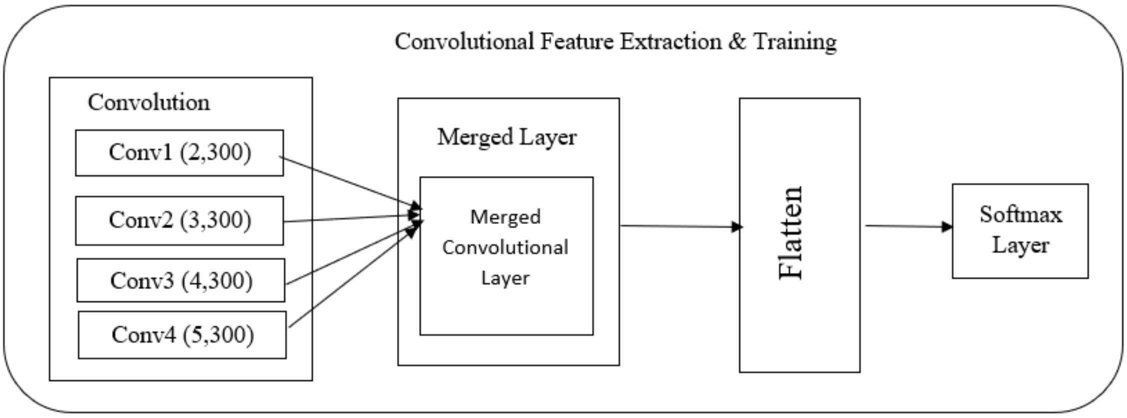

Figure 5.6: The AQC Training component without the FCL

The CNN without the Fully Connected Layers (FCLs) performed almost the same as the AQC model design which consists of two FCLs. This model performed a training accuracy of 97% with the validation accuracy of 88% with a five epoch cycle. This shows that the model is not affected even if we are not utilizing the dropout regularization function.

In general, the AQC CNN model performance has a competitive result with or without the FCLs. While the second scenario suggests a max-pooling layer for our CNN model.

5.5 Effects of Preprocessing on AQC model

To see the performance of DL for Amharic, we studied the Amharic language by studying the morphological structure, the phonemes, how it is combined with Amharic vowels (አናባቢ, i.e., አ ኡ ኢ ኧ ኤ እ ኦ), consonants, and affixes, what the preprocessing rules look like in verbs, nouns, adjectives, adverbs, to prepare both a lemmatized and stemmed data.

Normalization, Data cleaning, and tokenization are the basic components in the NLP preprocessing task. However, stop word removal, stemming, and root-level word extraction is dependent on the type of work the researcher is involved in.

AQC is mainly trained on stop words like ምን (What), ስንት(how many), የት (where), ማን (who), መቼ (When). We also use title names ወይዘሮ (Mrs), ወይዘሪት(Ms), and አቶ(Mr) to make it identify the taxonomy “HUM: title”. So we can’t use stop word removal not to lose the contents for training.

We stemmed and lemmatized the dataset to train the model with three different Amharic datasets: AQs (Surface form), Lemmatized AQs, and Stemmed AQs. Even though three of them have a difference in their form, the Stemmed ones are closer to the Surface AQs. As a result, they showed a relatively better result than the Lemmatized AQs.
Table 5.2 shows the three Amharic question datasets having similar interrogative words when even preprocessed. As a result, we tried to study if they can offer a promising result regardless of the difference we see. The model showed an outstanding performance for the AQs that are in a surface form. While the stemmed AQs AQs couldn’t progress more than 40% accuracy.

Table 5.2: Lemmatized and stemmed AQs

Interrogative words	Lemmatized Words	Stemmed Words
ለምን፣ ለምንድን፣በምን፣ምናቸው፣ ምንድን ፣ ስለምን	ምን	ምን
ለነማን፣የነማን፣በነማን፣በእነማን/ ከነማን፣ ማናማን	ም-ን	ማን
ትርጉሙ፣ትርጓሜው፣ ትርጓሜ፣ ትርጉም፣	ት-ር-ግ-ም	ትርጉም
ዘርዝሩ፣ ሲዘረዘር፣ መዘርዘር	ዝርዝር	ዘርዝር
ተንትን፣ተንትኚ፣ተንትኑ	ት-ን-ት-ን	ተንትን
የቱ፣ለየትኛው፣ከየትኛው፣የትኞቹ	ይ-ት	የት

Summary

In this Chapter, we have discussed how the AQ dataset was prepared and evaluated using the datasets. The dataset includes Amharic questions in Surface, Stemmed, and Lemmatized forms. We have experimented with these datasets and scored 90% accuracy for Amharic questions that are in surface form. The stemmed datasets couldn't progress with more than 40% accuracy. Here, we tried to see the performance of the preprocessed questions using a trained word2vec model on surface words, because these surface words were trained on 25,000 Amharic surface words which might include some of the common Amharic words even when preprocessed. If we see the Amharic interrogative words in Table 5.2, it shows how the preprocessed interrogative words are similar even when they are in a stemmed and lemmatized form. Thus we aimed at leveraging from having these common words in addition to the word2vec model, to put them in a vectorized form and train them.

This study takes three scenarios for training the AQC model. First, by using bi and trigram filters and by only having the quad and pent gram filters separately. Even though it has performed well by scoring 86% validation accuracy for both, the speed was comparatively slower than the model that extracts features using the four kernel filters at a time. Which also required more epochs. The second scenario showed that it is very important to have max-pooling layers to minimize the computational cost and increase the accuracy of feature extraction. The last scenario showed the performance of the feature extraction and training without having a fully connected layer.

We tried to compare our work with QC studied for Persian, Chinese, and Arabic languages using DL. QC for Persia [7] was developed using LTSM and had an accuracy of 81.77%. The performance of Chinese QC [12] was studied using the Attention-based CNN model which scored 79% accuracy. The author [64] stated that CNN responses in text classification have been rarely explored, as many approaches aim at explaining CNN classifications based on the visualization of image areas. Hence, we studied CNN using three scenarios by comparing their results as presented in section 5.3.

Chapter 6: Conclusion and Future Work

Question Classification analyses questions by comprehending their semantic and syntactic structure to categorize them into predefined taxonomies. QC is useful to retrieve precise answers to the system. They are applied in the Dialog system, Information Retrieval, Chatbots, Interactive game quizzes, and related fields that involve processing questions.

The main objective of this thesis work is, to design a generic QC for Amharic by using a multiclass question taxonomy. We also aimed to provide QC taxonomy following the QC standards used by researchers in addition to extracting the categories from the dataset we collected. We prepared the question datasets in three different forms. The Amharic Surface questions, Lemmatized and Stemmed forms, to explore the performance of AQC using a trained word2vec model for AQs. We were triggered to study this since most Amharic interrogative words are similar even when preprocessed.

To see the result of the AQC model to classify the generic Amharic questions into hierarchical (multiclass) categories. The model was learned from the 6 coarse grain (Abbreviation, Description, Human, Entity, Location, and Number) and 43 fine-grain question samples Abbreviation (Abbreviation and Expansion), Description (Description, Definition, Reason, List, Manner, Biography), Entity (Animal, Body, Color, Health, Event, Food, Instrument, Language, Letter, Movie, Plant, Product, Religion, Sport, Substance, Symbol, Technique, Term, Other), Location (City, Country, Place, Other) and Number (Age, Count, Date, Distance, Money, Percent, Time, Weight, Article). The AQC model was trained on 5000 labeled Amharic questions and validated with 3000 additional test samples. The performance of the AQC model acquired 90% and 97% percent validation and training accuracy. Having this result for Surface AQs, we tried to study the performance of Lemmatized and Stemmed AQs but the model tried to process only the stemmed ones with 40% accuracy.

6.1 Contribution

The main contributions of this thesis work include:

- ✓ We have adopted the tools and techniques used to classify questions for various languages to Amharic.

- ✓ We referred to question taxonomies designed by researchers and redefined a hierarchical taxonomy for Amharic questions and also prepared a labeled AQ dataset.
- ✓ The study showed the key components of the Amharic question classification and identified the language-dependent components.
- ✓ The AQC model can be a framework for factoid or non-factoid applications that involve question analysis as well as classification since we have developed a generic QC model from open domain datasets.
- ✓ The study showed the tools and techniques used for developing the AQC model by including the algorithms.
- ✓ We showed our experiment by using different combinations of n-gram filter kernels and changing the AQC architecture by removing the dense layer as well as the max-pooling layer.
- ✓ This study contributed by using CNN for NLP question classification by advocating the idea that CNN can also be used for natural language processing in addition to computer vision tasks.

6.2 Future work

Deep learning performs well when it has more data. We will consider adding more Amharic question datasets by exploring a better way of collecting datasets. We also aim to enhance the method we used to label our questions automatically.

This study presented the performance of our model using Stemmed and Lemmatized AQs. We in the future will train word embeddings using the preprocessed Amharic questions to compare them with our current result.

We developed the AQC model by grasping the syntactic and semantic structure of Amharic questions. We also have shown a glimpse of the AQC prototype by explaining the development phases, tools, and methodologies used for this purpose. Keras supported us in developing the AQC model with its APIs and libraries. Utilizing CNN, it is shown that this development tool can also be used for NLP tasks.

References

[1] Diksha Khurana, Aditya Koli, Kiran Khatter,and Sukhdev Singh, "Natural Language Processing: State of The Art, Current Trends and, Challenges," India, 2017.

[2] Makoto Nagao, "Natural Language Processing and Knowledge," in *Proceedings of the IEEE NLP-KE '05*, China, 2005.

[3] Wikipedia, "Dialogue system," [Online]. Available: https://en.wikipedia.org/wiki/Dialogue_ system. [Accessed 24 July 2019 at 17:09].

[4] "Search Engine Land," [Online]. Available: https://searchengineland.com/understanding-questionquery-answering-relates-website-digital-assets-207193. [Accessed 7 November 2019 at 9:28 am].

[5] Mohasseb A, "Question categorization and classification using grammar-based approach," 2018.

[6] Tilahun Abdisa, "Amharic Question Answering for Definitional, Biographical and Description Questions", Unpublished Masters Thesis, Department of Computer Science, Addis Ababa University, 2013.

[7] Mohammad Razzaghnoori, Hedieh Sajedi, and Iman Khani Jazani, "Question Classification in Persian using Word Vectors and Frequencies," *Cognitive Systems Research*, vol. 47, pp. 16-27, 2018.

[8] Arun D Panicker, Athira U, Sreesha Venkitakrishnan, Question Classification using Machine Learning Approaches, vol. 48 , International Journal of Computer Applications, June 2012.

[9] Medhanit Getachew, "Amharic Question Answering for Factoid and List Questions using Machine-learning Approaches", Unpublished Masters Thesis, Department of Computer Science,Addis Ababa University, 2019.

[10] Seid Muhie, "Amharic Question Answering System for Factoid Questions", Unpublished Masters Thesis, Department of Computer Science,Addis Ababa University, 2009.

[11] Zhengyu Fan, Lei Su, Xi Liu,, "Multi-Label Chinese Question Classification Based on Word2vec," in *the Proceedings of the IEEE 4th International Conference on Systems and Informatics*, 2017.

[12] Yihe Yang, Jin Liu and Yunlu Liaozheng, "Chinese Question Classification Based on Deep Learning," *Advanced Multimedia and Ubiquitous Engineering*, vol. 518, no. 2019, pp. 315-320.

[13] Alami Hamza, Noureddine En-Nahnahi, Khalid Alaoui Zidani, Said El Alaoui Ouatik, An Arabic Question Classification Method Based on New Taxonomy and Continuous Distributed Representation of Words, Elsevier B.V, 2019.

[14] Syed Mehedi, Hasan Nirob, Kazi Nayeem, and Saiful Islam, "Question Classification Using Support Vector Machine with Hybrid Feature Extraction Method," in *the Proceedings of the IEEE 20th International Conference of Computer and Information Technology (ICCIT)*, December 2017.

[15] Brook Eshetu, "Amharic Question Answering for list questions, A case of Ethiopian tourism", Unpublished Masters Thesis, Department of Computer Science, Addis Ababa University, 2013.

[16] Desalegn Abebaw Zeleke, "LETEYEQ (ሌጠየቅ) A Web-Based Amharic Question Answering System for Factoid Questions Using Machine Learning Approach", Unpublished Masters Thesis, Department of Computer Science.Addis Ababa University, 2013.

[17] Tilahun Yeshambel, "Amharic Document Image Retrieval using Linguistic Features", Unpublished Masters Thesis, Department of Computer Science, Addis Ababa University, 2011.

[18] Amit Mishra, Sanjay Kumar Jain, "A Survey on Question Answering Systems," *Journal of King Saud University,* vol. 28, no. 3, p. 345–361, 2016.

[19] Harish Tayyar Madabushi, Mark Lee, and John Barnden, "Integrating Question Classification and Deep Learning for improved Answer Selection," in *Proceedings of the 27th International Conference on Computational Linguistics,* Santa Fe, New Mexico, USA, 2018.

[20] Håkan Sundblad, Question Classification in Question Answering Systems, Sweden: Department of Computer and Information Science, 2007.

[21] Xin Li, Dan Roth, "Learning Question Classifiers," in *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002),* 2002.

[22] ባየ ይማም, የአማርኛ ሰዋሰው, ት.መ.ማ.ማ.ዴ, 1987.

[23] Michael Piotrowski, NLP-Supported Full-Text Retrieval., Master’s Thesis,: Friedrich Alexander University, 1998.

[24] Laude Loupy, Linguistic resources for Information Retrieval, ,France, 1998.

[25] Mohammed Terry-Jack, "Medium," [Online]. Available: https://medium.com/@b.terryjack/nlp-everything-about-word-embeddings-9ea21f51ccfe. [Accessed 22 April 2019].

[26] Wikipedia, "Word2vec," [Online]. Available: https://en.wikipedia.org/wiki/Word2vec. [Accessed January 21 2020].

[27] "FastText," [Online]. Available: https://fasttext.cc/. [Accessed 2020].

[28] Analytics Vidhya, "An Intuitive Understanding of Word Embeddings: From Count Vectors to Word2Vec," 2017. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/. [Accessed 04 June 2020].

[29] Medium, "Features in Machine Learning," Cogito Tech LLC, [Online]. Available: https://cogitotech.medium.com/what-are-features-in-machine-learning-and-why-it-is-important-e72f9905b54d. [Accessed 20 March 2020].

[30] CHOLLET FRANÇOIS, deeplearning with python, Manning Publications Co., 2018.

[31] Rao Anirudh, "edureka," Brain4ce Education Solutions, [Online]. Available: https://www.edureka.co/blog/convolutional-neural-network/. [Accessed 20 July 2020].

[32] Deep AI, " Logit," [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/logit. [Accessed 20 September 2020].

[33] Nikhil Buduma, Nicholas Lacascio, Fundamentals of Deep Learning, Boston: O'Reilly, 2017.

[34] Aurangzeb Khan, Baharum Baharudin, Lam Hong Lee, Khairullah khan, "A Review of Machine Learning Algorithms for Text-Documents Classification," *Journal of Advances in Information Technology,* Vols. Vol. 1, No. 1,, February 2010.

[35] Lan Courville, Goodfellow, Yoshua Bengio and Aaron, Deep Learning, MIT Press, 2016.

[36] Tim Dettmers, "Deep Learning in a Nutshell," [Online]. Available: https://developer.nvidia.com/blog/deep-learning-nutshell-core-concepts/#:~:text=A%20layer%20is%20the%20highest,output%20to%20the%20next%20layer.. [Accessed 3 November 2020].

[37] Pedro José Pereira Vieito, Convolutional Neural Networks for Efficient Object Detection on Ultra Low-Power Platforms, UNIVERSITAT POLITÈCNICA DE CATALUNYA, 2017.

[38] SAGAR SHARMA, "Towards Datascience," 2017. [Online]. Available: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6?gi=6798c351f27f.

[39] Amor Emma, "Machine Learning," Medium, 29 May 2020. [Online]. Available: https://medium.com/ml-cheat-sheet/understanding-non-linear-activation-functions-in-neural-networks-152f5e101eeb.

[40] IBM Cloud Education, "Convolutional Neural Networks," IBM, [Online]. Available: https://www.ibm.com/cloud/learn/convolutional-neural-networks. [Accessed 2020].

[41] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, Learning representations by backpropagating, 1986.

[42] AJAY SHRESTHA and AUSIF MAHMOOD,, Review of Deep Learning Algorithms, IEEE, 2019.

[43] Li Deng and Dong Yu, Deep Learning Methods and Applications, vol. Vol 7, Foundations and Trends in Signal Processing, 2014.

[44] Lars Schmarje, Monty Santarossa, Simon-Martin Schroder and Reinhard Koch, A Survey on Semi-, Self- and Unsupervised Learning for Image Classification,", 2020.

[45] Mohammad-Parsa Hosseini, Senbao Lu, Kavin Kamaraj, Alexander Slowikowski and Haygreev C. Venkatesh, "Deep Learning Architectures," in Deep Learning: Concepts and Architectures, Vols. vol. 866,, Switzerland: Springer Nature, 2020, pp. pp. 1-24..

[46] Jesper E. van Engelen and Holger H. Hoo, "A survey on semi-supervised learning," Machine Learning, vol. vol. 109, 2020, pp. pp. 373-440,.

[47] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad, Meysam Chenaghlu, Jianfeng Gao, Deep Learning Based Text Classification: A Comprehensive Review, ACM, 2020.

[48] Quara, "Feed-forward Neural Network," 2018. [Online]. Available: https://www.quora.com/Which-are-the-main-drawbacks-of-a-feed-forward-neural-network-composed-of-few-layers-with-thousands-of-neurons-each. [Accessed 14 November 2020].

[49] Cheng, Dong, and M. Lapata, Long short-term memory-networks for machine reading, : arXiv, 2016.

[50] Jason Brownle, "How Do Convolutional Layers Work in Deep Learning Neural Networks?," 2019. [Online]. Available: https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/. [Accessed 2020].

[51] Marco Baroni, Ngoc-Quan Pham and Gemma Boleda, Convolutional Neural Network Language Models, Ngoc-Quan Pham, 2016.

[52] LeCun, Li Bottou, Bengio, and Peter Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE, 1998, pp. 2278-2324.

[53] Jefkine, "Backpropagation In Convolutional Neural Networks," DESQUS, 2016. [Online]. Available: https://www.jefkine.com/general/2016/09/05/backpropagation-in-convolutional-neural-networks/. [Accessed 2018].

[54] Harsh Pokharna, "The best explanation of Convolutional Neural Networks on the Internet!," Medium, 2016. [Online]. Available:

https://medium.com/technologymadeeasy/the-best-explanation-of-convolutional-neural-networks-on-the-internet-fbb8b1ad5df8.

[55] DENNY BRITZ, "Understanding Convolutional Neural Networks for NLP," [Online]. Available: http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/#:~:text=In%20a%20traditional%20feedforward%20neural,we%20don't%20do%20that.&text=During%20the%20training%20phase%2C%20a,task%20you%20want%20to%20perform. [Accessed 10 April 2020].

[56] Deep Learning, 2016, 2016, p. Page 342.

[57] Alom, Md Zahangir, Taha, Tarek M., Yakopcic, Chris, Westberg, Stefan, Sidike, Paheding, Nasrin, Mst Shamima, Hasan, Mahmudul, Van Essen, Brian C., Awwal, Abdul A. S., Asari, Vijayan K, A State-of-the-Art Survey on Deep Learning Theory and Architectures, electronics MDPI, 2019.

[58] Jason Brownlee, "A Gentle Introduction to Pooling Layers for Convolutional Neural Networks," 2019. [Online]. Available: https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/.

[59] Araki, Marcin Skowron and Kenji, "Effectiveness of Combined Features for Machine Learning Based Question Classification," *Journal of natural language processing, The Association for Natural Language Processing,* Vols. 12, no. 6, 2005.

[60] XU-DONG LIN, HONG PENG, BO LIU, "Support Vector Machines for Text Categorization in Chinese Question Classification," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, China, 2006.

[61] Melese Tamiru, Automatic Amharic Text Summarization Using Latent Semantic Analysis, Unpublished MSc Thesis: Addis Ababa University, 2009.

[62] Krizhevsky, Sutskever, and Hinto, Imagenet classification with deep convolutional neural networks in Advances in Neural Information Processing Systems, Eds. Curran Associates, 2012, p. pp. 1097–1105.

[63] "Multiclass Classification," Oracle, [Online]. Available: https://docs.oracle.com/en-us/iaas/tools/ads-sdk/latest/user_guide/eval/Multiclass.html. [Accessed 2020].

[64] A Simple Convolutional Neural Network with Rule Extraction, applsci, 2019.

Annexes

Annex A: Coarse grain and Fine grain Amharic Question Dataset

Root/Lemmatized questions

ABBR:abb ሀ/መ ተ ዝ-ር-ዝ-ር አ ሲ ጽ-ሀ-ፍ እንዴት ነው?

ABBR:abb መ/ቤት ኦች ተ ዝ-ር-ዝ-ር አ ሲ ጽ-ሀ-ፍ እንዴት ነው?

ABBR:exp "ኢ.ፌ.ድ.ሪ." ተ ዝ-ር-ዝ-ር አ ሲ ጽ-ሀ-ፍ እንዴት ነው?

ABBR:exp መ/ደ/ፍ/ቤት ተ ዝ-ር-ዝ-ር አ ሲ ጽ-ሀ-ፍ እንዴት ነው?

DESC:biodesc ማሞ ውድነህ ማን ኸ ው?

DESC:biodesc ማርቲን ሉተር ኪንግ ማን ኸ ው?

DESC:def ሀይቅ ምን ድን ነው

DESC:manner ሃይል ወይም ኢነርጂ እንዴት ይ ፍ-ጥ-ር አል?

ENTY:body ፀጉር-ዎ በጣም በ ፍ-ጥ-ን ኧት የሚ ይ አ-ድ-ግ ኸ ው በ የት ኛ ው የ ሰው ነት ክፍል አ ላይ ነው?

ENTY:dismed ለ ት-ር-ፍ አንጀት ምን አይነት መድሃኒት ይ ስ-ጥ ኧል?

ENTY:event በ አለም ኣችን ላይ ለ አ-ጥ-ር ጊዜ የ ተ ካ ሀ-ድ ኸ ው ጦር ነት የቱ ነው?

ENTY: event በ አለም አስ ከ-ፍ ኢ የ ተ በ አ-ል ኸ አ ው-ል አ[አውሎ] ነፋስ የት ኛ ው ነው?

ENTY:food ለ እራት ወይም ከ መሸ በ ኋላ ለ መ ም-ግ-ብ የሚ ም-ክ-ር ኡ ም-ግ-ብ ኦች ምን ድ ን ን አቸው?

ENTY:food ለ እራት የሚ ም-ክ-ር ኡ ቅ-ል-ል የ አል ኡ ም-ግ-ብ ኦች የትኞቹ ን አቸው?

ENTY: food ለኦ የ ድ-ም አይነት የሚ ም-ክ-ር ኡ ም-ግ-ብ ኦች ምን ድ ን ን አቸው?

ENTY:food ለኦ የ ድ-ም አይነት የማ ይ ም-ክ-ር ኡት ም-ግ-ብ ኦች ምን ድ ን ን አቸው?

ENTY:lang በ ሰሜን አፍሪቃ ና በ ደቡብ ምዕራብ እስያ የ ሚ ን-ግ-ር ኡ ቋንቋ ዎች ቤተሰብ የ ሀ-ን ኸ ው የቱ ነው?

ENTY:lang በ ሲዮክስ ኦች የ ሚ ን-ግ-ር ቋንቋ ምን ድ ን ነው?

Stemmed Amharic questions

ABBR:abb ሀ/መ ተ ዘርዝር አ ሲ ጻፍ እንዴት ነው?

ABBR:abb መ/ቤት ኦች ተ ዘርዝር አ ሲ ጻፍ እንዴት ነው?

HUM:gr መኪና ን ማ ምረት በ እነማን ተጀመር ኸ?

HUM:gr ትልቅ ህንጻ ዎች ን መ ስር አት በ እነማን ተጀመር ኸ?

ENTY:word የ ወግ ሌላ ቃል ኡ ምን ድ ን ነው?

ENTY:word ሌላ ቃል ኡ ምን ድ ን ነው?

NUM:count በ 2012 የ አፍሪካ ዋንጫ በ አጠቃላይ ስንት ጎል ኦች ተቆጠር ኡ?

ENTY:event በ ሱርማ ብሄር ኧ ሰ ብ አዲስ የ ተ ወለድ ኧ ሕፃን ከተ ገረዝ ኧ በ ኋላ የ ሚ ከተል ኧ ው ሥርዓት ምን ድ ን ነው?

ENTY:event በ ቅንጦት አጓጊ ዎ ችኡ ካፕ ትራፍሌጋር እና በ ካርሜኒያ መካከል ምን ዓይነት የ ባህር ጦር ነት ተካሄደ?

ENTY:event በ ቴኒስ ውስጥ ትልቅ ኡ ክስት ምን ድ ነው?

ENTY:event በ ቻይና ውስጥ ትልቅ ኡ የ ውሃ ፕሮጀክት ምን አይነት ክስተት ን አስ ተናገድ ኧ?

ENTY:event በ አለም ኣችን ላይ ለ አጭር ጊዜ የ ተ ካ ሄድ ኧ ው ጦር ነት ምን ተ ብል አ ይ ታወቅ ኧል?

ENTY:food ሼፍ ኦች በ ብዝ አት የሚ ያ ዘጋጅ ኡት ምግብ ምን ድ ን ነው?

ENTY:food ከ 11 እጽዋት እና ቅመማቅመም ኦች ድብልቅ የ ሚ ዘጋጅ ኧ ው የ ምግብ አይነት ምን ድ ን ነው?

ENTY:food ከ ሰባት እስከ ዘጠኝ ወር አት ይ አል ኧ ህጻን ምን ሊ መገብ ይ ችል አል?

ENTY:food ጅብ ኦች ምን ይ መገብ ኣሉ?

ENTY:instru በተ ወጠር ኡ ክር ኦች የ ሚ ሰር አ የ ኢትዮጵያ የ ሙዚቃ መ ሳር ኢ ያ የቱ ነው?

ENTY:lang የ አዊ ዞን የ አፍ መ ፍች ኣ ቋንቋ ምን ድ ን ነው?

ENTY:lang ሐም አዊ የ የት ኛው የ ቋንቋ ቤተሰብ ንኡስ ክፍል ነው?

ENTY:lang በ ሰሜን አሜሪካ በ ሰፍ ኢ ው የ ሚ ጠቀም ኡ ብ አቸው አራት ቋንቋ ዎች ምን ድን አቸው?

ENTY:lang በ አሁን ኡ ወቅት በ አ ብዝ አኛ ው ኢትዮጵያ ው ያ ን የ ሚ ነገር ኡት ቋንቋ ዎች ምን እና ምን ን አቸው?

ENTY:lang በ አፍሪካ በ ብዝ አት የ ሚ ነገር ኧ ው ቋንቋ ምን ድ ነው?

ENTY:letter 26 ቱ የ ግዕዝ ፊደል ተነባቢ ድምጽ ኦች የ ሚ በ አል ኡት[የ ሚ ባል ኡት] እነ ማን ናቸው?

ENTY:letter ሁዋል ኧኛ የ አማርኛ አ ናባብ ኢ ድምጽ ኦች የ ሚ በ አል ኡት[የ ሚ ባል ኡት] የትኞቹ ን አቸው?

ENTY:letter ላይ ኛ የ አማርኛ አ ናባብ ኢ ድምጽ ኦች የ ሚ በ አል ኡት[የ ሚ ባል ኡት] የትኞቹ ን አቸው?

ENTY:letter ልከ ኛ የ አማርኛ አ ናባብ ኢ ድምጽ ኦች የ ሚ በ አል ኡት[የ ሚ ባል ኡት] የትኞቹ ን አቸው?

ENTY:letter በ ስታር ዋርስ ፊልም ለ መ ጀመር ኢ ያ ጊዜ የ ተ ነገር ኧ ው ቃል ምን ድ ን ነው?

ENTY:letter በ ሃይማኖት ሳብ ኢ ያ የ መጥ ኡ የ ግሪክ ቃል አት የትኞቹ ን አቸው?

ENTY:letter በ መተየብ ያ ቁልፍ ሰሌዳ ው ላይ ከኬ በ ስ ተ ቀኝ ምን ፊደል ይ አገኝ አል?

ENTY:letter በ ሳልቫዶር ዳሊ ፊርም አ ውስጥ ትልቅ ኡ ፊደል ምን ድ ነው?

ENTY:letter በ ስፔን በ ቀዝቃዝ ኣ ው የ ውሃ መታጠፍ ያ ላይ ምን ፊደል ይታያል?

ENTY:letter በ ታይፕራይተር ላይ ለመፅሐፍ በኤስ እና በ ኤፍ ፊደል መሀል ይ አለቅ ኧች ይ አለቅ ኧች ው ፊደል ማን ነች?

ENTY:letter በ አማርኛ ቋንቋ መሃል ኧኛ አ ናባብ ኢ ድምጽ ኦች የ ሚ በ አል ኡት[የ ሚ ባል ኡት] የትኞቹ ን አቸው?

ENTY:letter በ እንግሊዝ ኛ ውስጥ ሁለት ኛ ው አ ናባብ ኢ የ ሆን ኧ ው ፊደል የቱ ነው?

ENTY:letter በ ኮምፒውተር ኪቦርድ ኦች ላይ ለመፅሐፍ በ ኤች እና በኬ ፊደል መሀል ይ አለቅ ኧች ይ አለቅ ኧች ው ፊደል ማን ነች?

ENTY:movie ሂው ጃክማን አስማት ኧ ኛ ሆን ኦ በ ክርስቲያን ባሌ የ ተ ጫወት ኧ ው ገፀ ባህሪ በ የት ኛ ው ፊልም ውስጥ ነበር?

ENTY:movie ሊያ ከበደ ከ ሰር ኣ ችባ ቸው ፊልም ኦች መካከል 3ቱን ጥቀስ ኡ?

ENTY:plant በጣም በ ሰፍ ኢ ው የ ሚ መረት ኧ ው ተክል ምን ድ ነው?

ENTY:plant አንድ አንድ የ አውስትራሊያ ተ ወላጅ ዕፅዋት ምንድን አቸው?

ENTY:plant ከ ትንባሆ ጋር የ ተዛመድ ኧና ሲጋራ ዎች ከሚ ሠር ኡ በት የ ተክል አይነት ምን በ ሙብ አል ይ ታወቅ ኧል?

ENTY:plant የ ሊባኖስ ባንዲራ የ ምን ዓይነት ዛፍ ምስልን ይዝ ኡ ኣል?

ENTY:plant የ ናብራስካ ግዝ አት የሚ ታወቅ በት ዛፍ ምን ድ ን ነው?

ENTY:plant የ ወይን ጠጅ ከ ምን አይነት ተክል ነው የሚ ሰር ኧው?

ENTY:plant በ መጠን ኡ ትልቅ ዘር የ አል ኧ ው ተክል የት ኛ ው ነው?

ENTY:product ለ ፊት ጥር አት ከ ኬሚካል ነጽ ኣ የ ሆን ኧ ክሬም የቱ ነው?

ENTY:product ልብስ ኦች ን ለ መ ተኮስ የ ሚ ያ ገለግል ኡ ማሽን ኦች የትኞቹ ን አቸው?

ENTY:product ሙሉአለም አ ብዝ አ ኛ ው ን ጊዜ ምን ዓይነት ምርት ኦች ን ታስ ተዋውቅ ኣ ለች?

ENTY:product ሙሉአለም ና ሰራዊት አንድ ላይ በ መ ሆን ለ መ ጨረስ ኣ ጊዜ ያስ ተዋወቅ ኡት መ ዋ ብያ የቱ ነው?

ENTY:product ማክዶናልድ የ ተ በ አል ኧ ኮርፖሬሽን የ ሚ ት አወቅ ኧ ው በ ምን ምርት ኡ ነው?

ENTY:product ማዶና ያስ ተዋወቅ ኧች ው ምን አይነት የ ለስላስ ኣ መ ጠጥ ነው?

ENTY:product ምን አይነት ኦፕሬቲንግ ሲስተም ነው በ አ ብዝ አኛ ው ኮምፒውተር ኦች ላይ የ ተጫን ኧው?

ENTY:sport ላስቬጋስ ውስጥ በ ሴት ኦች በጣም ት አዋቅ ኢ የ ሆን ኧ የ ቁማር ጨዋት ኣ የቱ ነው?

ENTY:sport ላስቬጋስ ውስጥ በጣም ት አዋቅ ኢ የ ሆን ኧ የ ቁማር ጨዋት ኣ የቱ ነው?

ENTY:sport በ 1891 ጀምስ ናዚዝ ምን አይነት የ ስፖርት ጨዋት ኣ ፈጠር ኧ?

ENTY:sport በ16ኛ ሳምንት የ ኢትዮጵያ ፕርሚር ሊግ ሀትሪክ የ ሠር ኡ ተ ጫዋት ኦች እነ ማን ን አቸው?

ENTY:sport በ ምን ዓይነት ስፖርት ነው ክሪስ ጆጊስ ት አዋቅ ኢ የ ሆን ኧው?

ENTY:sport በ ስፖርት ማ ሰልጠኻ ኣ ዎች ምን አይነት ስፖርት ይ ዘወተር ኣ ል?

ENTY:sport በ ትልቅ መስክ ላይ የ ሚ ደረግ ስፖርት አዊ ጨዋት ኣ ምን ድ ን ነው?

ENTY:substance ሁለት ኛ ው ቀላል ንጥረ ነገር ምን ድ ን ነው?

ENTY:substance ሁለት ኛ ው በጣም ጠንካር ኣ ው ንጥረ ነገር ምን ድ ነው?

ENTY:substance ለ አሉምንየም ና ለ ፎቶ ስር አ የ ሚ ጠቅም ኧ ው ንጥረ ነገር ምን ድ ን ነው?

ENTY:substance ለ አ ንፀባራቅ ኢ ነገር ኦች የ ሚ ውል ኡ ኬሚካል ኦች ምንድን አቸው?

ENTY:substance ለ ኤሌክትሮላይሲስ የ ሚ ይ አስ ፈልግ ኡ ግብ አት ኦች ምን ድ ን ን አቸው?

ENTY:substance ለ ጤና ማ አጥንት እና ጥርስ የ ሚ መከር ኧ ው ንጥረ ነገር ምን ድ ን ነው?

ENTY:substance ለ ጤና ማ አጥንት ኦች እና ጥርስ ኦች ምን አይነት ንጥረ ነገር ኦች ይ ወሰድ ኣሉ?

ENTY:substance ማርብል ከ ምን ተ ሠር ኣ?

ENTY:substance ምራቅ ምን ን ያካተታል?

ENTY:letter በ የት ኛ ው ፊደል ነው የ ሩዋንዳ ባንዲራ ያ ጌጥ ኧው?

Amharic Surface Questions

ENTY:symbol ምሕፃረ ቃልን ለመፃፍ የምንጠቀመው ሥርዓተ ነጥብ ምልክት የትኛው ነው?

ENTY:symbol ምዕራፍና ቁጥርን ለመለየት የሚጠቅመው ምልክት የቱ ነው?

ENTY:symbol ሰያፍ መስመር በመባል የሚታወቀው ምልክት የቱ ነው?

ENTY:symbol ስም ጥሪን ወይም ተናጋሪ ገፀ - ባሕሪያትን ለማመልከት የሚጠቅመው ምልክት የቱ ነው?

ENTY:symbol ቃላትን በምህፃረ‚ቃል አሳጥሮ ለመጻፍ የሚያገለግለው ስራተ ነጥብ ምንድን ነው?

ENTY:symbol በተከታታይ የሚመጡ ንዑስ ሐረጎችን ለመለየት የምንጠቀመው ምልክት የቱ ነው?

ENTY:symbol በንግግር ላይ የተንጠለጠልን ሐረግ ከሌላናው ለመለየት የምንጠቀመው ምልክት የቱ ነው?

ENTY:symbol በአረፍተ ነገር መነሻ ላይ ሲገኝ ፣ አረፍተ ነገሩ ተቀባይነት የሌለው መሆኑን የሚያመለክተው ምልክት የቱ ነው?

ENTY:symbol በዘመናዊ የመቀምር ጽሑፍ ላይ ቦታ የሌለው ምልክት የቱ ነው?

ENTY:symbol በድምጽ ደረጃ ንበታዊ አጻጻፍን በሃአረግ ደረጃ መዋቅራዊ አንድነትን የሚያሳይ ልዩ ምልክት የቱ ነው?

ENTY:symbol በግጥም የቤት መምቻ ስንኞች መጨረሻ የሚገባው ምልክት የቱ ነው?

ENTY:symbol ብር ወይም ሲልቨር የተባለው ኢለመንት ምልክቱ ምንድን ነው?

ENTY:techmeth በሐይቅ ውስጥ ያሉትን ዓሳዎች ለመቁጠር ምን አይነት ተመራጭ ዘዴዎች አሉ?

ENTY:techmeth በምድጃ ውስጥ እሳትን ለመፍጠር ምን አይነት ዘዴዎች አሉ?

ENTY:techmeth በአማርኛ ስኀ-ድምጽ የሚፈጠርበት መንገድ ምንድን ነው?

ENTY:techmeth ቦርጭን ማጥፊያ ቀላል መንገድ ምንድን ነው?

ENTY:techmeth እንቁላል በአርተፊሻል መንገድ የሚፈጠርበት ዘዴ ምንድነው?

ENTY:techmeth ከፈርመንቴሽን ውጪ ምን አይነት ተመራጭ ዘዴዎች አሉ?

ENTY:term ሀሣብን የመግለጽና አመለካከት የመያዝ ነፃነት ምን ይባላል?

ENTY:term ሰውን የሚጠላ ሰው ምን ተብሎ ይጠራል?

ENTY:term ስለኮከቦች የሚያጠና ሳይንስ ምን ይባላል?

ENTY:word ረጅሙ የእንግሊዝኛ ቃል ምንድነው?

ENTY:word በመጽሐፍ ቅዱስ ውስጥ 46227 ጊዜ የተደጋገመው ቃል ምንድን ነው?

ENTY:word በእንግሊዝኛ ቋንቋ ውስጥ በጣም ታዋቂው የጀርመን ቃል ምንድነው?

ENTY:word በፍቅር እስከመቃብር ድርሰት ላይ ተደጋግሞ የሚገኘው ቃል ምንድን ነው?

ENTY:word ብዙ ፊደሎች ያሉት የእንግሊዝኛ ቃል ምንድን ነው?

ENTY:word ብዙግዜ ምንጠቀማቸው ቃሎች ምን ምን ናቸው?

ENTY:word ተውላጠ ስም የሚባሉት የትኞቹ ቃላት ናቸው?

HUM:gr የአዲስ አበባ ከተማ አስተዳደር ም/ቤት አባላት በነማን ይመረጣሉ?

HUM:ind ከኪም ባሳሪን ጋር በመሆን በ 9½ ሳምንቶች ውስጥ ፊልም የሰራው የትኛው አሜሪካዊ ኮከብ ተዋናይ ነው?

HUM:ind አጼ ዮሀንስ በማን እጅ ተቀብተው ነገሱ?

HUM:ind አጼ ገላውዲወስን የገደለው ማን ነው?

HUM:ind አጼ ገላውዲዎስ የማን ልጅ ናቸው?

HUM:org በኢፌዲሪ ህገ መንግስት መሰረት ከፍተኛ የስልጣን አካል ተጠሪነቱ ለማን ነው?

HUM:title ማእረግነቱ ከፊታውራሪ ዝቅ ብሎ የሚገኘ ምን ይባላል?

LOC:country 23ኛው የአፍሪካ ህብረት የመሪዎች ጉባኤ የት ሀገር ተካሄደ?

LOC:city የኪንታኪ የፈረስ ፓርክ ለየትኛው ከተማ ይቀርባል?

LOC:country ላፖዝ የየትኛው አገር ዋና ከተማ ነው?

NUM:age ዳግማዊ አጼ ኢያሱ ስልጣን ላይ የወጣው በስንት አመቱ ነው?

NUM:age ጀርመናዊው ኪሊንስማን በስንት አመቱ ነው ከፍተኛ ግቦችን ያስቆጠረው?

NUM:age ጠ/ሚ መለስ በረሃ የገቡት በስንት አመታቸው ነው?

NUM:article ስለ ልዩ ጨረታ ትርጉም የተጠቀሰው አዋጅ ቁጥር ስንት ላይ ነው?

NUM:article ስለ አካል ደህንነት ሙብት የሚናገረው ስንተኛው አንቀፅ ነው?

NUM:article ስለ አዋጅ ትርጉም የተጠቀሰው አዋጅ ቁጥር ስንት ላይ ነው?

NUM:article ስለ ጥንታዊ ቅርስ የተደነገገው አዋጅ ቁጥር ስንት ላይ ነው?

NUM:count ለመሆኑ ይህ ጦርነት ለምን ያህል ጊዜ የተካሄደ ነበር?

NUM:count ለስንተኛ ጊዜ እየተከበረ ይገኛል?

NUM:count ለብሔራዊ ፈተና ስንት የመፈተኛ ጣቢያዎች አሉ?

NUM:count ሊዮናርዶ ዲካፕሪዮ ምን ያህል የአካዳሚ ሽልማቶች አሸንፏል?

NUM:count ሊፋን ሞተርስ ምን ያህል አገር በቀል ግብረሰናይ ድርጅቶችን ለመርዳት ወሰነ?

NUM:date በንጉሱ ዘመን የኢትዮጵያ ሕዝብ መዝሙር መቼ ተጻፈ?

NUM:date በአ/አበባ ለመጀመሪያ ጊዜ የተከፈተው ፍርማሲ በስንት ዓ.ም ነበር?

NUM:money አንበሳ አውቶቡስ በአሁኑ ሰአት ምን ያህል ገቢ አለው?

NUM:money አንድ ሎቲ ስንት ዶላር ነው?

NUM:money አንድ ናይራ ስንት ዶላር ነው?

NUM:perc በኢ.ኤም.ኤስ የስራ ሂደት በ2008 ዓ.ም 1ኛ ሩብ ዓመት የተከፈለ የካሳ ክፍያ ስንት መቶኛ ይሆናል?

NUM:perc በእንቁላላ ጣይ ዶሮዎች መኖ ቀመር ውስጥ የኑግ ፋጉሎ እስከ ስንት ፐርሰንት ቢቀላቀል ውጤታማ ያደርጋል?
NUM:perc በውቅያኖስ ውስጥ ምን ያህል ፐርሰንት የሚሆነው ዕፅዋት ነው?
NUM:perc በዘመናዊ ዶሮ እርባታ ስራ ውስጥ ምን ያህል ፐርሰንት የሚሆነው ለእርባታ ወጪ ይወጣል?
NUM:perc በዝርግዝና ላይ፣ በወሊድ ወቅትና ጡት በማጥባት ግዜ የኤችአይቪ በሽታው ከእናትየው ወደ ህጻኑ የመተላለፍ አቅሙ ስንት ፐርሰንት ነው?
NUM:perc በዶሮዎች መኖ ውስጥ የዳቦ ስንዴ ስንት መቶኛ ይሆናል?
NUM:time አትሌት መሰረት በ36ኛው የቤጂንግ ማራቶን ውድድሩን በስንት ሰአት አጠናቀቀች?
NUM:time አትሌት መሰረት ደፋር የሴቶች የአምስት ሺህ ሜትር የቤት ውስጥ ውድድር በምን ያክል ሰዓት አሸነፈች?
NUM:time አንድ ሰአት ስንት ደቂቃ ነው?
NUM:time አንድ ትልቅ የስጋ ቁራጭ ለመብላት ከፈለጉ ከ ስንት እስከ ስንት ሰዓታት ይወስዳል?
NUM:time አንድ ደቂቃ ስንት ሰከንድ ነው?

Annex B: Amharic Coarse and Fine Grain Interrogative Words

Table 6.1: Coarse and fine grain with their interrogative and indicative terms

No.	Coarse grain	Fine_grain	Class indicative term	Interrogative term
1.	Abbreviation (አህጽሮተ-ቃል)	Abbreviation (አህጽሮተ ቃል)	አህጽሮተ ቃል፣ አህጽሮተ ቃሉ፣ በአጭሩ ሲጻፍ፣ አጭር ቃል፣	ምን - ምንድን እንዴት
2.		Expansion (ዝርዝር)	ተዘርዝሮ ሲጻፍ፣ሲዘረዘር ምንን ይገልጻል?	ምን - ምንን እንዴት
3.	Description (ግለጽ)	Definition (ትርጉም)	ትርጉሙ፣ ትርጓሜው፣ ትርጓሜ፣ ትርጉም፣ ሲተረጎም፣ ስንተረጉመው፣ ፍቺ፣ ሲፈታ፣ ስንፈታው፣ and ፍቺው	ምን - ምንድን
4.		Description (ግለጽ)	ጠቀሜታ፣ ድርሻ፣ ጥቅም፣ ፋይዳ፣ አገልግሎት፣ ሚና፣ ተግባር፣ አብራራ፣ አብራሪ፣ አብራሩ፣ ግለፅ፣ ግለጭ፣ ግለፁ፣ ፋይዳው፣ አገልግሎቱ፣ ሚናው፣ ተግባሩ፣ ድርሻው፣ ጥቅሟ፣ ፋይዳዋ፣ አገሌግልቷ፣ ሚናዋ፣ ድርሻዋ፣ ተግባሯ፣ ጥቅማቸው፣ ፋይዳቸው፣ አገልግሎታቸው፣ ሚናቸው፣ ድርሻቸው፣ and ተግባራቸው	ምን - ምንድን፣ ለምን አብራራ- አብራሩ/አብራሪ ግለጽ- ግለጭ/ግለጹ፣ አስረዳ ተንትን፣ተንትኚ፣ተንትኑ
5.		Manner (ሁኔታ)	መንገዶች	እንዴት ነው? ምን
6.		Reason (ምክንያት)	ምክንያት፣ ምክንያቶች፣ ምክንያቶቹ፣ እንደምክንያት፣ እንደ ምክንያትነት	ለምን - ለምንድን፣በምን ምክንያት ምን- ምንድን ፣ ምን ምን፣ ስለምን፣ እንደምን፣ ምንድን ነው?
7.		Biography (ባዮግራፊ)		ማን - ማነው፣ ማናት፣ and ማናቸው
8.	Entity	Animal (እንስሳት)	ተብሎ ይጠራል፣ስም፣ይባላል?	የቱ - የትኛው,የበየትኛው፣ እነማን፣ ምን - ከምን፣ምንድናቸው፣ ምንድን ነው?፣ማን፣
9.		Body (የሰውነት ክፍል)	የሰውነት ክፍል፣	ምን- ምንህን, ምንሽን, ምናቸው,ምንድናቸው? የት-ከየትኛው, የትኞቹ, የትኛውን,የየትኛው,በየትኞቹ
10.		Color (ቀለማት)	ቀለማት	ምን , በምን

11.	Health(ጤና)	የጤናችግር፣ምልክቶችን፣በሽታ፣ መድሃኒት፣ኪኒን፣ማከም	ምን - ምንድን ናቸው፣ ምንድን ነው፣ ምን ምን፣ በምን፣ለምን,ምን ያህል,በምንድን,የምን የቱ፣ለየትኛው
12.	Event (ክስተት)	ክስተት,አጋጣሚ፣በዓል	ምን - ምንድን፣ምንድነው፣ የትኛው፣በየትኛው፣የቱ
13.	Food (ምግብ)	ምግብ ፣ይመገቡ፣የሚበላው	ምን - ምንድን ነው?፣ከምን በምን አይነት፣ የቱ የትኛው፣በየትኛው፣
14.	Instrument (መሳሪያ)		ምን - በምን ምንድን የቱ፣የትኛው፣ከየትኞቹ፣ የትኛውን
15.	Language (ቋንቋ)		ምን - ምንድነው የት -የትኛው፣ የትኞቹ፣ የቱ፣ ከየትኛው፣የየትኛው፣ እነማን
16.	Letter (ሆሄ)		የት - የቱ፣በየትኛው፣የትኞቹ፣ የትኞቹን ፣ከየትኛው፣ ምን- ምንድን፣ምንድነው፣ በምን፣ምንድናቸው፣ እነማን፣ እነማንን
17.	Movie (ፊልም)		ምን - ምንድነው የት - የቱ፣በየትኛው፣የትኞቹ
18.	Plant (እጽዋት)		ምን - ምንድን,ምንድናቸው, ምንድነው, የት-የትኞቹ, የቱ, የትኛው,
19.	Product (ምርት)		የት- ከየትኛው, የትኞቹ, የየትኛው, በየትኞቹ ፣የቶቹ ምን- ከምን፣ምንድን ነው፣
20.	Religion (እምነት ሃይማኖት)		ምን - በምን፣ ፣ ምንድን፣ እነማን፣ የቱ፣ የትኞቹ
21.	Sport (ስፖርት)		ምን - በምን ፣ ምንድነው የቱ፣ የትኞቹ,የትኛው፣
22.	Substance (ንጥረነገር)		የት - ከየትኛው, የትኞቹ, የየትኛው, በየትኞቹ ፣ ምን - ምንን ምንድናቸው፣ ምንድን ነው? ከምን፣
23.	Symbol (ምልክት)	ምልክት	የት- የቱ,

24.		Technique (ዘዴ)	ዘዴ	ምን - በምን፣ምንድን፣ተናገር፣ የቱ፣የትኛው፣ በየቱ
25.		Term (ቃል)	ቃል	ምን
26.		Other		
27.	Human (ሰው)	Title (ማእረግ)	ማእረጉ፣ ማእረጓ፣ማእረጋቸው፣ ማእረጋችሁ፣ማእረጋቹ፣ ስራ፣በመስራት፣ይሰሩ፣በሙያው፣ ሙያው፣ሙያዋ/ው፣በሙያዋ፣ በሙያቸው፣የስራ ዘርፍ፣ ሲሰሩ፣ ሹመት፣ዝምድና/ህ/ሽ/ቹ/ችሁ	ምን - ምንህን፣ምንሽ፣ምንህ፣ ምንሽን፣ ምናቸው፣ ምንድን ነው?፣ በምን፣
28.		Group (ቡድን)	አካላት፣ቡድን ቡድኖችን፣ቡድኖች	ምን-ማንኛቸው፣እነማን፣ እነማንን፣የትኛው፣ የቶቹ፣የቱ፣ማን - ለነማን፣ የነማን፣በነማን፣በእነማን/ ከነማን,ማናማን፣
29.		Organization (ተቋም)	ተቋማት፣አካል፣ተቋም	የት-ከየትኛው፣የየትኛው፣ በየትኞቹ፣የቱ፣የትኛውን፣የት የትኛው ፣ ለየትኛው፣ በምን፣ ለማን፣ለማን፣ምን፣ማን
30.		Individual (ግለሰብ)		ማን- ማን, ለማን, ማነው, ማንን, የማን, ከማንኛው, ማንኛው, ማንኛይቱ, ማንኛዋ, በማን, ከማን፣ የትኛውን፣ የትኛው፣ማናት፣ የማን
31.	Location ()	City (ከተማ)	ከተሞች፣ከተማ ርዕሰ መዲና	የት-በየት፣ወዴት፣ከየት፣ የቱ፣ እስከየት፣ ወደየት፣የየትኛው፣ በየትኞቹ፣በየትኛው፣የትኛው፣ የየት፣የትኛው፣የትኞቹ፣ ማን-ማናት?ማነች
32.		Country (ሃገር)	ሃገር፣ሃገራት፣አገር፣አገሮች	እነማን፣የት፣በየት፣ወዴት፣ ከየት፣የቱ፣እስከየት፣ወደየት፣ የቷ፣ የየትኛው፣በየትኞቹ፣ ከየትኛው፣**የየት**፣የትኛው፣ የትኞቹ፣እነማን፣ማናት በማን፣የማን
33.		Place (ቦታ)	ቦታ፣አካባቢ፣ ዋና ዋና፣ቦታዎች፣	ምንድን ነው፣ የት- ከየት፣ወዴት፣በየትኛው፣ የየት ፣ የትኛዋ፣በየት፣የየት ፣ እስከየት ምን ከየት ወደ የት,ከየት፣የትኞቹ,

				ምን ተብሎ ይጠራል,ቦታ ምን ይባላል,የቦታ ስም ምንድን
34.	Numeric	Age (እድሜ)	ዓመት፣አመቱ፣ዐመቷ፣ ዐመታቸው፣	ስንት -ከስንት፣ስንት፣በስንት፣ በስንተኛው
35.		Count (ቁጥር)	ብዛት፣ኣመታትን፣ቀናት፣ ቁጥራቸው፣ቁጥር	ስንት- ስንቱ, ከስንት፣በስንት፣ በስንተኛው፣ በየስንት፣ስንቴ ስንቱ, ከስንት, በስንት, ለስንት,እስከ ስንት፣በየስንት በስንተኛው ምን - ምን ያህል, ለምን ያክል ፣ከምን ያህል፣በምን ያህል
36.		Date (ቀን)	ቀን፣ዓ.ም.፣ክፍለ ዘመን	መቸ, መቼ, በመቸ, እስከመቸ፣ እስከመቼ, ለመቸ, ለመቼ፣ የመቸ,እስከ መቼ,በስንት፣ በሰ̈ንተኛው፣ ከስንተኛው፣ ዓ.ም.,መች መች፣ ከመቼ እስከ መቼ፣በስንት ዓመት ምህረት እስከ ስንት፣የትኛው፣ከስንት እስከ ስንት፣በምን ቀን ፣ምን፣ የትኞቹ፣የቱ፣በየትኛው
37.		Distance (ርቀት)	ርቀት ፣ ሜትር፣	ምን ያህል፣በምን ያህል፣ስንት፣ በስንት
38.		Money (ገንዘብ)	ወጪ፣ዋጋ፣ገቢ፣ክፍያን፣ ካፒታል፣ ገንዘብ፣ሲተመን፣ፈጀ፣ ይከፈለዋል?በጀት ፣ወለድ፣ቀረጥ	ስንት፣ከስንት፣በምን ያህል፣ ስንት፣በስንት, የምን ያህል
39.		Percent (ፐርሰንት)	ፐርሰንት፣ ክፍል፣ክፍሉ	ምን ያህል፣ስንት በ መቶ፣ ስንት መቶኛ፣ስንት፣ከስንት ስንት በስንተኛ
40.		Time (ጊዜ)	ሰአት,ሰዓት,የሰአት,ደቂቃ, በደቂቃ,ጊዜ,ደቂቃዎች, ሰአታት	በምን ያህል፣ስንቱ መቼ በመቼ እስከመቼ የመቸ ለመቸ (Date) ስንቱ፣ስንት፣ከስንት፣በስንት፣ ለስንት፣ምንያህል
41.		Weight (ክብደት)	ክብደት	ስንት
42.		Article (አንቀጽ)	አንቀጽ ቁጥር፣ አዋጅ	በስንተኛው፣ስንተኛው፣ ስንተኛ፣የቱ፣በየትኞቹ፣የትኞቹ ፣በየትኛው፣የትኛው
43.	Veracity	Yes/No	These two vary by the question mark and statement type structure respectively.	ወይ, እንዴ, ይሆን, ነው፣ አለው? ያስፈልገዋል? እንዴ?
44.		True/False		ናቸው፣ አሏቸው፣ ይሆናል፣ ነው፣ አለው? ያሳያል።፣

Declaration

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all sources of materials used for the thesis have been duly acknowledged.

Declared by:

Name: ____

Signature: ____

Date: ____

Confirmed by advisor:

Name: ____

Signature: ____

Date: ____