2023-02

AUTOMATIC QUESTION GENERATION FROM AMHARIC SENTENCES USING A RULE-BASED APPROAC

MESRET, BEYENE ASCHALE

BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

FACULTY OF COMPUTING

MSc Thesis:

AUTOMATIC QUESTION GENERATION FROM AMHARIC SENTENCES USING A RULE-BASED APPROACH

By:

MESRET BEYENE ASCHALE

February 2023

Bahir Dar, Ethiopia

BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

FACULTYOF COMPUTING

AUTOMATIC QUESTION GENERATION FROM AMHARIC SENTENCES USING A RULE-BASED APPROACH

By:

MESRET BEYENE ASCHALE

A Thesis submitted

In partial fulfillment of the requirements for the Degree of

Master ofScience in Computer Science

Advisor: Adane Nega (Ph.D.)

February 2023

Bahir Dar, Ethiopia

BAHIR DAR UNIVERSITY
BAHIR DAR INSTITUTE OF TECHNOLOGY
SCHOOL OF GRADUATE STUDIES
FACULTY OF COMPUTING
Approval of thesis for defense result

I hereby confirm that the changes required by the examiners have been carried out and incorporated in the final thesis.

Name of Student Meseret Beyene Aschale Signature ________ Date 22/10/2015 E.C

As members of the board of examiners, we examined this thesis entitled "**Automatic Question Generation From Amharic Sentences Using A Rule-Based Approach**" by Meseret Beyene. We hereby certify that the thesis is accepted for fulfilling the requirements for the award of the degree of Masters of Science in "**Computer Science**".

Board of Examiners

Name of Advisor	Signature	Date
Dr. Adane Nega	________	22/10/2015 E.C
Name of External examiner	**Signature**	**Date**
Dr. Martha Yifir	________	22/10/2015 E.C
Name of Internal Examiner	**Signature**	**Date**
Mr. Belete Biazen	________	22/10/2015 E.C
Name of Chairperson	**Signature**	**Date**
Mr. Alemu Kumilachew	________	22/10/15 E.C
Name of Chair Holder	**Signature**	**Date**
Ms. Kidest Meshesha	________	22/10/2015 E.C
Name of Faculty Dean	**Signature**	**Date**
Asegahegn E	________	22/10/2015 E.C

Faculty Stamp

Bahir Dar University Bahir Dar Institute of Technology Faculty of Computing

Acknowledgment

First, I would like to thank the Almighty God for blessing and helping me all of my life and in good progress this study. The next deep gratitude shall be given to Dr. Adane Nega my thesis advisor. His strict follow-up, constructive ideas and comments worth great for my current study and gave me a nice experience for the future. Without his scheduled follow-up, this thesis was not have been well done by this time. I would like to thank Mr.BelaynehAlamir (Linguistic Expert) who supported me in his linguisticexperience on Amharic Discourse language basic concept. And also I would like to thank my parents.

List of Figures

List of Tables

Abbreviation

AQG ………………Automatic Question Generation

AI ………………..Artificial Intelligence

NLG ………………Natural Language Generation

NLU……………...Natural Language Understanding

NLP ……………….Natural Language processing

QA ……………....Question Answering

QG ………………. Question Generation

MCQ …………….. Multiple Choice Questions

NER………………Named Entity Recognition

NLTK……………...Natural language Tool kit

POS……………….Part Of Speech

AQA ………………Automatic Question Answering

IR ………………. Information Retrieval

Pip ………………… package installer Python

ORG ………………Organization

PER ………………Person

GPE ……………… Geopolitical Entity

ART……………….Artifact

EVE ………………Event

NAT ……………....Natural Phenomenon

Abstract

Question generation is part of Natural Language Processing. Students today have numerous challenges when preparing for exams. Professors and Teachers spend a lot of time and effort to make exams. The Automatic Question Generation Model proposes a solution to save time, effort, and the student's learning process, which helps in self-calibration for educational purposes. In this thesis work, we present an automatic question generation system for the Amharic language.AQG aims to develop a system that takes sentences of text and produces a good-quality question based on the text, such that the answer to the question can be worked out from the base sentences. The proposed system proceeds by transforming declarative sentences into interrogative sentences, based on preliminary named entity recognition of the base sentence.

AQG generates questions automatically by using its model, which is generated using a rule-based approach.Question generation generates shallow questions that focus more on facts such as who/‹‹ማን››, what/‹‹ምን››, when/‹‹መቼ››, where/‹‹የት››, and why/‹‹ለምን››.We used Python programming language on Jupyter notebook Anaconda navigator which isa web-based interactive computing notebook environment withappropriate libraries.

Our system is rule-based, runs on sentence-based parse information of a single-sentence input, and achieves high accuracy in terms of syntactic correctness and fluency. The AQG model uses pre-trained data from the open repository GitHub as a training dataset, which helps to get more accurate results.

As a result, the question Generation systems rely on manual Evaluation. However, the proposed system was evaluated based on the quality of the output system and the linguistic well-formed type of criteria to evaluate the Result. Based on syntactic correctness 82.78% was generated and 88.78% was fluency generated questions. We also present an evaluation of the output system result, which shows that Recall was 82.35%,precisionwas 70.00% and F-measure was 75.67%. Which is good according to it generated automatically using rule-based approaches.

Keywords: *Automatic Question Generation, Named entity Recognition*

Contents

CHAPTER ONE

1. INTRODUCTION

1.1. Background

Question generation is part of Natural Language Processing (NLP). It is an area of research where many researchers have presented their work and is still an area of research to achieve higher accuracy in the English language and other languages. The number of non-English web pages is rapidly expanding, and there is a need for people all over the world to be able to use their language when accessing the information on the web. This requires the existence of a variety of natural language processing applications, including local language spell-checkers, word processors, machine translation systems, search engines, etc. A little over half of the home pages of the most visited websites on the World Wide Web are in English.

In the Amharic language, very few computational linguistic resources have been developed and very little has been done in terms of making useful higher-level Internet or computer-based applications available to those who only speak Amharic (Gambäck& Asker, 2014). Natural languages processing applications such as information retrieval, text classification, or document filtering could benefit from the existence and availability of basic tools such as stemmers, morphological analyzers, or part-of-speech taggers.

Questions are used to check the evidence from such types of documents or contents and also extract information from the existing documents or contents. Therefore, questions are the basic constraint on learning and knowing. Automatic question generation is the process of generating reasonable questions from a given input sentence.

Natural Language Generation is the task of translating machine-readable, non-linguistic information into an equivalent human language representation. Question generation is an emerging research area of artificial intelligence in education (Ming and Li, 2017). Automatic question generation can help individuals generate questions from the given sentences automatically. It is a process in which given input sentences to the scheme, it creates reasonable questions from the input as an output. Question Generation from Text is a Natural Language Generation task concerned with generating questions from unstructured sentences. As such, it uses tools from both natural language understanding and natural language generation. This thesis deals with the implementation of automated question generation from the raw text in Amharic. We also proposed a system to generate the questions automatically from Amharic sentences by using rule-based approach techniques.

The potential benefits of using automated systems to generate questions help reduce the dependency on humans to generate questions and other needs associated with systems interacting with natural language (Ishita et al., 2016). Recently, question generation and question answering in the field of computational linguistics have attracted enormous attention from researchers (Rus and Grasser. 2009).

Many researchers have worked in the area of automatic question generation through NLP, and numerous techniques and models have been developed to automatically generate the different types of questions in the English language. Work has also been done in many other non-Amharic languages.

Hence, we propose and develop a wh-word approach to question generation that combines sentence-level Amharic language semantic analysis with semantically motivated question templates. We would enable good templates to be produced.

There has been little research done to demonstrate the effectiveness of the automatic question generation model and design for Amharic (Getaneh, 2021), (Fikir, 2022).NLP is important because it's an approach that helps the researcher, which improves our communication and influence skills at a time when these are becoming even more important. NLP also helps us develop our logical, emotional, and intuitive intelligence, which are particularly useful for surviving and thriving in the world today.

Generating such questions, and authoring reading assessments more generally, can be a time-consuming and effortful process. In this thesis, we work toward automating that process. In particular, we focus on the problem of automatically generating Wh-word type questions from Amharic language sentences. We aim to create a system for question generation (QG) that can take simple sentences (e.g., a lower-grade textbook) and build as an output a list of Wh-word type Amharic questions. The automatic generation of questions from sentences consists of automatically transforming a declarative sentence into an interrogative sentence. To construct a question from a given sentence, the system should first process the sentence to extract basic text elements. Sentence processing is a subject of the NLP fields of computer science in such systems. This is a complex task, which mobilizes several resources and tools in NLP, such as named entity recognition and sentence simplification in predefined tokenized Amharic sentences (Adelani,seid.,.et al., 2021).

Automatic question generation aims to develop a system that takes sentences of text and produces a good-quality question based on the text such that the answer to the question can be worked out from the base sentences.

1.2. Statement of the problem

Nowadays, teachers, professors, and tutors (academicians) spend a lot of time manually generating test papers and quizzes. Similarly, students spend a lot of time on self-analysis and self-calibration. Moreover, students are dependent on their mentors for self-analysis in the English language and many other languages.

Most of the question generation research has focused on the English language. However, Amharic Question Generation has been little research done through different tools and technique (Getaneh, 2021), (Fikir, 2022).

AQG is done by Getaneh (2021), who demonstrated the automatically generated questions could be as effective as human generation.

A high-quality corpus is essential when designing NLP systems. The POS and NER taggers will perform better as the corpus gets larger. As a result, researchers should develop a high-quality corpus for their research area that is larger in size (Getaneh, 2021).

Design an automatic question generation model for Amharic legal text documents using deep learning is depends on questions and answer pair's dataset.The model had three modules: a preprocessing module, a module for model building, and a feature mapping trained model with an Amharic question-answer set (Fikir, 2022).

The problem is that they are not effective for large-scale data and on their study does not include mapping more than one question from one sentence at a time Getaneh (2021). Deep learning is effective for a specific Amharic question-answer set (Fikir, 2022).
However our approach is automatic question generation and the answer set is generated or mapping from the base sentences.

The automatic question generation from Amharic sentences consists of automatically transforming a declarative sentence into an interrogative sentence. This is a complex task, which mobilizes several resources and tools in NLP, such as named entity recognition and sentence simplification.
The automatic generation of questions is generally associated with a simplification step, which may be upstream of the generation. (Michael & Noah A., 2010) describe an algorithm for extracting propositions independent of complex sentences. To do this, the algorithm is based on the structure of the sentence (relative and subordinate clauses, adverbs, and appositions) to break it up and remove the less significant parts.
Automatic question generation systems can help learners to find their knowledge gaps by helping them to formulate their valuable queries.

Automatic question generation based on the discourse connectives was proposed to use the method of content selection and question formation to form types of questions like Why, when, where, in which, in the English language, and Evaluation and Result are shown by manually evaluating for semantic and syntactic soundness of questions by two evaluators (Manish Agarwal, R. S., 2011).

Automatic question generation using natural language processing techniques was proposed to use a methodology for rule-based automatic question generation in the English language (Onur, K., 2018).

Additionally, automatic question generation in multimedia-based learning was proposed, and the proposed methodology was a question generation for documentary video, definitions, selection of system-based generated post-question, generation of human QGs, selection of pre-question, and selection of images, participants and interface, and procedure. And the question type was who, whom, where, whose, when, and what with the

English language whose evaluation and results were A large-scale experiment investigating the productivity of generating questions (time taken to post-edit questions vs. time taken to generate questions from scratch) is planned (Yvonne, S., et al., 2012).

In this paper, we develop a system that is used to generate questions from Amharic sentences using a rule-based approach with in larger corpus.
Our specific approaches would add to the literature on Amharic sentence QG techniques. This thesis would answer the following basic questions:

- What are the challenges in developing an automatic question generation for the Amharic language?
- How to model automatic question generation from Amharic Text.
- How do we evaluate the performance of the automatic question generation system?

1.3. Objective

1.3.1. General objective

The general objective of this thesis is to model an Automatic Question Generation system for the Amharic language.

1.3.2. Specific Objective

In Amharic Automatic Question Generation, the specific objectives of the study are as follows:

- To review of related literature
- To modelan automatic question generation for the Amharic language
- To generate a valid and fluent question according to a given sentence.
- To evaluate the performance of the proposed model
- Advancing recommendations for future work in the area of Question Generation.

1.4. Scope and Limitation of the study

In our study, we limited ourselves to Wh-type questionsquestions. This thesis would focus only on automatic question generation from Amharic sentences using a rule-based approach, particularly for lower-grade textbooks. Furthermore, AQG constructs Where/የት/, who/ማን/, what/ምን/, when/መቼ/, why/ለምን type of wh-Question. In addition, our system does not deal with questions of quantity (how much?) and measurement (which size? which length?) to improve the system performance.

1.5. Significance of the Study

This thesis mainly improves Amharic languagethrough natural language processing. The possible relief of using automated systems to generate questions helps decrease the dependency on users to generate questions and other needs associated with schemes cooperating with natural languages (Swali& Shah, 2016).

Our proposed approach to automatically generating questions has an important impact on the broad area of intelligent authoring tools, which are needed to help teachers, learners, and other users save their time and effort in creating pedagogical content and to assist educational technology developers to reduce development costs (Ming &Vasile, April-June 2017). Also, the other important of these facilities are:

- Recommend good questions that learners or users might ask while reading Amharic documents.
- Suggest questions for self-evaluation and evaluate others.
- Help teachers to reduce the amount of work and time spent on generating questions.
- It helps other researchers to study this interesting researchable area.

1.6. Methodology of the Study

We used an experimental methodological approach for the proposed solutions and a rule-based approach for the analysis. Using this methodology, different experimental setups were implemented and evaluated for their effects on the proposed thesis work.

1.6.1. Literature Review

A literature review is a key methodology used to investigate and find out the state of the art in related works and identify gaps. In this methodology, a variety of NLP techniques and algorithms will be analyzed related to question generation.Secondary data sources, like electronically available books, articles, publications, and other previously written resources related to the topic were referred to so that there could be more understanding of this particular subject matter. Reviewing and criticizing related works aids in the acquisition of knowledge in the field as well as skills in the selection and application of appropriate algorithms and tools.

1.6.2. Software Tools

The relevant tools for the developing automatic question generation system were developed using Anaconda, which is a distribution of the Python programming languages for this type of study and other scientific computing(machine learning applications, large-scale data processing on NLP, predictive analytics, and so on) that aims to simplify package management and deployment.

The distribution includes data-science packages suitable for Windows. The Jupyter notebook programming language is also a tool in Anaconda Navigation that is used to store the learned rules and in building user-friendly interfaces that facilitate how users interact with the system.

The proposed system requires a web browser such as Opera, Chrome, or the default Windows Microsoft Edge, and so on, as a runtime environment. The system was developed and tested on an HP laptop with a Core i5 6th edition and 8 GB of RAM. The system is stable and uses the basic graphic adapter well. The computer was configured and powered by Windows 10 64-bit.

1.6.3. Method of Evaluation

The proposed system was tested to evaluate its performance. To evaluate the performance of the output question generation system, we have used standard confusion metrics, namely accuracy, precision, and recall. To this end, a prototype of the system has been developed.

1.7. Thesis organization

This paper begins by explaining experience in the development of Amharic question generation-related work. It also describes possible improvements in the applicability of potential scopes. This thesis is organized into several chapters. In Chapter 2, we review related work and explain an approach toward automatic question generation using sentences and in this chapter, we list the previous works in this area and also compare our work with them. Chapter 3 explains the methodology and Design of question generation. It explains in detail how the domain influences the quality of appropriate tools and techniques. Chapter 4 explains the experimental setup and evaluation of automatic question generation. The remaining chapters contain the thesis's conclusions and recommendations for future work.

2. LITERATURE REVIEW

2.1. INTRODUCTION

This section concentrates and reviewson addressing related to Automatic Question generation system design, developmentstrategies, and issues. Recently, Question Generation (QG) and Question Answering (QA) in the field of computational linguistics have got enormous attention from researchers (Rus and, Graesser, 2009).

2.2. Question Generation

Question Generation (QG) is the task of automatically generating questions from various input texts such as raw text, database, or semantic representation. One of the most significant uses of questions is reflection, which helps us better understand what we have learned. People frequently spend hours alone studying concepts and pondering challenges brought about by what they have read. These concepts and issues are frequently expressed as questions. From the earliest stages of learning to original research, questions are used. A question is often the starting point of an investigation in the scientific process, and it can be thought of as a transition between the observation and hypothesis stages (Chal, Y. l., &Hasan, S. A., 2012).

In Amharic Question and answering system some part of language processing is related to Automatic Question Generation. The question processing module accepts the user's question and performs tasks such as question type determination, question focus (important terms about the question) identification, and expected answer type determination (Seid, M., & Mulugeta, L.,2008). The question type was determined based on the question particles and the question focuses. Since most of the question particles in Amharic are multipurpose, the question focus plays a greater role in determining the question type. They have developed a question typology that is used to determine the expected answer type. The question processing modules also generate the proper IR query that is submitted to the document retrieval component of AQA.

Natural Language Processing includes automatic question production. Many researchers have presented in this field, particularly in the English language. Questions are used to verify information from current content or to collect data from existing content. As a result, questions are an essential component of learning and self-evaluation. The process of generating reasonable questions from a given text is known as question generation.

These systems save a lot of time because manually creating questions takes a long time. Deep question generation and shallow question generation are the two basic types of question generation, based on the goal complexity. Shallow QG generates shallow questions that focus more on facts (such as who, what, when, where, which, how many/much, and yes/no questions), but deep QG generates deep questions that involve more logical reasoning (such as why, why not, what-if, what-if-not, and how questions) (Jaspreet& Ashok, 2015).

In this phase (Swali et al., 2016) they divided the simple sentences into subsections of English sentences i.e. Subject, Verbs, and Objects. Then Named Entity Recognizer is processed over the Subject and Object of the sentence to identify the coarse class classification of it. The NER then specifies the tagged type of the words as Person/human, Location, and Organization. The coarse class classification is as follows:

For examples:

Input: Sachin plays cricket at 5 am.

Output: Who plays cricket?

Who plays cricket at 5 am?

What does Sachin play?

In the same way, in every language, questions are constructed with the help of question particles (interrogative words) and question marks (?) which are placed at the end of the question (Seid, M., & Mulugeta, L.,2008). Table 1 shows some of the Amharic question particles.

Question word	Romanization	Description
ማን	man	Who related questions
ለማን	leman	to whom …
ማነው	manew	Who is ….
የት	*yet*	Where …
ስንት	*Sint*	How many ….
ለምን	*Lemin*	Why…

Table 1: Amharic Question Particles

2.3. The Amharic Language

Currently, on the web Amharic document is increasing as many newspaper agencies provide their service electronically. The traditional Information retrieval techniques were considered insufficient in retrieving precise information for the user. While information retrieval is effective by itself, users these days demand a better tool.

Amharic is a Semitic language spoken in many parts of Ethiopia and the official working language and also official working language for the regions within the federal system, including Amhara and the multi-ethnic Southern Nations, Nationalities, and Peoples Region (Seid & Mulugeta, 2019).

The Ethiopian Orthodox Church uses the Ge'ez alphabet to write Amharic, which is a unique script. Ge'ez has been written since at least the 4th century A.D. Unlike Arabic or Hebrew, the language is *written* from left to right. The basic symbols in later versions of the writing system primarily represent consonant-vowel (CV) phoneme pairs, whereas the basic characters in earlier versions primarily represented consonant-vowel (CV) phoneme pairs. Gambäck and Asker (2014) note that the script features its own set of punctuation marks and numbers, as well as some special characters for labialized consonants

The Amharic language has been declared to have word categories as ስም (noun), ግስ (verb), ቅፅል (adjective), ተውሳከግስ (Adverb), መስተዋድድ (preposition), and ተውላጠስም (pronoun). (Medhanit, G., 2019)

Affixes, prefixes, and suffixes, especially suffixes, are used to denote gender, number, definiteness, case, and direct object status in Amharic nouns (Medhanit, G., 2019). Gender can be masculine or feminine in Amharic nouns. Amharic nouns may have a masculine or feminine gender. Suffixes are added to denote a masculine or feminine noun gender. Some nouns may have both masculine and feminine gender, while other nouns may only have one gender. The feminine gender is used to indicate female as well as smallness. For example, ቤቴትንሽነች፡፡Plurals are formed by adding ዎችor ኦችwhether the word ends with a vowel or consonant.

Verbs are words derived from roots and affixes to inflect person, number, gender, mood, voice, and polarity. Verbs agree with their subjects. Verb agreement with objects is optional. Verbs in Amharic are mostly placed at the end of the sentence.

Adverb: it can be used to qualify a verb by adding an extra idea to the sentence. The Amharic adverbs are limited in number and include ትላንት፣ዛሬ፣ገና፣ቶሎ፣ etc.

Adjective: any word that modifies a noun or an adverb, which comes before a noun ጎበዝተማሪ፣በጣምጎበዝ. Another specific property of adjectives is, when pluralized, it repeats the previous letter of the last letter of the word (e.g. ትንሽ፡ትንንሽ).

Preposition: a word that can be placed before a noun and perform adverbial operations related to place, time, cause, and so on; which can’t accept any suffix or prefix; and which is never used to create a new word. It includes ስለ፣ወደ፣እንደ፣ከ…

Pronoun: this category further can be divided as a deictic specifier, which includes እሱ፣እኔ፣አንተ፣አንቺ፣እነሱ…; quantitative specifier, which includes ጥቂት፣አንዳንዶ፣አንዶ…; and possession specifier such as የአንተ፣የኔ፣የሱ…etc

2.4. Amharic Morphology

Amharic is one of the most morphologically complicated languages, just like other Semitic languages. Nouns in Amharic are inflected for gender, possessive/genitive case, accusative/objective case, number, and definiteness. Adjectives in Amharic can be marked for number, definiteness, cases, and gender in a manner similar to how nouns are inflected. Except for certain plural constructions, the affixation of morphemes to convey numbers is comparable to that of nouns. Contrarily, Amharic verbs can take any combination of person, gender, number, case, tense/aspect, and mood inflections. As a result, a single verbal root can produce tens of thousands of distinct verbs (in surface forms). As in the example yisebrenal ('he will break me'), a single verb can make up a complete phrase because they are designated for several grammatical units (Mesfin & Yaregal, 2014). It is determined how to analyze this verb (sentence).

For another ways are states to analysis a language. A lattice rescoring framework has been used to apply a number of language models (statistical, linguistic, morpheme-based, and FLMs) to an Amharic speech recognition challenge. The 100 best choices for each test sentence were compiled into lattices, which were then graded again using different language models. The use of factored language models has been found to significantly enhance WRA(Martha, Solomon,etal., 2009).

2.5. Morphological Analysis

The morphological analysis module's training phase is essential to the system's successful implementation. The memory-based learning approach is used to implement the morphological analysis. Therefore, in this phase, morpheme identification is used to classify and extrapolate the class of new instances, feature extraction is used to make the input words suitable for memory-based learning classification, stem extraction reconstructs and inserts identified morphemes, and root extraction is used to obtain root forms and stems with their grammatical functions (Mesfin & Yaregal, 2014).

2.6. Rule-based Question Generation

Researchers start with a rule-based method that extracts key aspects from the input text and then inserts these aspects into human-generated templates for interrogative sentence generation (Heilman, 2011). Question Generation has been studied for some time now and, from 2008 to 2011, there was a huge contribution to its research mostly due to a QG workshop (Rus and Lester, 2009, Rus et al., 2010, 2011, 2012) taking place, with the last two containing a Shared Task Evaluation Campaign. Syntactic parsers are also widely used. They allow the mapping of sentences into trees, by grouping words and the associated tokens into nodes representing their syntactic tags. These tags, such as noun phrases (NP) and prepositional phrases (PP) identify different targets for the question generation process, with their textual value being the answer to those questions.

Named entity recognition is used to help choose the correct Wh-word to use for the question. For example, the sentences Bob hit Ana and The car hit Ana have the same syntactic tree, but the chosen Wh-keyword must be different, depending on the subject of the sentence, if one wants to generate a question of the type <Wh-word>.

They developed a semi-automated method using natural language processing techniques to generate grammatical test items. Their approach implied handcraft patterns to find authentic sentences and distractors from the web that transform into grammar-based test items (Chen et al., 2006).

2.7. Approach Question Generationin Different Language

This section reviews various approaches to Automatic Question Generation of the Wh-word type of question generation as listed in the above section and related to it.
Questions beginning with a Wh-phrase, i.e. what, who, which, why, when, where, etc. are factoid questions. How is a Wh-phrase by default?These words are capitalized here when they are used in a technical, rather than generic, sense (Ambuja & Manisha, 2017).

Questions about conditional context are the ones that have stems containing phrases like if, then, so, etc.

Example 1: Sentence: If it rains, the picnic will be cancelled.

Question: What would happen if it rains?

(2) Questions about temporal information are the ones whose stems contain date and time expressions.

Example 2: Sentence: Tilak obtained Bachelor of Arts in first class in Mathematics from Deccan College of Pune in 1877.

Question: When did Tilak obtain Bachelor of Arts in first class in Mathematics from Deccan College of Pune?

(1) Stems of questions on possibility and necessity contain in their expressions the words would, will, should, could, must, may.

Example 3: Sentence: You should read daily two hours for your exam.

Question: Why should you read daily?

It has been noted that factoids with the first two words who or whom are simple to construct because (1) they are based on a subject, the entity person, and (2) the order of the sentences stays the same when forming a question. Due to the fact that they are based on the position of the entity, questions containing where-phrases are simple to predict and develop (Ambuja & Manisha, 2017).

2.7.1. Automatic question generation for Swedish: The current state

Question generation has its background in AI research, and a new community of academics interested in QG, particularly concerning the English language, has formed, culminating in a regular international workshop series. The QG problem is typically defined as the task of exhaustively producing all questions to which a text may be said to provide answers. (This definition 1 is taken from Rus and Graesser's 2009 proceedings, which can be seen at http://questiongeneration.org/.) Of course, whether such a complete collection can be found for any text is a matter of controversy. QG for arbitrary text is a type of NLP application in which the contribution of basic NLP methodologies is held to a different standard. Swedish is abbreviated as QG.

In the previously indicated information extraction scenario, the goal of QG was to allow a user to only ask questions that had been prepared, ensuring that responses would appear in the text. One clear difficulty for the user was locating the query – or, more accurately, a formulation of a question – amid the frequently large number of questions created. Wilhelmson (2010, 2011) proposed a seemingly counterintuitive method for assisting the user in locating a question: by adding reformulations to the 'set of questions,' he was able to extend the 'set of questions' even further. Early tests also revealed that adding alternative question formulations without word-sense disambiguation by substituting Swedish near-synonyms from Folkets synonymordlista (Kann and Rosell, 2005) and Swedish WorldNet (Viberg, Lindmark, and Lindvall, 2002) resulted in a large number of incorrect questions.

2.7.2. Towards Automatic Topical Question Generation English Language

In order to assess the value of the generated questions, they employ LDA (Blei et al., 2003) to choose significant sub-topics from a given body of texts. The LDA method is a probabilistic approach to topic modeling. The basic concept is that each document should be viewed as a collection of diverse subjects.

Table 2: Semantic roles with possible question words question words

Arguments	Question Words
ARG0...ARG5	*Who, Where, What, Which*
ARGM-ADV	*In what circumstances*
ARGM-CAU	*why*
ARGM-DIS *AGRM-EXT*	*how* *To what extent*
ARGM-LOC	*Where*
ARGM-MNR	*how*
ARGM-PNC	*why*

ARGM-TMP	*when*

A probability distribution across words is used to represent each subject. The bag-of-words assumption (Misra et al., 2008) states that documents are made up of words and that the order in which they are written does not matter. When producing a new document, the basic principle is to choose a topic distribution. For each word in the new text, a topic is chosen at random from this distribution, and a word is pulled from that topic.

2.7.3. Automation of Question Generation from Sentences English Language

The Question Generation from a single input sentence with a target question type (e.g. who? where? when?) was tested. For this, they used development data from the Question Generation Shared Task Evaluation Challenge 2010.QGSTEC. Sentences for the QGSTEC were drawn from primary data sources (Wikipedia, Yahoo! Answers, and OpenLearn), with sentences that are unusually long or short being avoided. Because complicated sentences may have a complex structure with multiple clauses, it would be difficult to produce accurate questions from them (Rakshit, s., 2012). As a result, we simplify the procedure by exploiting syntactic information to extract elementary sentences from complicated statements.

We categorize sentences based on the subject, verb, object, and preposition to identify the sort of questions that were created. The basic structure of our Question Generation system is shown in Figure 2. Elementary Sentence Construction this module extracts the elementary sentences from the complex input sentences by syntactically parsing each complex sentence (Rakshit, s., 2012).

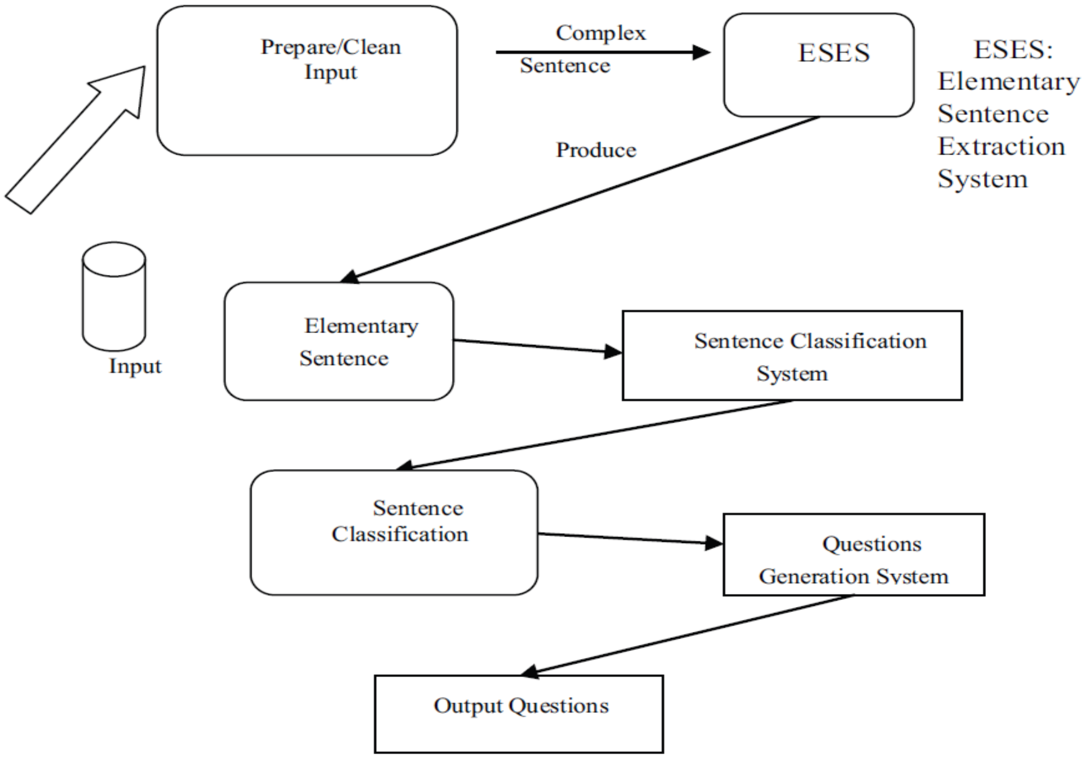

Figure 1: Overview diagram for the QG system [Rakshit, s., 2012]

2.7.4. Automatic Chinese Factual Question Generation

Each sentence taken from an article is parsed using the LTP program. In this step, our system performs word segmentation, part of speech tagging, named entity recognition and dependency parsing, as well as semantic role label parsing.

In the first stage, the researcher simplifies the sentences to reduce the complexity of the inquiry generation process. Sentence splitting and compression are required because Chinese sentences are often very long and commonly tie two or more self-complete sentences together. For summarization, researchers focus on preserving the critical information of the sentence (Zhang, M. Hu, T. Xiao, et al., 2013). They can simplify the source sentence by removing parentheses (The parts in a sentence that operate as explanatory or qualifying remarks and have no apparent dependent links with the other constituents of a phrase.) and adverbial modifiers.

Question transformation a key subtask of question generation is target content selection, i.e., what is the target content the question is asking about? In their case, they identify answer phrases in the input declarative sentence as potential targets for generating questions. In Chinese, a question is generated by using an interrogative pronoun to replace the target answer phrase in the declarative sentence. Unlike question generation in English, it does not require subject-auxiliary inversion and verb decomposition. In this respect, the question-generation process in Chinese is simpler.

2.7.5. Amharic Automatic Question Generation

AQG is done by Getaneh (2021), who demonstrated the automatically generated questions could be as effective as human generation.The system's performance was assessed at many stages because one stage's performance influences the following stage's performance. The system's overall performance has been assessed after various system components have been examined. The computed accuracy of the POS tagger, NER, and informative sentence selection based.

For their study the accuracy of POS tagger comes up with 84.6%. The size of training dataset and type of data has an effect on the result since the training data does not cover large amount of historic data. As they have seen in the second experiment, the training data contains more historical data which helps to improve the performance of the tagger. The POS tagger performance has an effect on NER system, since the NER developed by adding, replacing, or changing part of speech. The experimental results of NER got the accuracy of 82.0%. NER has an effect on informative sentence selection. For example: from the sentence <<በጨቅላዕድሜውጥሩየገናተጫዋችነበር>> “ጥሩ” is wrongly tagged as person (per) name but is an adjective (good). The informative sentence selection methods select the sentence as informative sentence wrongly. As a result, the questions generated from the sentence become wrong. In Amharic some person names are adjectives or vice versa. For example, from the sentence <<ቴዎድሮስመልካምተዋጊነበር፡፡>> “መልካም” tagged as person (per) incorrectly as a result, the generated questions becomes incorrect. For example, instead of the question <<ማንመልካምተዋጊነበር?>> it becomes <<ማንተዋጊነበር?>> incorrectly.

From their experiment, some question types showed good results but other question types showed low results. The accuracy of questions generated by question phrase “ስንት” has got high score; question phrase “የት” has got lower score. To construct grammatically and semantically correct questions, use a sentence with named entity location that may or may not take the question phrase "የት". For example: <<አበበቢቂላ 1925 ዓ.ምደነባከምትባልስፍራተወለደ::>> from the sentence, question can be constructed based on their named entity location (“ደነባ”) such as <<አበበቢቂላ 1925 ዓ.ምየትከምትባልስፍራተወለደ ?>> this sentence has grammatical error. Instead of “የት”, using “ማን” question phrase can create grammatically correct sentence.

According to Getaneh (2021), the study shows that a good coverage of domain specific training dataset which helps to identify named entities in the sentence and defining more rules which includes more word classes has a great impact on automatic Amharic question generation system.

A high-quality corpus is essential when designing NLP systems. The POS and NER taggers will perform better as the corpus gets larger. As a result, researchers should develop a high-quality corpus for their research area that is larger in size (Getaneh, 2021).

Design an automatic question generation model for Amharic legal text documents using deep learning is depends on questions and answer pair’s dataset. Using the dataset and hyper-parameters that were built up in three experiments for each mode the trained model predicts the predefined class and finally generate questions within answer pair.The model had three modules: a preprocessing module, a module for model building, and a feature mapping trained model with an Amharic question-answer set (Fikir, 2022).

The problem is that they are not effective for large-scale data and on their study does not include mapping more than one question from one sentence at a time Getaneh (2021). Deep learning is effective for a specific Amharic question-answer set (Fikir, 2022).
However our approach is automatic question generation and the answer set is generated or mapping from the base sentences.

2.8. Overview of Related Work

The General overviews of Question Generation Related Work are stated below in the tabular form. The objective of this review is to automatic question generation systems and find why automatic question generation is still interesting for researchers.The literature review shows that most of the researchers paid attention to generating objective-type questions, automatically or semi-automatically. A limited number of approaches are found in the literature that shows interest in question generation.Questions rely heavily on informative sentences. A quality question is generated by an informative sentence. We found that text sentence simplification and some rule-based techniques in the literature exacted the informative sentences from an input text. The majority of the previous articles did not devote enough attention to the step of selecting informative sentences. But it is a useful step for generating quality questions. Generate simple sentences from complex and compound sentences are also complex.

General overviews are stated in table [2] below:

Table 3: General Overview of Related Work

no	Algorithm	Methodology	Type of Question	Language	Evaluation and Result
1	Automatic question generation using natural language processing techniques(Onur , July 2018)	the rule-based automatic question generation system		English	
2	Amharic Question Answering for Biography, Definition, and Description Questions(Tilahun & Mulugeta, 2019)	Document retrieval and answer extraction and a machine learning tool have been used for question classification and biography detection.		Amharic	
3	Automatic question generation based on the discourse connectives (Manish Agarwal, 2011)	Content selection and Question Formation	QG like Why, when, where, In which	English	Manually evaluated for semantic and syntactic soundness of question by two evaluators
4	Automatic Question Generation Using Software Agents for Technical Institutions(Shivank & Pandey, 2013)	Document Processing, Information Classification, and Question Generation.	Define, Describes, Give an example, and long descriptive questions	English	

5	Automatic Multiple Choice Question Generation System(Ibrahim , 2014)	Extract sentence from Data Set, Prepare Question sentence, Measure the similarity between the question sentence and all sentences in the knowledge base, Return the three sentences that have the highest similarity values, three keywords of three sentences as distractor selection	MCQ	English	In this research out of nearly 145 parsed sentences, there were 109 considered good according to the keywords that are extracted from them.
6	Semantic-Based Automatic Question Generation(Fattoh, 2014)	Input sentence, Feature Extraction through SRL, NER, Choose MCS, Test Sentence pattern, and Test the Question type pattern	WH-questions like who, when, where, why, and how.	English	170 sentences are extracted and mapped into 250 patterns using SRL and NER. The 250 patterns are used in training and testing.

7	Automatic question generation in multimedia-based learning(Skalban, Ha, & Mitkov, 2012)	A QG for documentary video, Definitions, selection of system-based generated post-question, Generation of human QGs, selection of pre-question and selection of images, participants and interface, and procedure.	who, whom, where, whose, when what	English	A large-scale experiment investigating the productivity of generating questions (time taken to post-edit questions vs. time taken to generate questions from scratch) is planned
8	Automatic Generation of Question Bank Based on Predefined Templates(Ahmed Ezz Awad, 2014)	Knowledge Descriptor, Questions Generator, and E-learning Executer.	MCQ with only One Solution, two Solutions, All Of The Above, and none Of The Above		A question bank is developed for course General Biology course (Bio110) at the Faculty of Science, King Abdul-Aziz University (KAU): The bank contains 12 chapters with 239 sub-topics and a total of 46345 questions
9	Automatic Generation of multiple choice questions from domain	ontology-based strategies like class-based, property-based,	MCQ (Choose the correct sentence	English	The generated questionnaires were evaluated in three dimensions:

	ontologies(Konstantinos Kanaris, January 2008)	terminology based strategies			Pedagogical quality, linguistic/ syntactical correctness, and the number of questions produced
10	Review of Question Generation System From Punjabi Text(Jaspreet & Ashok , 2015)	Extract person name, Generate who, Extract location generate Where, and Extract date, generate when,	where, who, and when	Punjabi	
11	Mind the Gap: Learning to Choose Gaps for Question Generation(Lee Becker, 2012)	• sentence selection, • question construction, and • classification/scoring	the Gaps fill the question	English	manually analyze the generated questions and rate the question
12	Linguistic Considerations in Automatic Question Generation(Karen & Paul , 2016)	The source text is divided into sentences, tokenizing, POS tagging, syntactic constituency parsing, and semantic role labeling are used in the system, and the matcher function is called which were return a list of patterns that match the source sentence's predicate-argument structure.	why what one-line questions	English	This evaluation was conducted with one file (Chemistry: Bonds) which had 59 sentences, from which the system generated 142 questions. The average linguistics score per pattern in this evaluation was 5.0 to 4.18

13	Automatic Amharic Factual Question Generation from Historic Text Using RuleBased Approach. (Getaneh, 2021)	Rule-based Approach	WH-questions like who, “□□”,when, ”□□” where,”□□” why,”□□□ “	Amharic	Manually evaluated
14	Amharic Question Generation from Amharic legal Text Documents by Using Deep Learning Approach (Fikir, 2022)	Deep Learning Approach (Amharic question-answer set)	WH (what (□□□□/□□/□□□□/□□□□), when (□□), where (□□/□□□), how much (□□□) and who (□□/□□□/□□□/□□□)) questions.	Amharic	LSTM, CNN, and Bi-LSTM, respectively, achieved accuracy of 92%, 94%, and 95%

2.9. Summary of Related Work

Ideal learners are frequently active self-regulators of their learning who ask examining questions. In other words, they recognize their knowledge gaps, formulate queries that address them, and then research trustworthy information sources to provide answers. Unfortunately, due to the difficulty most students have in recognizing their knowledge gaps, this idealized picture of intelligent inquiry is rarely realized.

A high-quality corpus is essential when designing NLP systems. The POS and NER taggers will perform better as the corpus gets larger. As a result, researchers should develop a high-quality corpus for their research area that is larger in size (Getaneh, 2021).

Design an automatic question generation model for Amharic legal text documents using deep learning is depends on questions and answer pair’s dataset. The model had three modules: a preprocessing module, a module for model building, and a feature mapping trained model with an Amharic question-answer set.So, Deep learning is effective for a specific Amharic question-answer set (Fikir, 2022).

The problem is that they are not effective for large-scale data and on their study does not include mapping more than one question from one sentence at a time Getaneh (2021).
However our approach is automatic question generation and the answer set is generating or mapping from the base sentences.

CHAPTER THREE

3. Methodology

3.1. Introduction

We used an experimental methodological approach for the proposed solutions. Using this methodology, different experimental setups were implemented and evaluated for their effects on the proposed thesis work.

In this thesis, we describe automatic question Generation systems that receive user input in the form of natural language text and produce questions.Questions beginning with a Wh-phrase, i.e. what, who, which, why, when, where, etc. are factoid questions.

Our goal is to develop questions that evaluate the content knowledge that a student has learned via reading a text. We limit our research to fact-based queries about the text's literal information, but we think our methods can be expanded to provide queries regarding inference.

3.2. Development Tools

The other aim of this thesis is to investigate how NLP tools and techniques can beapplied to generate questions from Amharic text. This Thesis wasdescribing a question generation system for Amharic. The transformation of declarative sentences into questions relies on named entity recognition tools. We used Python programming language on Jupyter notebook Anaconda navigator which is a web-based interactive computing notebook environment with appropriate libraries.

Also, This Thesis used the following python Libraries for processing textual data and using multilingual packages.The purpose of a library, a module and a function are the same. They are used for reusing the code. The concept of a function or a library is not new or specific to Python. It has been there for a long and is present in most the computer languages.

- Spacy
- NLTK
- pandas
- re

- text blob

Python is an ocean of libraries that serve various purposes and as a Python developer; you must have sound knowledge of the best ones. Interrogative sentences have primarily been studied in the context of question-generation systems in autonomous language processing. The questions were mainly centered on factual (person, time, place, etc.) and definitional (what is/ ምንነው? who is/ ማንነው?) questions. We restricted ourselves to factual questions in our research.

Question Generation would have involved two tasks, the sentences selected for question generation and question formation. Question formation further has the subtasks of finding suitable question type (Amharic question), and rearranging the phrases to get the final question.

The question generation system was based on Amharic sentences. The sentences initially go through a preprocessing step, which includes all of the procedures. Run a comparison between the raw sentence that wasbeing fed into the system and the one that werebeing processed. Because all sentences in the database were accepted by the algorithm, no semantic selection was made. A set of syntactic transformation rules is used to produce the syntactic structure of inquiries. Finally, the shape of the question's surface is obtained through a series of post-processing operations on the question's deep structure.

We use NLP tools such as syntactic parsers and named entity recognizers, those tools provide valuable structural analyses with which input sentences can be generated into candidate questions.

The proposed system was generating the questions from a given text written in the Amharic language with the help of a rule-based approach which rules to extract and simplify sentences.

The Python programming language Anaconda 3 (spider python 3.9) and Jupyter notebook navigator were used for developing the tools used in this study such as the POS tagger and the Named Entity Recognizer and also a library NLTK. A good performance of NER is important for the type of question and we are designing more templates aiming at generating targeted questions.

3.3. Rule Based Approach

Rule-based processes, also known as an expert or generation systems represent a type of artificial intelligence.In this approach, wide coverage of NLP techniques is used to achieve the accuracy of theanswers retrieved.In this approach, handcrafted rules are developed to generate the questions using the key entities from a given Amharic sentence such as location names, person names, date formats, numbers, etc. and it generates different rules from those entities (Jaspreet & Ashok , 2015).

Table 4: Comparing the rule-based approach with other approaches (Carew, 2020)

Rule Based Approach	Machine Learning and Deep Learning Approach
It need a domain expert	It don't need a domain expert
doesn't need a large amount of data	need a very large amount of data
It need to find patterns manually(Handcraft rule)	find patterns on our behalf as per the data and input features
often a good approach for developing the first cut of our end product and still popular in practice	Complexity and requires more long-term expertise

In the above table scenario, a rule-based approach has been selected to proposeAutomatic Question Generation in the Amharic language, and also in the Amharic language, there is a lack of training data and resources.

An example Based Approach also a rule-based process to generate questions but in work, it is an appropriate approach rule-based approach.Example based approach is used to improve the performance of the existing systems but in the Amharic language, there is no existing system, andbased on the proposed specific objective.

Here we use the POS tagging to find the Nouns and Named Entity Recognition to find the wh_word associated with the Named Entity from the pre-defined rule. The following rule-based Approach is used for the generation of the questions for the sentences that fall into this template:

- Find all the Named Entities present in the sentence.

- Then for every named entity present in the sentence we identify the wh_word associated with that entity

Some general guidelines on how these rules are used in the proposed system:

Rule 1: If any name is found in the sentence then replace it with the who/"ማን" word.

Rule 2: If any city name, state name, country name, or any location name is found in the given sentence then replace it with the where/"የት" word.

Rule 3: If any date format or any year is found then replace it with the when/"መቼ" word.

Rule 4: If any integer or number is found replace it with how much/"ምንያህል" word.

Rule 5: If any reasoning case is found replace it with how much/"ለምን" word.

3.4. Method of Question Construction

Construct questions from Amharic sentences and focus on the structure and relation between words to construct questions. We design different rule-based approaches to construct targeted questions from a construct parser tree. The process is generated based on Question type identification, and question construction. Firstly, we design templates and rules. If an input sentence contains location, person, noun, or time finding them by Named Entity Recognition, we construct Where is? Who is? What is?When etc., type of questions,wewerereplacing a question type with the named entity to generate a question by declarative-to-interrogative.

The simplified declarative sentences were transformed into simple sentences in this stage, which were then transformed into a set of questions based on a predefined question generation rule. The selection of target material, or what the question is asking about, is an important subtask of question generation. In our approach, we look for Question phrases in the input declarative sentence that could be used to generate questions. In Amharic, a question is formed by replacing the target question-type phrase in a declarative sentence with an interrogative pronoun. It does not involve subject-auxiliary inversion or verb deconstruction, unlike question generation in English. In this regard, the Amharic question generation method is not simpler.

We input the following sentences፡

E.g. ‹‹ባህርዳርከተማበከፍተኛየእድገትደረጃላይትገኛለች፡፡››/Amharicterm

/*bahiridariketemabekefitenyayedigetiderejalayitigenyalechi*/Romanized term

"The city of Bahir Dar is in a state of rapid development."/English term

Sample generated questions based on the rule Named Entity“Location” will be:

የትኛዋከተማበከፍተኛየእድገትደረጃላይትገኛለች? (Based on location name)

Once the sentence words have been classified into named entity classes, we consider the relationship between the words in the sentence. As an example, if the sentence has the structure "Location Name "NN” <<ባህርዳር>>”,it will be classified as a "where and who" question type. If it is followed by a preposition that represents a date, then we add the "When" question type to its classification.

3.5. Method of evaluation

We would carry out a first manual evaluation of the questions generated from declarative sentences corresponding to different levels of language difficulty.Automatic evaluation of any natural language generated text is difficult (Rakshit, july 2012).

So, our proposed approach would evaluate manually with human reviewers compare with our approach. The evaluation would be performed by Amharic graduate students with good Amharic proficiency. The evaluators had to give a measure of the quality of the questions generated, on a scale from 1 or 0, the value 0 corresponding to an inadmissible and grammatically incorrect question and the value 1 indicating a question perfect, grammatical, well-formed, and relevant.A comparative evaluation between proposed question generation and questions by humans would have been generated from the same prepared corpus.

To evaluate the performance of the output question generation system for Amharic, we have used standard metrics namely Precision, Recall, and F-measure.The majority of question Generation systems would rely on manual Evaluation. However, the proposed system would be evaluated based on the quality of the output system and the linguistic well-formed type of evaluated criteria. For a given sentenceprecision, recalland F-measurewould be calculated for the proposed Automatic Question Generation. Recall Value: this is the value of the total number of questions that are generated by the proposed system to the total number of questions that can be generated manually by a human being. Precision: This is the total number of accurate questions from all the questions generated by the system.

$$Recall = \frac{Qaqg \cap Qmanually}{Qaqg}$$

$$Precision = \frac{Qaqg \cap Qmanually}{Qmanually}$$

Q_{aqg}: The number of questions generated by an AQG.

$Q_{manually}$: The number of questions generated manually.

3.6. Design of Amharic Question Generation

The designed system is explained in general as well as its components. The main focus of this study is natural language understanding. It is achieved by exploiting the Named Entity Recognition of Amharic sentences.

The designed system detects words that are categorized as person, location, date, or number after first revealing the semantic structure of a sentence via semantic role labeling. As a result, the system's design allows for appropriate who, where, when, and why question asking. The associated semantic argument then loses the marked word. The designed system then determines whether the generated sentence has likely components by applying the remaining named entity recognition.
The designed system focuses on shallow questions, which are mentioned in the methodology part. As shallow questions, the designed system can generate the following question sentences:

- ምን፣የምን/What........? ,
- ማን/ Who...?
- የት/ Where...?
- መቼ/ When…..?,
- ምን/why……?

3.6.1. Dataset Preparation

Before getting to the design and development steps, we need to perform corpus preparation, which is the first thing in the architecture of the system. The corpus used for this study is concerned with the Ethiopian Amharic language's different sources of information. For testing purposes, wewere collect 150 simple sentence corpus from different textbooks, newspapers, electronically available books, and the web and pre-defined tokenized Amharic sentences. When we proposed the system we used a pre-trained dataset from the web and corpus data from the following open source.[1]The corpus data source was news from Walta Information Center Amharic WIC corpus, 200 thousand tokens.[2] In addition to that we used an open repository dataset from GitHub with 9939 tokenized sentences in analytic Amharic language with other multilingual labeled datasets and we have to pickle this corpus dataset.MasakhaNER is the first large publicly available high-quality dataset for named entity recognition (NER) in ten African languages (Adelani, 2021).

3.6.2.Preprocessing

After the data collection, preprocessing is the next thing that follows to make the corpus that we have on hand in the appropriate format and to make it easier for the machine to read. The designed system uses our predefined dataset to handle contractions. The predefined dataset was imported and installed into appropriate libraries and packages related to the Amharic language, such as for open source GitHub dataset sources and import Amharic datasets. The preprocessing steps performed for this thesis include character normalization, short word expansion, tokenization, stop word removal at the sentence retrieval phase and text cleaning for a predefined dataset from open source. In addition, other tools such as POS tagger and NER are used before getting to the question generation components.

When we developed Amharic NLP, we first install pre-trained Amharic NER and POS tagging appropriate libraries, and a package installer for python (pip) was installed into the system.

3.6.3. Question generationApproach

Transformation rules for question generation are learned during the training phase by initial pairs of sentences and questions assigned to these sentences. From parts of the data (input

[1]This corpus was prepared by: https://bit.bdu.edu.et/ict4d/http://wwwiti.cs.unimagdeburg.de/iti_dke/Datasets/Contemporary_Amharic_C...(CACO)-version_1.1.zip

[2]https://github.com/uhh-lt/amharicprocessor.git

sentences and questions), the sentence Pattern is obtained, and the model extract and stores the list of transformation steps based on the difference between these sentence patterns.

It simply records a common set of actions that are known to be performed by similarity matching algorithms (insert token, remove the token, replace token or change the position of token). Later on, we included a particular operation for changing the token's form. The training phase's output is a model that maintains a collection of transformation operations between sentences and questions, which are then used to transform sentences into questions. The similarity between these sentences and previously stored sentences is used to select transformation patterns for fresh phrases. This method is based on a mathematical computation of the difference in composite pattern (similarity is calculated as a portion of identical tokens).

The question phrase following any leading sentence level simplifies phrases on question generation.

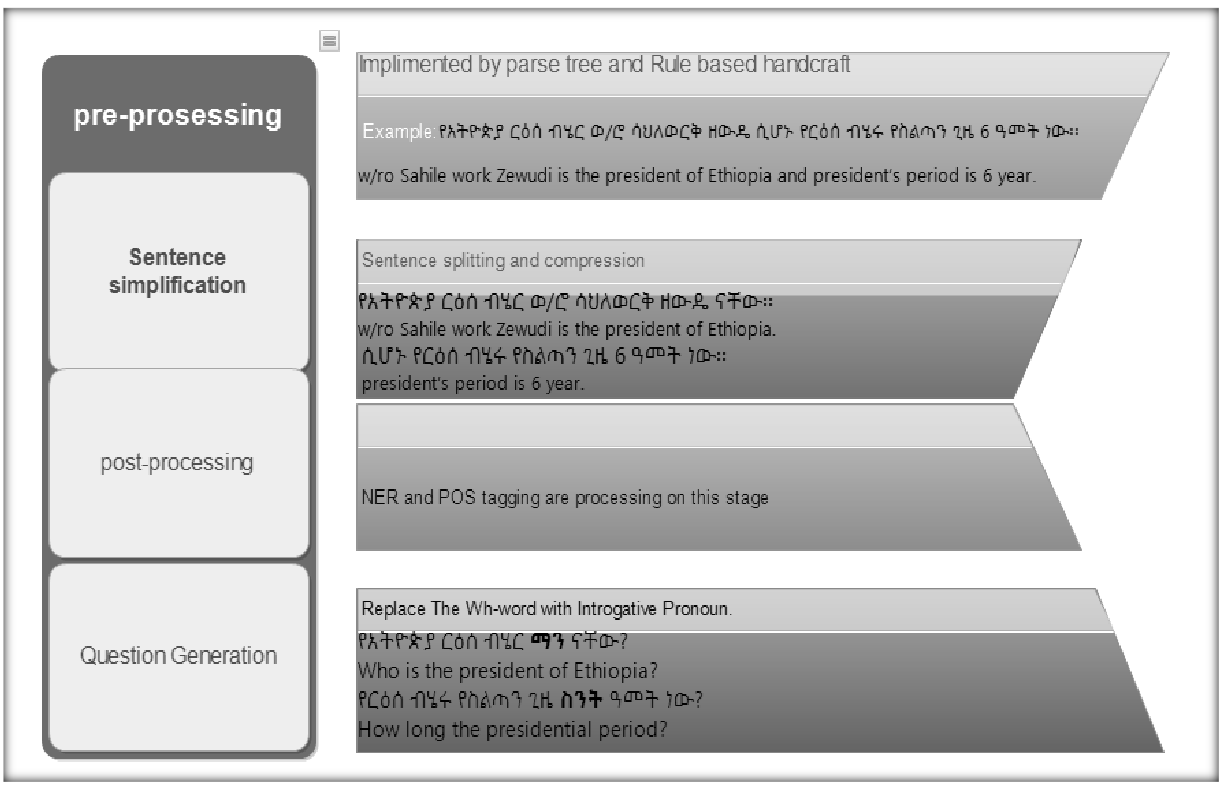

Figure 2: Stage of question Generation

3.6.4. Named Entity Recognition

To generate the questions automatically from a given A corpus, which contains all names related to people, places (cities, states, etc.), countries, and other entities, is required in Amharic. But the main problem is that there is no proper corpus available in the Amharic language that can fulfill the requirements of our system.

So a tool is also needed to create one that extracts the named entities from a given Amharic text and classifies them in proper categories such as location names, person names, etc. These names can be used to generate the questions from the given text.

The NER system can be implemented by using a rule-based approach in which rules are created to extract the named entities from a given text. The accuracy of the NER system depends mainly on the rules created for it. An accurate NER system also tends to increase the accuracy of the Question Generation (QG) system.

Here we use the POS tagging to find the Nouns and Named Entity Recognition to find the wh_*word keys*<*የት: ‘location’*,*ምን: ‘Event’*,*ማን, 'Noun'*,*መቼ: ‘date'*>associated with the Named Entity. The rule-based algorithmis used for the generation of the questions for the sentences that fall into the candidate question.
Then for every Named Entity present in the sentence which takes as a keyword. Itidentifies the *wh_word* associated with that entity. Appropriate libraries and packages were installed and imported for using the NLP system for Question Generation.

3.6.5. Post-processing

Post-processing mechanisms are necessary to ensure suitable formatting and punctuation mark. The process begins with NLP transformations for the input sentence. The type of question to be formed is determined by the Amharic grammar connective. Remove full stop<Aratnetib> ("::") from a given text Amharic sentences and replace by punctuation mark with a question mark (?) is implemented on the postprocessor phase.It also filters out questions that are long in length sentences since such extremely long questions.

3.6.6. Wh- Type Amharic Question Generation

Due to a lack of resources based on the Amharic language, we chose simple sentences for this work. We divide simple sentences into subsections of Amharic sentences, such as subject, verb, and object, at this phase. The sentence's Subject and Object are next processed by the Named Entity Recognizer (NER) to determine the sentence's coarse class classification. Person/human, Location, and Organization are the tagged types of words, according to the NER. The coarse class classification is as follows: Human: This includes the name of a person. Entity: This includes animals, plants, mountains, and any object. Time: This was any time, date, or period such as year, last week, etc. Location: These are the words that represent locations, such as country, city, school, etc. Count: This class isholding all the counted elements, such as the number of workers, measurements like weight and size, etc. Organization: Organizations that include companies, institutes, government, markets, etc.

The proposed system is also integrating with the NER tool to extract and classify the named entities to generate the questions automatically written in the Amharic language.

So, the proposed methodology of a Question Generation System is shown in Figure 3below:

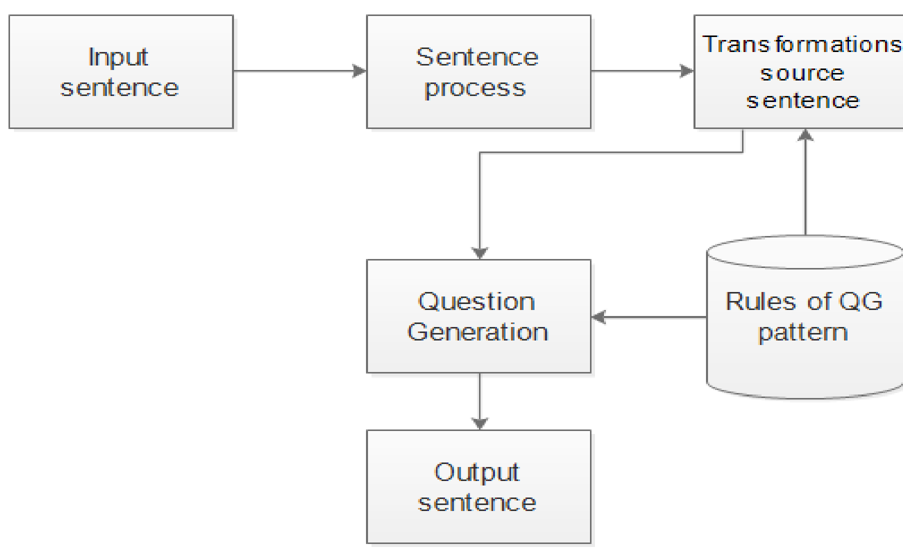

Figure 3: Design of Automatic Question Generation model

3.6.7. The Description of the Model

To formulate the question replace the per-type entity with the appropriateinterrogative term. The rest of the questions will be composed of the matched words from our rule. The system takes the input of a text file from the user, which is first processed by the document processing agent, which extracts the words from the ranked list of words extracted from the input text; the list is based on the occurrence of words. The output of the document processing agent is fed as input into the information classification agent, whose task is to classify the input based on Bloom's taxonomy levels with the help of the rules repository. Then it is processed by the Question Generation Module to the final stage of generating questions with the help of question templates from the database. The output is in the form of questions stored in a text file.

The following sets of transformations are applied to the content to get the final question. If a Wh-question is to be formed, the question word is added just before the auxiliary. In the case of Wh-questions, the question starts with the auxiliary itself as no question word is needed. A question mark (?) is added at the end to complete the question. Then the question type why “□□□”, who “□□”, what “□□”, where “□□”, etc.is added just before the auxiliary is, and a question mark is added at the end to get the final question.

Once the Amharic sentence words have been classified into coarse classes, we consider the relationship between the words in the Amharic sentence. As an example, if the sentence has the structure Verb Human, it is classified as whom and who question types. If it is followed by a preposition that represents the date, then we add the When question type to its classification. So then based on the sentence structure and the sentences are classified based on the various Amharic grammar and its given rules.

This is the interesting and challenging part of the Thesis. In this process, the system was trying to generate Wh-type of Amharic questions i.e. who/ማን, what/ምን, why/ለምን scenario-based questions? To access the ability of knowledge grasped by the user, these kinds of questions help in gauging it to a certain extent. The algorithm used to generate this type of question is described below:

i. Select simple important sentences.

```
# This function is used to tokenize and split into sentences
```

```
defAmharic_question_Generation():
global sentences
tokenizer = nltk.data.load('tokenizers/punkt/Amharicsent.pickle')
fp = open('Testing.txt','r+', encoding="utf-8")
data = fp.read()
sentences = tokenizer.tokenize(data)
discourse()
```

ii. Identify the keyword of a named entity using the logic described as a generation code on python programming.

```
# This function is used to get the required wh-word
defget_wh_word(entity, sent):
   wh_word = " "
if entity[1] in ['TIME', 'DATE']:
      wh_word = 'መቼ'
elif entity[1] == ['PRODUCT', 'EVENT', 'WORK_OF_ART', 'LAW',
'LANGUAGE']:
      wh_word = 'ምን'
elif entity[1] in ['PERSON']:
         wh_word = 'ማን'
elif entity[1] in ['NORP', 'FAC',' ORG', 'GPE', 'LOC']:
index = sent.find(entity[0])
if index == 0:
         wh_word = "በማን"
else:
         wh_word = "የምን"
else:
      wh_word = "የት"
return wh_word
```

iii. Terminate the sentence and replace full stop/four-point (::) with “?”

```
# this function generates questions based on NER templates to remove full stop (::)
defgenerate_one_word_questions(sent):
```

```
named_entities = get_named_entities(sent)
questions = []
if not named_entities:
return questions
for an entity in named_entities:
    wh_word = get_wh_word(entity, sent)
if(sent[-1] == ':'):
sent = sent[:-2]
ifsent.find(entity[0]) == 0:
questions.append(sent.replace(entity[0],wh_word) + '?')
continue
return questions
```

In order to try and stumble on whether or not a sentence must be related to wh_word listing of styles changed into advanced through generalizing from the sentence capabilities which we observed ourselves use whilst we generated questions from the text. Each sample changed related to a given set of policies and the query changed into formulated primarily based totally on matching sample. For example, if the goal keyword decided on is classed as a Person entity then the clause uses extra recognized Amharic wh_word kind questions. But at the equal time, the procedure of producing this Wh_word type clause questions does now no longer appear to be clean as area know-how is widespread and a whole lot of education is needed.

After marking unmovable phrases, the system iterates over the possible answer phrases, generatingpossible questions for each one. This process also enables the extraction of questions to accompanyeach question.Recall that question phrases can benoun phrases (labeled “NNP”).For Location Noun Phrases,the system only extracts the question phrase **(**e.g., *‹‹የትከተማየጥናትአው፡ደርዕይተካሄደ››?* Extract question phrases from Amharic sentences or phrases such as *‹‹ባህር፡ዳርከተማየጥናትአው፡ደርዕይተካሄደ››***).**

Our system generates questions from a set of sentences. It can generate more than one question for a single sentence, depending on the length of the sentence. Humans are also given the same set of sentences to generate questions.

For example:

Sentence: የአማራክልልዋናከተማባህርዳርትባላለች፡፡

============# System-Generated Questions #=============

Question 1: የአማራክልልዋናከተማ**ማን**ትባላለች?

Question 2: ባህርዳር**የማን**ዋናከተማነች?

============# Human-Generated Questions #=============

Question 1: የአማራክልልዋናከተማ**ማን**ትባላለች?

Question 2: ባህርዳር**የማን**ዋናከተማናት?

CHAPTER FOUR

4. Experiment

4.1. Experimental setup

In this section, we describe the experimental setup of a question generation system which includes the preprocessing and rule-based induction in addition to evaluation metrics.We began this part by discussing a few datasets that can be utilized on the Question Generation task and how the task can be assessed.

For this type of research and other scientific computing (machine learning applications, large-scale data processing on NLP, predictive analytics, and so on), the suggested system is developed using Anaconda, a Python distribution that attempts to data-science package management and deployment. Windows-compatible data science packages are included in the distribution. In Anaconda Navigation, the Jupyter notebook programming language is also a tool used to record the learned rules and build user-friendly user interfaces that simplify how users interact with the system.

4.2. Named Entity Recognition Based Templates

Sentences that do not contain any discourse marker fall into this category such as simple assertive or declarative sentences. Here we use the POS tagging to find the Nouns and Named Entity Recognition to find the wh_word associated with the Named Entity. The following rule based algorithm is used for the generation of the questions for the sentences that fall into this template:

- Find all the Named Entities present in the sentence.
- Then for every named entity present in the sentence we identify the wh_word associated with that entity.

```python
# This function is used to get the required wh word
def get_wh_word(entity, sent):
    wh_word = " "
    if entity[1] in ['TIME', 'DATE']:
        wh_word = 'መቼ'
    elif entity[1] == ['PRODUCT', 'EVENT', 'WORK_OF_ART', 'LAW', 'LANGUAGE']:
        wh_word = 'ምን'
    elif entity[1] in ['PERSON']:
            wh_word = 'ማን'
    elif entity[1] in ['NORP', 'FAC' ,'ORG', 'GPE', 'LOC']:
        index = sent.find(entity[0])
        if index == 0:
            wh_word = "የት"
        else:
            wh_word = "ለምን"
    else:
        wh_word = "የማን"
    return wh_word
```

Figure 4: Identification of WH Word

- Now, we remove full stop (።) from the sentence, if present.
- If Named Entity is present in the beginning of the sentence, we simply replace the Named Entity with its wh_word and append the question mark at the end. The procedure is continued for the next Named Entity in the sentence.
- Then the rearrangement of the question part is done to satisfy the Grammatical Syntax of Language. Finally, the wh_word is prepended and the '?' is appended to the sentence.

```python
# This function generate questions based on NER templates
def generate_one_word_questions(sent):
    named_entities = get_named_entities(sent)
    questions = []
    if not named_entities:
        return questions
    for entity in named_entities:
        wh_word = get_wh_word(entity, sent)
        if(sent[-1] == '።'):
            sent = sent[:-2]
        if sent.find(entity[0]) == 0:
            questions.append(sent.replace(entity[0],wh_word) + '?')
            continue
```

Figure 5: Placing WH word and Question Mark

Now, we remove the Amharic full stop (።) from the sentence, if existed. If Named Entity is not present in the beginning, we find the Auxiliary Verb present in the sentence.

The following rule-based algorithm is used for the generation of the questions for the sentences that fall into this template:

- Find all the Named Entities present in the sentence.
- then for every named entity present in the sentence we identify the *wh_word* associated with that entity.

```python
# This function generate questions based on Amaric NER templates:
def generate_one_word_questions(sent):
    named_entities = get_named_entities(sent)
    questions = []
    if not named_entities:
        return questions
    for entity in named_entities:
        wh_word = get_wh_word(entity, sent)
        if(sent[-1] == '።'):
            sent = sent[:-4]
        if sent.find(entity[0]) == 0:
            questions.append(sent.replace(entity[0],wh_word) + '?')
            continue

            if question_part[-1] == ' ':
                question_part = question_part[:-1]
        else:
            for i, grp in enumerate(tags):
                #Break the sentence after the first non-auxiliary verb
                word = grp[0]
                tag = grp[1]
                if(re.match("VB*", tag) and word not in aux_list):
                    question_part += word
                    if i<len(tags) and 'NN' not in tags[i+1][1] and wh_word != ' ':
                        question_part += " "+ tags[i+1][0]
                    break

                question_part += word + " "
        question = question_part.split(" "+ aux_list[pos])
        question = [aux_list[pos] + " "] + question
        question = [wh_word+ " "] + question + ["?"]
        question = ''.join(question)
        questions.append(question)
    return questions
```

Figure 6: NER Algorithm*(Michael & Noah A., 2010)*

The system is here, a given statement, for example, is discussed to describe powerful of the system and its capability to extend and test functionality. The testing mode was taken the most popular and official Ethiopian language Amharic statement is <<*የባህርዳርከተማበከፍተኛእድገትላይትገኛለች።*>>. The statement is very simple and contains a

Location Namemark <<*ባህርዳር*>> that let the system predict a location in the statement so it could suggest a where/ የት/wh_word question type.

If there is a Named Entity Recognition and predefined part of speech tagging rule stored in the path text of named entity as <<*ባህርዳር*>> is a location can be detected by it so the key solution based on detecting a custom location name.

For the given articulation <<የባህርዳርከተማአስተዳደርበከፍተኛእድገትትገኛለች፡፡>> it would be predefined parsed utilizing OpenNLP parser as take after, PRP, VBD, PRP, NNS, IN, DT, NN, individually.

When we use pre-defined dataset, first install Amharic compatible multilingual package by using the following class.

```
#Class initializations
nlp = spacy.load('en_core_web_sm')
stemmer = LancasterStemmer()
```

```
nlp = spacy.load('xx_ent_wiki_sm') # multilinguail dataset
```

Figure 7: class initialization

The way of Some Amharic text Named Entities tagger is looks like the functions are as follows:

Label: LOCATION

Word: ('የባህርዳር', 'NN')

Label: ORGANIZATION

Word: ('ከተማ', 'NNP')

Word: ('አስተዳደር', 'NNP')

Word: ('በከፍተኛ', 'VBD')

Word: ('እድገት', 'VBD')

Word: ('ላይ', 'PRP')

Word: ('VB፡፡', 'NN')

From GitHub open public repositories The Amharic dataset is loading into the Python library. The dataset matches the Amharic-English machine translation corpus prepared through the website.

```python
from datasets import load_dataset
dataset = load_dataset("masakhaner", 'amh')
```

Figure 8: Installing NER dataset (Adelani, 2021)

The function is used to generate the questions from sentences that have already been preprocessed. For in the entire text, if Named Entity has existed in the sentence, we simply replace the Named Entity with its wh_word and append the question mark at the end.
Sentences that do not contain any discourse marker fall into this category such as simple assertive or declarative sentences. Here we use the POS tagging to find the Nouns and Named Entity Recognition to find the wh_word associated with theNamed Entity. The following rule-based algorithm is used for the generation of the questions for the sentences that fall into this template:

Find all the Named Entities present in the sentence.

Then for every named entity present in the sentence, we identify the Wh_word associated with that entity.

The procedure is continued for the next Named Entity in the sentence. This function generates questions based on NER templates.The implementation and experiment part of the proposed system is importing appropriate Python libraries and placing wh_word and the question mark is shown below.

Import from python package which supports generate Amharic language

```python
[*]: # Imports
import nltk
import nltk.data
import re
import amtokenizers
import spacy
import csv
import json
import vert
import textblob
import amseg
import gensim
import codecs
import pickle as pickle
import datasets
import pandas as pd
```

Input is an unstructured text (e.g. sentences) output is a question in a sentence form (e.g. interrogative sentence), and the sample output of the question generation Pattern Application is shown in the below figure [9].

```
In [43]:  1  Amharic_question_Generation()

1
[ዓ.ነገር]:  የባህር ዳር ከተማ አስተዳደር በከፍተኛ ደረጃ እድገት ላይ ትገኛለች::
[ጥያቄ]:  የትኛዋ ከተማ አስተዳደር በከፍተኛ ደረጃ እድገት ላይ ትገኛለች?

-----X-----
```

Figure 9: Sample wh_word Generated Amharic Questions

Therefore, based on the experimental results, the developed proposed system can be used in the educational environment and for other users such as schools and colleagues because of its high ability to adapt to different levels of Amharic sentences, articles, or even tutors.

4.3. Evaluationand Result

Automatic evaluation of any Question Generation system is challenging for the following reasons: first is difficult to agree on standard evaluation data and second one is no single set of Generate Questions is correct (Rakshit, s., 2012). As a result, the majority of question Generation systems rely on manual Evaluation. However, the proposed system wasevaluated based on the quality of the output system and the linguistic well-formed comparison with human judgment.

Human evaluation was also performed to measure the quality of generated questions. Human evaluators evaluate the submitted questions according to evaluation criteria, which are difficulty, relevance, syntactic correctness, and ambiguity based on Amharic grammar.

So, we agree in principle that human evaluation is indeed necessary for a proper evaluation of the Amharic Question Generationsystem output. In this work, we go with manual evaluation based on expert-based evaluation, where two (2) human evaluators for Amharic speakers that are proficient in Amharic and used one expert language teacher to assess the quality of the questions the other students, give scores to questions generated from the proposed system. The evaluators provide a score of one(1) when the questions are syntactically well-formed and normal and a score of zero(0) when they are syntactically unacceptable and not well to determine the syntactic correctness of the ordering or arrangement of words and phrases to construct proper sentences.

Similarly, for fluency, the raters give a score of one(1) when the questions are fluent and a score of zero(0) when they are not.

$$\textbf{Syntactic Score} = \frac{No.\,of\ Syntactically\_correct\_Questions}{Total\ No.\,of\ Questions}$$

$$\textbf{Fluency Score} = \frac{\text{No. of Fluent Questions}}{\text{No. of Syntactically Correct Questions}}$$

We also defined composite scores. In general, composite scores are the average of individual composite scores. An individual composite score summarizes the scores along a dimension, e.g. syntactic correctness, and is computed by taking the average of individual scores shown by the formula below where Qn is the total number of questions or Length of Questions.

$$Syntactic\_Score = \frac{\sum_{Qn} individual_{\ Score}}{Qn}$$

The composite score for specificity is more challenging to define. The goal would be to have a syntactic score with values from 1 and 0. Also, fluency questions are evaluated in the same way.

4.3.1. Criteria based on linguistic well-formed for System Outputs and Human Judges

Syntactic Correctness and Fluency are the criteria that were identified for question generation. For the evaluated purpose we used 150 sentences. We give the questions generated per sentence to human evaluators who rate based on described two criteria. The average ratings of both criteria that our proposed system gets are 0.8278 per 1.0 on syntactic correctness and 0.8878 per 1.0 on fluency.

Table 5: Briefly Comparison of AQG Systems: Syntactic correctness and Fluency

Total No. of Input sentences.	Total No.Generated Questions	Syntactic correctness	Fluency
150	122	0.8278 per 1.0	0.8878 per 1.0
	81.33%	82.78%	88.78%

Fromthe above figure, it is shown thatout of 150 sentences 122, means 81.33% is generated question and from these generated question syntactic correctness or truly generated questions82.78% is retrieved and from this retrieved list 88.78% was fluency questions. Generally, the result shows that the proposed system works with an overall efficiency of the proposed system were 81.33%.

Based on the experimental results and the testing, the proposed system can be used in an educational environment such as a school and colleagues because of its high ability to adapt to different levels of topics, articles, or even lectures based on the Amharic Language.

4.3.2. Criteria for Quality of System Outputs and Human Judges

The proposed system has been tested on 20 different sentences that have been collected from various Amharic websites and Amharic language science textbooks. To evaluate the performance of the question generation system for Amharic, we have used three standard metrics namely Precision, Recall, and F-measure.

Our system was generating questions, and we have a set of ground truth questions generated by humans, we assume the humans have generated every acceptable question possible for every sentence, we know that any question our system is generating is a positive either True or False(Manisha & Ambuja , 2017).

Recall Value: this is the value of the total number of questions that are generated by the proposed system to the total number of questions that can be generated manually by a human being.

Precision: This is the total number of accurate questions from all the questions generated by the system.

F-Measure: This is defined as the harmonic mean of recall and precision. The Evaluate result which comparison:

$$Recall = \frac{Qaqg \cap Qmanually}{Qaqg}$$

$$Precision = \frac{Qaqg \cap Qmanually}{Qmanually}$$

$$\mathrm{F-Measure} = \frac{2 * Recall * Precision}{(Recall + Precision)}$$

Q_{aqg}: The number of questions generated by an AQG.

$Q_{manually}$: The number of questions generated manually.

The results are important to define well-founded evaluation metrics for AQG tasks. Classification performance is measured using F-measure, precision, and recall (figure11).

Table 6: Result table of the proposed system

Question Type	No. input Sentences	No. Generated Question	No. of Matching	Recall	Precision	F-Measure
Wh_word type Questions.	20	17	14	82.35%	70.00%	75.67%

Fromthe above figure, it is shown thatout of 20 input sentences, the Automatic question generation system generates 17 questions which of them 14 questions are matching with in manually generated Questions which means 82.35% are generatedFrom comparisons with the Total number Human-generated questions70% is precision of the performance of proposed of automatic question generation system and the F1 Score is 75.67%.

Generally, the result shows that the proposed system works with an overall efficiency based on syntactic correctness is 82.78% generated and 88.78% fluency generated questions. We also present an evaluation of the output system result, which shows that Recall is 82.35%, precision is 70.00%, and F-measure is 75.67%.

4.4. Shortcoming

The results revealed that the technique is still weak in various areas, particularly the number of questions that are not ideal, including grammatical and semantic problems, as well as being incomplete.One of the limitations of the currently proposed system is itsusefulness over a limited domain.

Once we tag the sentences in consideration and their associated body of texts, we use some general purpose rules to create some basic questions even though the answer is only from the existing in the body of texts. For example, ‹‹የባህርዳርከተማበከፍተኛደረጃላይትገኛለች›› is tagged as an organization, so we generate a question ‹‹የማንከተማበከፍተኛደረጃላይትገኛለች? ››

the main motivation behind generating such questions is to add variety to the generated question space.

Our proposed system does not achieve the best performance across all wh_word question categories.So, adapting the designed system to the Amharic language would not be easy due to the lack of syntactic and semantic parsers. Without high-performance parsers, adapting predefined rules into the Amharic language would not give a good performance.

Some of the entities such as things, objects, etc. are not recognized by Named Entity Recognition by the appropriate Spacy library. All types of wh_word questions are not being generated by our system. As the sentence becomes more and more complex, it would be difficult to generate an appropriate question through our proposed system and wh_word is becoming for the beginning of the sentences is the big challenge of the study.

The problem is one of the lexical challenges and semantic problems that we have wanted to state.

The other shortcoming of this work, as the sentence becomes more and more complex; it would be difficult to generate an appropriate question generation through our proposed system.

CHAPTER FIVE

5. Conclusion and Recommendation

5.1. Conclusion

AQG is a thrust area for researchers in natural language processing (NLP). In this Thesis work, we attempted and proposed an Automatic question generation system that takes an Amharic sentence as input and produces a good-quality question based on the text, such that the answer to the question can be worked out from the base sentences.The system has been evaluated for the syntactic correctness of the question by evaluators.The first task we have tackled is normalizing the sentences so that standard sentences are indexed and pattern-matching relevant entities during preprocessing of a sentence.

This thesis supports the idea that natural language processing can help students and teachers efficiently create instructional content and self-calibration. It provides solutions to some of the major challenges in question generation and an analysis and better understanding of those that remain.

The challenges of QG in Amharic language was the same words and phrases can have different meanings according the context of a sentence and many words – especially in English – have the exact same pronunciation but totally different meanings. Ambiguity in Amharic QG refers to sentences and phrases that potentially have two or more possible interpretations.

- Lexical ambiguity: a word that could be used as a verb, noun, or adjective.
- Semantic ambiguity: the interpretation of a sentence in context.

Generally, the proposed system implemented the Wh-Type of question generation which Inputs sentence, Feature Extraction through NER, Test Sentence pattern, and Tests the Question type pattern then it generated all possible questions from a given sentence.

So, our proposed system results confirm that statement by performing a human evaluation study when generating the most natural (human-like) questions. Question Generation for the Amharic language was originally an idea for optimal use. From the beginning, an idea was to produce whether useful or not, this Question Generation approach was thought to bring forward some interesting aspects.

When we proposed the system we used a pre-trained dataset from the web and corpus data from the following open source.[3] The corpus data source was news from Walta Information Center 200 thousand tokens. In addition to that we used an open repository dataset from GitHub with 9939 tokenized sentences in analytic Amharic language with other multilingual labeled datasets and we have to pickle this corpus dataset.
Out of 150 test sentences 122, that means 81.33% is generated question and from this generated question syntactic correctness or truly generated questions82.78% is retrieved and from this retrieved list 88.78% was fluency questions.

Generally, the result shows that the proposed system works with an overall efficiency based on syntactic correctness is 82.78% generated and 88.78% fluency generated questions. We also present an evaluation of the output system result, which shows that Recall is 82.35%, precision 70.00%, and F-measure is 75.67%.

Generalizing from this work, it can be said that NLP techniques have the potential to improve educational technologies by assisting teachers in making effective use of fresh digital text resources and enabling the delivery of instruction that is tailored to students' interests and learning requirements. For now, we have to stick to using subjective human evaluation to decide whether our question-generation system is good or not.

In this thesis, we have made the following contributions to the literature on natural language processing and educational technologies:

- We emphasize our scientific contribution to the Automatic Question Generation field, for the Amharic language.We also fill in deficiencies for the AQG community as well by proposing new ways to moderate the difficulty of generated questions in the Amharic language.
- Using python packages feature rules and training data to identify whether a question is “who/ማን”,“who is/የማን”,”why/ለምን”,”what/ምን"," when/መቼ” type of Amharic wh_word question type.
- Currently, a proposed system has been implemented on the python platform. However, this domain is open-ended as the knowledge base required is extremely

[3]This corpus was prepared by: https://bit.bdu.edu.et/ict4d/
http://wwwiti.cs.unimagdeburg.de/iti_dke/Datasets/Contemporary_Amharic_C...(CACO)-version_1.1.zip

large and cannot be completed with limited sources and time. In future work, another interesting possibility is to feed the questions to the machine for conversations. Since the system tries to generate all possible questions from the sentence, it can train the machine to ask questions to the user to know more about the topic as well.

5.2. Recommendation

Our proposed system tries to automate the question generation process to great extent and very little manual intervention is needed to ensure the semantic correctness of the questions generated.

Our proposed approach to automatically generate questions have an important on the broad area of learning tools that are needed in order to help teachers, learners, and other users especially since Amharic is the second language and to save their time and effort for creating pedagogical content and to assist educational technology developers to reduce development costs (Ming &Vasile, April-June 2017).

The development of effective evaluation methods is another area where QG will see further progress. The development of automatic QG performance measurements similar to the BLEU metric for machine translations or the ROUGE (Lin, 2004) meter for summaries would be very helpful.

Some of the entities such as things, objects, etc. are not recognized by Named Entity Recognition by the appropriate Spacy library. As the sentence becomes more and more complex, it would be difficult to generate an appropriate question through our proposed system and wh_word are becoming for the beginning of the sentences is the big challenge of the study so the other researcher tried to address this problem too.

Finally, this Thesis work recommended the following key points for future work:

- Our syntactic approach to WH word phrase QG can be characterized as follows: a moderately difficult analysis step using well-studied but imperfect tools for a rule-based transfer step leveraging linguistic knowledge about syntactic embedding and question formation, and a simple generation step.
- Our proposed system does not achieve the best performance across all wh_word question categories. So, adapting the designed system to the Amharic language would

not be easy due to the lack of syntactic and semantic parsers. Without high-performance parsers, adapting predefined rules into the Amharic language would not give a good performance.

- Our analysis may provide future QG researchers, as well as those working on related problems, with a useful roadmap.
- The feature will attempt to include a question estimator that estimates the quality of generated questions, removes duplicates and wrong questions, and sorts the questions by estimated quality.

References

A, M. B. (Jun 2021). *Automatic Question Generation Based on Sentence Structure Analysis using Machine Learning Approach.* Kempelen Institute of Intelligent Technologies, Natural Language Engineering. Bratislava, Slovakia: Miroslav Blˇst´ak, Viera Rozinajov´.

Ahmed Ezz Awad, M. Y. (2014). Automatic Generation of Question Bank Based on Pre-defined Templates. *International Journal of Innovations & Advancement in Computer Science* (pp. 80-87). Jeddah, KSA.: KING ABDULAZIZ UNIVERSITY.

C. Zhang, M. H. (2013). Chinese sentence compression: Corpus and evaluation. *Based Naturally Annotated Big Data* (pp. 257–267). 12th Int. Conf. Chinese: Chinese Comput.Linguistics Natural Lang. Process.

Carew, J. M. (2020, Jul 23). *TechTarget.* Retrieved october 12, 2021, from https://www.techtarget.com/searchenterpriseai/feature/How-to-choose-between-a-rules-based-vs-machine-learning-system: https://www.techtarget.com/searchenterpriseai/feature/How-to-choose-between-a-rules-based-vs-machine-learning-system

Chal, Y. l., & Hasan, S. A. (December 2012). Towards Automatic Topical Question Generation. *Proceedings of Technical Papers* (pp. 475–492). Mumbai: COLING 2012.

Chen, C.-Y. L.-C. (2006). Fast–an automatic generation system for grammar tests. *Association for Computational Linguistics* (pp. 1-4). Sydney,: ACL on Interactive Presentation Sessions.

D. Mollá, M. V. (2006). Named entity recognition for question answering. (pp. 51–58). ALTW: Proc. ALTW.

F. S. (2022). *Amharic Question Generation from Amharic legal Text.* BAHIR DAR INSTITUTE OF TECHNOLOGY, Faculty of Computing. Bahr Dar,Ethiopia: BAHIR DAR UNIVERSITY.

Fasya, E. L. (August 2017). *AUTOMATIC QUESTION GENERATION FOR VIRTUAL HUMANS.* Human Media Interaction. Enschede, The Netherlands: University of Twente.

Fattoh, I. E. (2014). Semantic Based Automatic Question Generation using Artificial Immune System. *Computer Engineering and Intelligent Systems, Vol.5, No.8* , 74-82.

G. D. (2021). *Automatic Amharic Factual Question Generation From Historic Text Using Rule Based Approach.* NATURAL AND COMPUTITIONAL SCIENCE, SCHOOL OF INFORMATION SCIENCE. ADDIS ABABA,Ethiopia : ADDIS ABABA UNIVERSITY.

GAMBÄCK, B., & ASKER, L. (21 May 2014). *Experiences with Developing Language Processing Tools and Corpora for Amharic.* Sweden: Swedish International Development Cooperation Agency.

Gasser, W. M. (2019). *Learning Morphological Rules for Amharic Verbs Using Inductive Logic Programming.* Addis Ababa: Addis Ababa University,.

Ghader , K., Jared , L., Bijan , P., Uli , S., & Salam , A.-E. (2019, November 12). *A Systematic Review of Automatic Question Generation for Educational Purposes.* Manchester, UK: Springer.

Husam Ali, Y. C. (2010). Automation of Question Generation From Sentences. *Proceedings of QG2010* (pp. 58-68). Lethbridge, AB, Canada: The Third Workshop on question Generation.

Ibrahim , E. F. (2014). Automatic Multiple Choice Question Generation System for Semantic Attributes Using String Similarity Measures. *Computer Engineering and Intelligent Systems* (pp. 66-73). Cairo, Egypt.: IISTE Knowledge Sharing Partners.

Jaspreet , K., & Ashok , K. B. (2015, September 27). AUTOMATIC QUESTION GENERATION FROM HINDI TEXT USING HYBRID APPROACH. *2nd Internatinal Conference on Science, Technology and Management*, 1410-1427.

Karen , M., & Paul , T. (2016). Infusing NLU into Automatic Question Generation. *Proceedings of The 9th International Natural Language Generation conference* (pp. 51–60). Edinburgh, UK: Association for Computational Linguistics.

Kenneth , W. (2012). Automatic question generation for Swedish: The current state. *Proceedings of the SLTC 2012 workshop on NLP for CALL* (pp. 71–79). Gothenburg, Sweden: Linköping Electronic Conference Proceedings 80.

Kenneth, W. (2011). Automatic Question Generation from Swedish Documents as a Tool for Information Extraction. *NODALIDA 2011 Conference Proceedings* (pp. 323–326). Borås: Gunta Nespore and Inguna Skadin.

Konstantinos Kanaris, K. K. (January 2008). AUTOMATIC GENERATION OF MULTIPLE CHOICE QUESTIONS FROM DOMAIN ONTOLOGIES . *IADIS International Conference e-Learning 2008* (pp. 427-434). Samos, Greece: DBLP.

Lee Becker, S. B. (2012). Mind the Gap: Learning to Choose Gaps for Question Generation . *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 742–751). Montréal, Canada: Association for Computational Linguistics.

Leite, B. J. (2020). *Automatic Question Generation for the Portuguese Language.* Portuguese: FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO.

Lindberg, D. (June 26, 2013). *AUTOMATIC QUESTION GENERATION FROM TEXT FOR SELF-DIRECTED LEARNING.* Computing Science. canada: SIMON FRASER UNIVERSITY.

M. A., & Y. A. (September 2014). Development of Amharic Morphological Analyzer Using Memory-Based Learning. *Development of Amharic Morphological Analyzer* (pp. 1-13). Addis Ababa University, Ethiopia: ResearchGate.

Manish Agarwal, R. S. (2011). *Automatic Question Generation using Discourse Cues.* Hyderabad, AP, India: Language Technologies Research Center.

Manisha , D., & Ambuja , S. (2017, NOVEMBER 10). Automatic question generation approaches and evaluation techniques. *CURRENT SCIENCE, 113*(9), 1683-1691.

Maria , C., & Detmar , M. (September 8, 2017). Question Generation for Language Learning:From ensuring texts are read to supporting learning. *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 334-344). Copenhagen, Denmark: Association for Computational Linguistics.

Medhanit , G. (February 2019). *Amharic Question Answering for Factoid and List Questions using Machine-learning Approaches.* SCHOOL OF GRADUATE STUDIES, SCHOOL OF INFORMATION SCIENCE. ADDIS ABABA,Ethiopia: ADDIS ABABA UNIVERSITY.

Michael , H., & Noah A., S. (2010). *Question Generation via Overgenerating Transformations and Ranking.* Pittsburgh: Carnegie Mellon University.

Michael, H. (2011). *Automatic Factual Question Generation from Text.* Language Technologies Institute. Pittsburgh: School of Computer Science,Carnegie Mellon University.

Ming, L., & Vasile, R. a. (APRIL-JUNE 2017). Automatic Chinese Factual Question Generation. *IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES, 10*(2), 194-204.

Miroslav, B. (2018). Automatic Question Generation Based on Sentence Structure Analysis. *Information Sciences and Technologies Bulletin of the ACM Slovakia*, 1-5.

Niko , M. (2007). Automatic Generation of Prediction Questions during Program Visualizatio. *Electronic Notes in Theoretical Computer Science*, 43–49.

Onur , K. (July 2018). *AUTOMATIC QUESTION GENERATION USING NATURAL LANGUAGE PROCESSING TECHNIQUES.* Computer Engineering. IZM IR: Izmir Institute of Technology.

Payal, K., Konigari , R., Mukul , H., & Manish, S. (July 15 - 20, 2018.). Automatic Question Generation using Relative Pronouns and Adverbs. *Student Research Workshop,* (pp. 153–158). Melbourne, Australia,: Association for Computational Linguistics.

Rakshit, s. (july 2012). *Automatic Question Generation using Discourse Cues and Distractor Selection for Cloze Questions.* Computer Science. India: Language Technology and Research Center (LTRC).

Rodrigues, H. P. (May 2017). *Learning Semantic Patterns for Question Generation and Question Answering.* mellon: Carnegie,mellon University.

Seid, M., & Mulugeta , L. (n.d.). TETEYEQ: Amharic Question Answering For Factoid Questions., (pp. 17-25).

Sheetal , R., & Dr. Y. R. , G. (2015, January). Literature Review of Automatic Question Generation Systems. *International Journal of Scientific and Research, 5*(1), 1-5.

Shivank, P., & Pandey, S. (2013). Automatic Question Generation Using Software Agents for Technical Institutions. *International Journal of Advanced Computer Research.Volume-3 Number-4 Issue-13*, pp. 2277-7970. Tamil Nadu,India: Computer Society of India.

Sisay, F. A. (June 2005). Part of Speech tagging for Amharic using Conditional Random Fields. *roceedings of the ACL Workshop on Computational Approaches to Semitic Languages* (pp. 47–54,). Amsterdam, The Netherlands: Association for Computational Linguistics.

Skalban, Y., Ha, L., & Mitkov, R. (2012). *Automatic Question Generation in Multimedia-Based Learning.* UK : Proceedings of COLING 2012.

Swali, D., Palan , J., & Shah, I. (2016, December). Automatic Question Generation from Paragraph. *International Journal of Advance Engineering and Research Development, volumr 3*(Issue 12), 73-78.

Tilahun, A., & Mulugeta, L. (2019). *Amharic Question Answering for Biography, Definition, and Description Questions.* Communications in Computer and Information Science. Addis Ababa: Information and Communication Technology for Development for Africa.

Yvonne, S., Le An, H., Lucia, S., & Ruslan, M. (December 2012). Automatic question generation in multimedia-based learning. *Proceedings of COLING 2012* (pp. 1151–1160). Mumbai: COLING 2012.

ይማም, ባ. (1987). *የአማርኛሰዋሰው.* Addis Ababa, Ethiopia: በት.መ.ማ.ድ.

Appendix

#sample screenshot python programming code with appropriate libraries

```
In [*]:
# Imports
import nltk
import nltk.data
import re
import amtokenizers
import spacy
import csv
import json
import vert
import textblob
import amseg
import gensim
import codecs
import pickle as pickle
import datasets
import pandas as pd
```

```
In [14]:
# This function is used to get the named entities
def get_named_entities(sent):
    doc = nlp(sent)
    named_entities = [(X.text, X.label_) for X in doc.ents]
    return named_entities
```

```
In [15]:
# This function is used to get the required wh word
def get_wh_word(entity, sent):
    wh_word = " "
    if entity[1] in ['TIME', 'DATE']:
        wh_word = 'መቼ'
    elif entity[1] == ['PRODUCT', 'EVENT', 'WORK_OF_ART', 'LAW', 'LANGUAGE']:
        wh_word = 'ምን'
    elif entity[1] in ['PERSON']:
            wh_word = 'ማን'
    elif entity[1] in ['NORP', 'FAC' ,'ORG', 'GPE', 'LOC']:
        index = sent.find(entity[0])
        if index == 0:
            wh_word = "በማን"
        else:
            wh_word = "የምን"
    else:
        wh_word = "የማን"
    return wh_word
```

```
In [10]: # This function generate questions based on NER templates
def generate_one_word_questions(sent):
    named_entities = get_named_entities(sent)
    questions = []
    if not named_entities:
        return questions
    for entity in named_entities:
        wh_word = get_wh_word(entity, sent)
        if(sent[-1] == '።'):
            sent = sent[:-2]
        if sent.find(entity[0]) == 0:
            questions.append(sent.replace(entity[0],wh_word) + '?')
            continue
```

#Install Pre-trained Amharic Dataset

```
In [91]: from datasets import load_dataset
dataset = load_dataset("masakhaner", 'amh')
```

```
for question_part in nondisc_sentences:
    s = " "
    sentence = question_part
    text = nltk.word_tokenize(question_part)
    if(text[0] == ' '):
        question_part = question_part[5:]
        s = " "

    elif(text[0] == ' '):
        question_part = question_part[4:]
        s = " "

    q = generate_question(question_part, s)
    if(q != ""):
        questions.append([sentence,q])
    l = generate_one_word_questions(question_part)
    questions += [[sentence,i] for i in l]
print(len(questions))
for pair in questions:
    print("[ዓ.ነገር:] ",pair[0])
    print( "[ጥያቄ:] ",pair[1])
    print()
    print("-----X-----\n\n")
```

#sample output Amharic_question_Generation()

```
In [119]: Amharic_question_Generation()
```

[ዓ.ነገር]: የወይዘሮዋ ጉዳይ ከሱዳን መንግሥት ጋ የሚያገናኘው ነገር የለም።

[ጥያቄ]: የማን መንግሥት ጋ የሚያገናኘው ነገር የለም?

-----X-----

[ዓ.ነገር]: በሱዳን የሚሰራበት ሕገ መንግሥት የሀይማኖት ነፃነትን እንደሚፈቅድ ላረጋግጥ እፈልጋለሁ።

[ጥያቄ]: የማን ሕገ መንግሥት የሀይማኖት ነፃነትን እንደሚፈቅድ ላረጋግጥ እፈልጋለሁ?

-----X-----

[ዓ.ነገር]: በወይዘሮዋ ቤተሰብ ውስጥ ጥል ከተፈጠረ በኋላ የባል ቤተዘመዶች ጉዳዩን ለፍርድ ቤት አቅረቡት እዚህ ላይ መዳረሳቸው ያሳዝናሉ።

[ጥያቄ]: በማን ውስጥ ጥል ከተፈጠረ በኋላ የባል ቤተዘመዶች ጉዳዩን ለፍርድ ቤት አቅረቡት እዚህ ላይ መዳረሳቸው ያሳዝናል?

#Evaluation of the system output by human judgment

```
In [111]: # Syntactic Score and Fluency using Manual Evaluation

syntactic_score = [1,1,1,1,1,1,1,0,0,1,1,1,0,1,1,1,1,1,0,1,1,1,0,1,0,1,1,1,1,1,1,0,0,1,1,1,1,1,1,1,1,1,1,1,1,0,
                   1,1,1,1,1,0,1,1,1,1,1,1,0,0,1,1,1,1,1,0,1,1,1,0,1,0,1,1,1,1,1,1,1,0,1,1,0,1,0,1,1,1,1,1,1,
                   1,1,1,1,0,1,1,1,1,1,1,1,1,1,1,0,0,1,1,1,0,1,1,1,1,1,1,1,1,1,1,1,0,1,1,0,1,1,1,1,
                   1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,1,1]
fluency_score   = [1,1,1,1,1,1,1,0,0,1,1,1,0,1,0,1,1,1,0,1,0,1,1,1,1,1,1,1,0,0,0,0,1,0,1,1,1,1,1,0,0,1,1,1,
                   0,0,0,0,1,1,1,1,1,1,0,0,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,
                    1,1,1,0,1,1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1,1,0,0,1,1,1,1,1,1,1,1,1,1,
                   1,1,0,1,1,1,1,0,1,1,0,1,1,1,1,1,1,1,1,1,1,0,1,1,1,0,0,0,0,0,1,1]
print(len(syntactic_score))
print("syntactic:" ,sum(syntactic_score)/ len(syntactic_score))
print("fluency:" ,sum(fluency_score)/ sum(syntactic_score))

150
syntactic: 0.8266666666666667
fluency: 0.8870967741934484
```

#The Amharic Letters Are as follows

Order							Labialized				
1st	2nd	3rd	4th	5th	6th	7th					
ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ					
ለ	ሉ	ሊ	ላ	ሌ	ል	ሎ			ሏ		
ሐ	ሑ	ሒ	ሓ	ሔ	ሕ	ሖ					
መ	ሙ	ሚ	ማ	ሜ	ም	ሞ			ሟ		
ሠ	ሡ	ሢ	ሣ	ሤ	ሥ	ሦ					
ረ	ሩ	ሪ	ራ	ሬ	ር	ሮ			ሯ		
ሰ	ሱ	ሲ	ሳ	ሴ	ስ	ሶ			ሷ		
ሸ	ሹ	ሺ	ሻ	ሼ	ሽ	ሾ			ሿ		
ቀ	ቁ	ቂ	ቃ	ቄ	ቅ	ቆ	ቈ	ቊ	ቋ	ቌ	ቍ
በ	ቡ	ቢ	ባ	ቤ	ብ	ቦ			ቧ		
ተ	ቱ	ቲ	ታ	ቴ	ት	ቶ			ቷ		
ቸ	ቹ	ቺ	ቻ	ቼ	ች	ቾ			ቿ		
ኀ	ኁ	ኂ	ኃ	ኄ	ኅ	ኆ	ኈ	ኊ	ኋ	ኌ	ኍ
ነ	ኑ	ኒ	ና	ኔ	ን	ኖ			ኗ		
ኘ	ኙ	ኚ	ኛ	ኜ	ኝ	ኞ			ኟ		
አ	ኡ	ኢ	ኣ	ኤ	እ	ኦ					
ወ	ዉ	ዊ	ዋ	ዌ	ው	ዎ					
ዐ	ዑ	ዒ	ዓ	ዔ	ዕ	ዖ					
ከ	ኩ	ኪ	ካ	ኬ	ክ	ኮ	ኰ	ኲ	ኳ	ኴ	ኵ
ኸ	ኹ	ኺ	ኻ	ኼ	ኽ	ኾ					
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ			ዟ		
ዠ	ዡ	ዢ	ዣ	ዤ	ዥ	ዦ					
የ	ዩ	ዪ	ያ	ዬ	ይ	ዮ					
ገ	ጉ	ጊ	ጋ	ጌ	ግ	ጎ	ጐ	ጒ	ጓ	ጔ	ጕ
ደ	ዱ	ዲ	ዳ	ዴ	ድ	ዶ			ዷ		
ጀ	ጁ	ጂ	ጃ	ጄ	ጅ	ጆ					
ጠ	ጡ	ጢ	ጣ	ጤ	ጥ	ጦ			ጧ		
ጨ	ጩ	ጪ	ጫ	ጬ	ጭ	ጮ			ጯ		
ጸ	ጹ	ጺ	ጻ	ጼ	ጽ	ጾ			ጿ		
ፀ	ፁ	ፂ	ፃ	ፄ	ፅ	ፆ					
ጰ	ጱ	ጲ	ጳ	ጴ	ጵ	ጶ					
ፈ	ፉ	ፊ	ፋ	ፌ	ፍ	ፎ			ፏ		
ፐ	ፑ	ፒ	ፓ	ፔ	ፕ	ፖ					

ቨ	ቩ	ቪ	ቫ	ቬ	ቭ	ቮ