Harla Journals

Harla Journal of Engineering and Technology

Journal home page: https://journals.ddu.edu.et/index.php/HJET

Harla journal of Engineering and Technology

MASTER: A Machine Led Solution to Amharic Arithmetic Word Problem

[1]Andinet Assefa Bekele

[1]Department of Computer Science, School of Computing, Dire Dawa University Institute of Technology, Dire Dawa, Ethiopia

andbeyes@gmail.com

Abstract

Arithmetic Word problem solving is a challenging yet exciting task requiring an accurate understanding of a given problem text to solve the underlying arithmetic equation. Several research works have been proposed to solve arithmetic word problems in English and other languages. The Amharic language has fewer natural language processing (NLP) resources and has morphologically complex verbing with a unique sentence structure. Though the existing machine translation approaches have shown tremendous progress, translating the contextual meaning within a given sentence or phrase still needs to be completed. Moreover, arithmetic word problems contain contextual meanings which need to be interpreted precisely within the context of the language structure, which the current machine translation practices could not help much. Hence, a strategy based on the linguistic structure of the language to understand and solve a given Amharic arithmetic word problem is needed. By far, previous research has yet to be conducted in the area. This paper proposes a novel strategy using a schema-based machine-led solution to Amharic arithmetic word problems (AWP) as a step towards enabling comprehension in mathematics and teaching problem-solving for children in the elementary grades. The proposed Amharic AWP solver involves four stages to a complete solution: NLP task of preprocessing on a given problem text, simplification of the problem text into more straightforward sentences, knowledge representation using Amharic schema to understand the discourse structure in addition to extracting relevant concepts that led to a systematic solution and finally, generation of a natural language answer based on the propositions made under knowledge representation phase. The learning component in the proposed strategy uses schema production rules to define and incorporate a new schema that allows for handling new varieties of problems. To validate the proposed approach, a prototype has been developed using python, and experimental results have shown an accuracy of 88.81% on a large corpus of Amharic arithmetic word problems.

Keywords: Amharic Arithmetic Word Problem, Natural Language Answer, Natural Language Understanding, Amharic Schema

* Corresponding Author: Andinet , andbeyes@gmail.com

DOI:

 Manuscript received March, 2022; revised April, 2022; accepted June, 2022.

1. INTRODUCTION

The arithmetic word problems are a type of exercise in mathematics where the main background information on the problems is described in words rather than in arithmetic equations. As arithmetic word problems often involve a description of a real-world scenario, they are sometimes referred to as story problems and may vary in the wording and verbification used in the text [1]. Arithmetic word problem-solving benefits the learners with basic linguistic knowledge in conjunction with skills of basic math operations such as addition, subtraction, multiplication, and division. Solving arithmetic word problems is a complex task that involves reading comprehension (understanding the different parts of the problem given), retrieving, and building a math equation described in the story, and solving the equation. It is agreed that children can solve very well on math problems given in formulas, but they find it challenging when it comes to word problems using textual narration.

Because of this challenge, the subject of arithmetic word problem-solving often needs the teacher`s closer attention to students' activities so that students may get proper problem-solving assistance. Although the teacher`s guidance cannot be ignored, teachers cannot solve all types of problems and lack to supply a sufficient volume of arithmetic word problem-solving exercises that consider individual differences among the learners. Besides this, these days of pandemic where coronavirus (COVID-19) is a global challenge, schools are closed in most countries and students are at home on their own apart from their teacher`s physical guidance. This still worsens the challenge for students to get appropriate arithmetic word problem exercise to solve as per their needs. It is admitted that the use of computer models for solving arithmetic word problems can greatly assist students to get an opportunity to practice on a volume of problem-solving exercises as per their individual needs and hence improve their problem-solving ability [2]. Accordingly, quite a few research works have been proposed towards a machine-assisted solution to arithmetic word problems for the English language as well as for other few languages. Motivated by the aforementioned challenges, in this paper, I propose a novel approach using a schema-based model for **Ma**chine Led **S**olution to Amharic Ari**t**hm**e**tic Wo**r**d Problem (**MASTER**) that automatically solves arithmetic word problems for the Amharic language.

Amharic is a member of the Semitic language family having commonalities with Hebrew, Arabic, Aramaic, and Syriac languages. Next to Arabic, it is the most spoken Semitic language. It is one of the major languages of Ethiopia spoken by over 30 million people as a first language and is spoken as a second language by millions of Ethiopians for most Ethiopians are multilingual. Hence, it is the official working language of the Federal Democratic Republic of Ethiopia and thus has nationwide status. Outside Ethiopia, Amharic is the language of millions of emigrants across the world and is also spoken in Eritrea [3]. It is written in a script called Fidel or abugida that was adopted from the Ge'ez writing system, which is now extinct. It is one of the major languages used for instruction in elementary schools across Ethiopia. Amharic being one of the morphologically rich languages presents a challenge to the area of natural language processing because of complex inflectional and derivational verb morphology with a unique sentence structure [4]. Despite being one of the under-resourced languages in the field of linguistics and NLP, quite a few foundational studies on the Amharic language have been conducted up to this point. The Automatic Generation of Math Word Problem and Equation [1], POS tagging and Morphological analyzer [5], Spell checking, and Named Entity Recognition [6] are a few examples of research work on the Amharic language to date. These research works open several avenues to conduct further research works for the Amharic language. Basing these studies and with the mentioned motivation above, this paper presents the proposed Machine Led Solution to Amharic Arithmetic Word Problem using a schema-based model as a step towards enabling comprehension and learning problem-solving in mathematics for elementary level students. The contributions of this paper are:

→ *Introduce basic Amharic schema with keywords for solving arithmetic word problems to JOIN, SEPARATE, PART-PART-WHOLE, and COMPARE type of problems for the Amharic language.*

→ *Introduce a heuristics-based simplification strategy for Amharic arithmetic word problem text. This avoids complexity in sentence structuring by rewriting the same problem using a set of simpler sentences that would be easily understood and hence allows students to get a chance to try to solve on their own even before the machine solves the problem completely, and*

→ *Introduce schema expansion and problem rewriting rules to define a new Amharic schema to handle new varieties of problems.*

The rest of this paper is organized as follows. Section 2 discusses the related research works.

Section 3 discusses the preliminary concepts to the realm of this research work. Section 4 presents the proposed machine-led solution to Amharic arithmetic word problems. Section 5 presents the implementation of a software prototype. Section 6 discusses experimentation and finally, Section 7 concludes this study and draws future research direction.

2. RELATED WORKS

Given an arithmetic word problem, the goal of machine solvers is to produce an answer for a math operation given behind a problem text. The answer produced could be a natural language answer or just a calculation result of the arithmetic operation given in the problem text. By far, several research works have been proposed for solving arithmetic word problems mainly for the English language with varying strategies. The proposed strategies were based on specific sentence structures with predefined schema [7,8,14], statistical-based approach [9], neural network approach [9], and template-based shallow NLP approach [1], which devises NLP techniques on top of a predefined sentence structure using templates. Supap et.al [7], proposed a schema-based strategy for solving arithmetic word problems for the Thai language, and Limguda et.al. [14] have devised the same approach to solving arithmetic word problems for the German language. Schemas are predefined templates to show the interaction between entities in a given arithmetic problem text. Bakman in [8] proposed a strategy to solve arithmetic word problems with extraneous information. The proposed strategy made by Bakman could help to extend schemas to handle more information and can solve multi-step problems, unlike its predecessors. Recently, Kushman et al. [9] proposed a strategy that uses statistical analysis to solve arithmetic word problems. The authors handled more complex word problems that involve solving a set of simultaneous linear equations. On the other hand, Chiang et.al. [10] have proposed an approach to automatically solve arithmetic word problems by retrieving the semantic meaning of symbols used in the problem text.

The Amharic language has fewer NLP resources and has morphologically complex verbing with a unique sentence structure [5]. Hence, solving the Amharic arithmetic word problem needs a strategy that is aware of the complexity in verb morphology and the unique sentence structure used in the Amharic language. Recently Andinet A. [1] has proposed a hybrid approach that combines templates with shallow NLP techniques for automatic generation of Amharic math word problems and equations from a given sample problem text. The approach first forms a template for both the word problem and the equation that characterizes the problem from a given sample and then, an equation template is then used for later semantic aware instance generation and solved by Numpy[15]. However, the approach cannot be generalized to all types of problems for two main reasons. First, it is limited in scope that it does not address compare types of arithmetic word problems, which characterizes most of the comparison operations in math. Secondly, because the approach is based on a specific sentence structure that is made to fit a specific equation template and uses a shallow NLP technique, it is limited to generating valid answers for problem texts that use complex sentences, ambiguous wording, or verbification. In addition to addressing compare types of word problems, the proposed approach in this paper devises a problem simplification process for a robust understanding of the problem text even in the presence of complex sentences and ambiguity. Moreover, the proposed approach incorporates schema production to handle new kinds of AWP types in the context of word problems defined in the schema database. Generally, in this paper, an end-to-end strategy that uses predefined keywords based on basic Amharic schema with the application of NLP techniques for a robust understanding of the discourse in the problem text to generate a natural language answer is presented. The basic schema is used to drive context-based a new schema to handle new kinds of AWP problems. The approach can also be extended to a new class of problems just by introducing new schemas with their keywords.

3. PRELIMINARIES

In this section, several notions used in the remaining part of this paper are given.

3.1. Sentence Structure in the Amharic Language and Existing Amharic NLP Resources

Like sentence structure in English, Amharic sentence structure comprises of Noun Phrase and Verb phrases. The Verb phrase is made of Prepositional Phrase, Noun Phrase, and Verb. Unlike English, Amharic is a VERB-FINAL language in that the verb phrase in Amharic comes in the order of Prepositional phrase followed by Noun phrase and then the Verb at the end [11]. In other words, the Amharic language follows subject-object-verb (SOV) grammatical patterns as opposed to, for example, the English language which follows SVO grammatical rules. For instance, the Amharic equivalent of the sentence "Betsegaw bought five apples" is written as "በፀጋው (Betsegaw) አምስት አፕሎችን (amist kuasochin /five balls) ገዛ (geza/bought)". Here, the part-of-speech (POS) tags of individual words are used as inputs to check the validity of grammatical patterns and used to identify the type of AWP used in the sentence. Figure 3-1 shows the structure of the Amharic sentence with a complex sentence example.

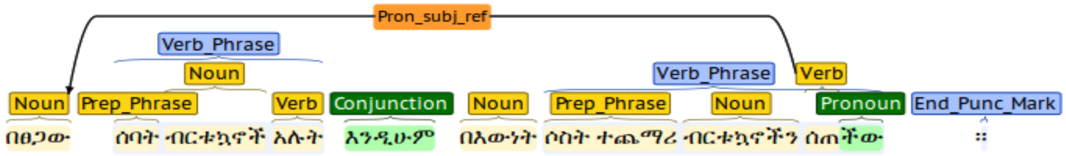

Fig. 1. Annotation sample showing structure of Amharic sentence using Brat

Solving Amharic AWP involves understanding the Amharic natural language sentence structure that makes up a given AWP text. Hence, several NLP tasks need to be accomplished for an accurate understanding of the given natural language problem text and to give a precise solution. The Amharic language has several useful NLP tools, including a sentence parser. In AWP, phrase parsing is a fundamental process for extracting meaning from natural language sentences. Sentence parsing, also known as syntactic parsing, is the process of determining how words can be put together to form proper sentences and what structural role each word plays in the sentence, as well as which phrases are subparts of which others. A sentence parser generates a parse structure that can be utilized in MASTER to determine the type of AWP employed in the given problem text, as well as the owners mentioned, objects with amounts, and object transfer words. Other than the Amharic sentence parser, the most complete morphological analysis tool used for NLP tasks in Amharic is HornMorpho [5]. Its performance was measured and reported to have a 95% efficiency based on the analysis made on two hundred randomly picked Amharic verbs. Analysis, segmentation, and generation of words can be carried out by the system. As a result, it can be used as a part of speech tagger (POS) and morphological analyzer for it can attach syntactic information of function words that are related to the word. The proposed strategy in this paper applies this reasonable NLP resource to realize the machine-led solver for Amharic arithmetic word problems.

3.2. Amharic Arithmetic Word Problems

Arithmetic word problems are used to present a real word story using textual narration for learning basic arithmetic operations and gaining basic linguistic knowledge. What makes learning to solve arithmetic word problems interesting is that it presents normal arithmetic operations in terms of the learner's daily life situation. The problem text of arithmetic word problems can contain one or more sentences with varying discourse structures. Each sentence in the problem text in Amharic follows the Amharic sentence structure. Example 3-1 provides a sample arithmetic word problem in Amharic with English translation alongside.

Example 3-1: [Amharic Arithmetic Word Problem]

"በፀጋው ሰባት ብርቱኳኖች አሉት፤ እንዲሁም በእውነት 3 ተጨማሪ ብርቱኳኖችን ሰጠችው፡፡ በፀጋው በአጠቃላይ ስንት ብርቱኳኖች ይኖሩታል?"
["Betsegaw has seven oranges. Bewunet gave him 3 more oranges. How many oranges does Betsegaw have now?".]

In this typical example, the equation that involves the arithmetic operation addition is described using words. Solving such a kind of arithmetic word problem requires a comprehension skill of identifying the discourse in the story and retrieving the equation laid behind. There are four types of arithmetic word problems, which are common in elementary school's mathematics subject i.e., JOIN, SEPARATE, PART-PART-WHOLE, and COMPARE, that characterize most of the addition and subtraction problems [1]. The simplest of all these is COMPARE problems, which involves a comparison of the counts of elements in given sets. Table 3-1 presents types of Amharic arithmetic word problems with example translation in English alongside.

Table 1. Arithmetic word Problems in Amharic with English translation

Problem Type	Example
Join	*ፌበን* 5 *እርሳሶች አላት፡፡ ስንት እርሳሶችን ብትጨምር በአጠቃላይ* 7 *እርሳሶች ይኖራታል*? [*Feben had* 5 *pencils*. *How many more pencils does she has to put with them so she has* 7 *pencils altogether*?]
Separate	*ሩት* 12 *ሂጦች አሉአት፡፡* 5*ቱን ለርብቃ ሰጠች፡፡ ሩት ስንት ሂጦች ይቀሩአታል*? [*Ruth has* 12 *dimes*. *She gave* 5 *dimes to Rebecca*. *How many dimes does Ruth has now*?]
Part – Part Whole	6 *ወንዶችና* 7 *ሴቶች በእጅ ኳስ ቡድን ውስጥ አሉ፡፡ በአጠቃላይ ቡድኑ ስንት ልጆችን ይዞአል*? [*There are* 6 *boys* and 8 *girls on the volleyball team*. *How many children are in the volleyball team*?]
Compare	*ከበደ* 11 *ኳሶች አሉት፡፡ ተሰማ ከከበደ በ*4 *የሚበልጡ ኳሶች አሉት፡፡ ተሰማ ስንት ኳሶች አሉት*? [*Kebede has* 11 *balls*. *Tesema has* 4 *more balls than Kebede*. *How many balls does Tesema has*?]

4. MASTER AMHARIC ARITHMETIC WORD PROBLEM SOLVER

The system architecture of the proposed MASTER Amharic arithmetic word problem solver is shown in Figure 4-1. The solving process in the system involves four main phases: preprocessing, simplification, Knowledge representation, and natural language answer generation. The preprocessing task mainly focuses on segmentation to identify individual sentences that make up the problem text, tokenization to identify individual tokens in each sentence of the problem text, and POS tagging to identify the part of speech each token in each sentence belong. This is followed by the problem simplification phase, which involves the normalization of numeral values and the conversion of complex sentences into simpler sentences. Then, the preprocessed problem text containing a set of simplified sentences with their linguistic information is passed on for the knowledge representation phase. Amharic schemas together with keywords are used for knowledge representation. Concept extraction from each sentence to make up a concept vector is the first task in knowledge representation which is followed by schema instantiation to identify a specific schema to apply for a given problem. If no matching schema is found in the schema list then the knowledge representation step will be done by a new schema provided by the schema producer component and later based on user rating the schema producer component incorporates the new schema into the schema database.

Finally, in the natural language answer generation phase a set of heuristics are applied to compile the final answer. The details of these processes in the system is presented in the coming sections.

4.1. Preprocessing of AWP

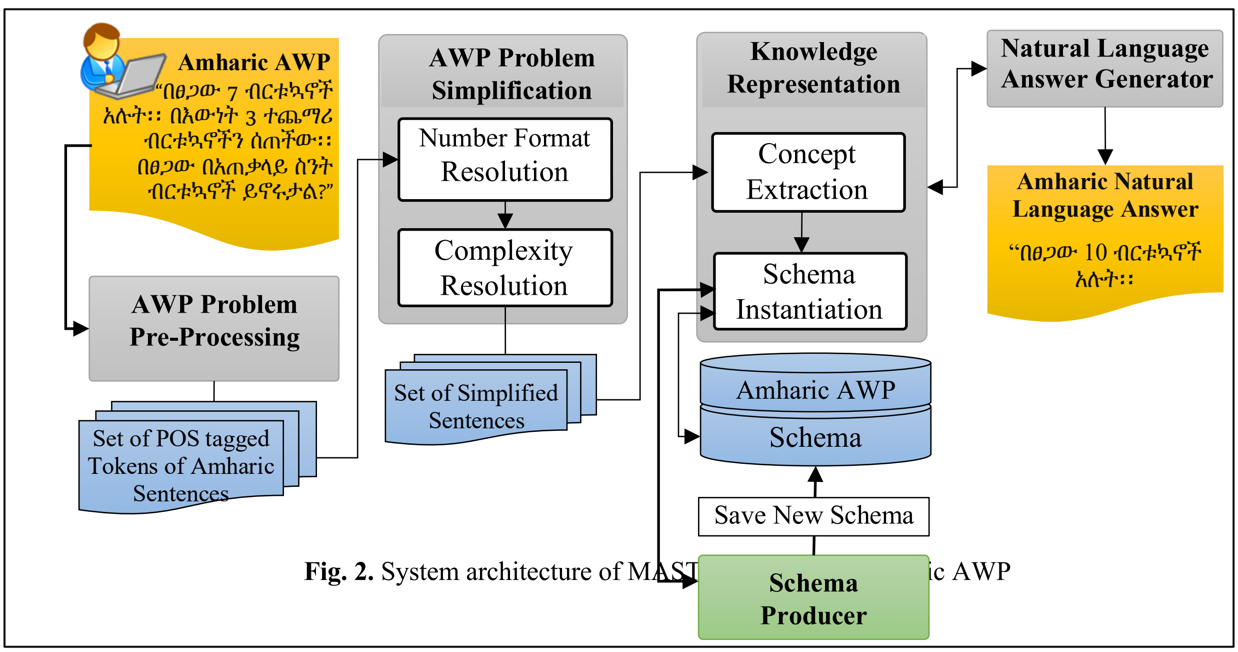

Figure 4-1: MASTER AMHARIC ARITHMETIC WORD PROBLEM SOLVER

The preprocessing phase involves NLP tasks of segmentation, tokenization, and POS tagging. This phase transforms the given Amharic arithmetic word problem text into a set of POS tagged tokens of Amharic sentences for simplification in the next phase. HornMorpho [5] the Amharic distribution is a morphological analyzer is used for the preprocessing task.

Example 4-2: [Segmentation, Tokenization, and POS tagging]

Considering a problem text given by Example 3-1, the segmentation task splits the problem text at an end punctuation mark "።" to give a set of sentences $S = \{S_1, S_2\}$, where: -

S_1 = {በፀጋው ሰባት ብርቱኳኖች አሉት እንዲሁም በእውነት 3 ተጨማሪ ብርቱኳኖችን ሰጠችው።} {Betsegaw has seven oranges and Bewunet gave him 3 more oranges. } , and

S_2 = {በፀጋው በአጠቃላይ ስንት ብርቱኳኖች ይኖሩታል? } {How many oranges does Betsegaw has now? }

Then, the tokenizer produces a set of sentences set $S_T = \{S_{T1}, S_{T2}\}$, where each element contains a collection of tokens of each sentence in S as given below: -

S_{T1} = {በፀጋው, ሰባት, ብርቱኳኖች, አሉት, እንዲሁም , በእውነት, 3, ተጨማሪ, ብርቱኳኖችን, ሰጠችው, ።} {Betsegaw, has, seven, oranges, and Bewunet , gave, him, 3, more oranges, . } and

S_{T2} = {በፀጋው, በአጠቃላይ, ስንት, ብርቱኳኖች, ይኖሩታል, ? } {How, many, oranges, does, Betsgaw, has, now, ? }

Finally, the part of the speech tagger produces a set of POS tagged sentences $S_{TG} = \{S_{TG1}, S_{TG2}\}$ as given below: -

S_{TG1} = {"(በፀጋው, N), (ሰባት, NUM), (ብርቱኳኖች, N), (አሉት, V), (እንዲሁም, CNJ) (በእውነት, N), (3, NUM), (ተጨማሪ, N), (ብርቱኳኖችን, N), (ሰጠችው, V), (።)} [{(Betsegaw, N), (has, V), (seven, NUM), (oranges, N) (and, CNJ) (Bewunet, N), (gave, V), (him, PRP), (3, NUM), (more, N), (oranges, N)}]

S_{TG2} = {(በፀጋው, N), (በአጠቃላይ, ADJ), (ስንት, JJ), (ብርቱኳኖች, N), (ይኖሩታል, V), ('? ')} [{(How, WRB), (many, JJ), (oranges, N), (does, VBZ), (Betsegaw, N), (has, V), (now, RB), (? ,)}]

4.2. Problem simplification

At the end of this phase, a problem text as a set of simplified sentences would be available for knowledge representation in the next phase. The simplification process involves number format resolution and sentence complexity resolution on a given preprocessed Amharic arithmetic word problem as explained in detail in the coming sections.

4.2.1 Number format resolution (NFR)

The numeric quantities in a given Amharic arithmetic word problem text can be given in symbolic, textual, or mixed format. Identification and normalization of numeral values to a commonly agreed format i.e., the symbolic format is the aim of the number format resolution task. For example, in the problem text "በፀጋው ሰባት ብርቱኳኖች አሉት። በእውነት 3 ተጨማሪ ብርቱኳኖችን ሰጠችው። በፀጋው በአጠቃላይ ስንት ብርቱኳኖች ይኖሩታል? ", the word "ሰባት" is used instead of the number "7". Hence, the number resolution process replaces numeric values in the problem text expressed in words with their corresponding Arabic symbolic representations.

4.2.2 Sentence complexity resolution (SCR)

Taking NFR resolved POS tagged set of sentences representing a problem text, the sentence complexity resolution process rewrites all those complex sentences into simple sentences i.e., a sentence with only one verb. Complex sentences in Amharic are made of two simple sentences connected with conjunction words like "እንዲሁም" or "እና" ("and"), "ግን" ("but"), and "ቢሆንና…ባይሆን" ("if…else"). In addition to dealing with complex sentences, pronouns would be substituted with the subject it refers to. The overall process to resolve the complexity of sentences devises the following Amharic sentence structure-aware heuristics:

SCR_HEUR 1. If there is a conjunction word in a given sentence, split the sentence into two sentences: SS_1 *and* SS_2.

SCR_HEUR 2. If SS_1 has no verb, then no need for splitting. If we look at the sentence "Bewunet and Betsegaw have 7 oranges altogether", then it makes it illogical to split this sentence as "Bewunet has 7 oranges" and "Betsegaw has 7 oranges."

SCR_HEUR 3. Split both SS_1 *and* SS_2 into two strings: *Noun-Phrase (NP) and Verb-Phrase (VP).*

SCR_HEUR 4. Split *Verb-Phrase* into three strings *Prepositional Phrase (PP), Noun-Phrase (NP), and Verb (V)*.

SCR_HEUR 5. If the *verb* in the *verb phrase* ends with one of the Amharic letters *"ችው", "ጣት"*, or is found in the schema keyword list, then insert a concatenated string ***[ለ(to) + Noun-Phrase]*** before the verb.

SCR_HEUR 6. Concatenate strings to build two simple sentences and append an end punctuation mark **"::"** at the end of each sentence.

Heuristic steps 1 to 4 are the splitting phase and steps 5 and 6 are the integration phase in SCR. After SCR step every sentence in the new problem has exactly one verb. Pseudocode-1 presents the algorithm for NFR and SCR for sentence simplification.

Example 4-2: [SCR Illustration]

Referring to a POS tagged set of sentences from Example 4-1, the first sentence STG1 = "በፀጋው 7 ብርቱኳኖች አሉት እንዲሁም በእውነት 3 ተጨማሪ ብርቱኳኖችን ሰጠችው::" [Betsegaw has 7 oranges and Bewunet gave him 3 more oranges] is a complex sentence containing two simple sentences connected with a conjunction word "እንዲሁም" ["and"]. Hence, splitting this sentence gives two simple sentences "በፀጋው 7 ብርቱኳኖች አሉት::" ["Betsegaw has 7 oranges.] and "በእውነት 3 ተጨማሪ ብርቱኳኖችን ሰጠችው::" ["Bewunet gave him 3 more oranges."]. Table 4-2 shows the whole process of splitting a complex sentence for this typical example.

Again, the second simple sentence contains a pronoun "ችው" [him] as part of the verb "ሰጠችው" which refers to the subject "በፀጋው" [Betsegaw] in the first simple sentence. Adding a subject prefixed with “ለ” [to] to make a word “ለበፀጋው” [to Betsegaw] and appending it before the verb “ሰጠችው” [gave] modifies the sentence to become “በእውነት 3 ተጨማሪ ብርቱኳኖችን ለበፀጋው ሰጠችው፡፡” [“Bewunet gave 3 more oranges to Betsegaw.”]. Table 4-2 shows the splitting and integration phases for this typical example and the overall process of applying SCR on a set of sentences S from Example 4-1 yielding a set of simplified sentences $S_{SIM} = \{S_{SIM1}, S_{SIM2}, S_{SIM3}\}$, where: -

S_{SIM1} = {(በፀጋው, N), (7, NUM), (ብርቱኳኖች, N), (አሉት, V)}, [{(Betsegaw, N), (has, V), (7, NUM), (oranges, N)],

S_{SIM2} = {(በእውነት, N), (3, NUM), (ተጨማሪ, N), (ብርቱኳኖችን, N), (ሰጠችው, V), [(Bewunet, N), (gave, V), (him, PRP), (3, NUM), (more, N), (oranges, N) }],

S_{SIM3} = {(በፀጋው, N), (በአጠቃላይ, V), (ስንት, JJ), (ብርቱኳኖች, N), (ይኖሩታል, V), (‘?’)}
[{(How, WRB), (many, JJ), (oranges, N), (does, V), (Betsegaw, N), (has, V), now, RB), (?)}]

Table 4-1 SCR process for sentence S_{TG1} of POS tagged sentence set S_{TG} from Example 4-1

COMPLEX SENTENCE	**በፀጋው 7 ብርቱኳኖች አሉት እንዲሁም በእውነት 3 ተጨማሪ ብርቱኳኖችን ሰጠችው፡፡** [*Betsegaw has 7 oranges and Bewenet gave him 3 more oranges.*]	**SCR RULE**

Pseudo Code 1: Algorithm for Amharic Arithmetic Word Problem Simplification

```
Inputs:
   AWP[m][n] String; // Vector containing pre-processed set of problem sentences
Intermediates:
   Flag   Boolean;
   genNumber[] StringArray;  //genNumber is a text representation of DetNumber
   DetNumber Integer; //DetNumber is a number converted from text to number
   Temp_AWP[]   StringArray;   //temporary string array
   Norm_AWP   String //Normalized Amharic AWP containing all numerals given in Arabic symbols
Result:
   Simplified_AWP  StringArray;
Start:
     Flag ← 0;
     For p ← 0 to m
       For k ← 0 to n
         IF  isNumber(AWP[m][n]) = TRUE Then
                 Push(AWP[m][n])
                 Flag ← 1
         Else
              IF flag ← 1 Then
               For t ← 0 to sizeOf(stack)
                  genNumber[t] ← pop();
               End For
                 detNumber ← textToNumber(genNum);
                 Norm_AWP ← replace(AWP[m][n], detNumber, n, counter + sizeofStack);
              Else
       End For
     End For
        For l ← 0 to sizeOf(Norm_AWP)
            IF isComplex(NormAWP[l]) = TRUE  THEN
                 Temp_AWP ← resolveComplexity(Norm_AWP[l])
                 Simplified_AWP.Append(Temp_AWP)
            Else
                 Simplified_AWP.Append( Norm_AWP[l]))
        END For
     Return Simplified_AWP     // Return a simplified Amharic AWP
End
```

4.3. Knowledge Representation

This is the phase where the core understanding process is made on a given problem text that comes down from the preprocessing and simplification phases. Here, concept extraction and schema representation are the two main steps for knowledge representation, which leads to the final natural language answer compilation.

4.2.3 Amharic Schemas

Schemas are predefined templates for knowledge representation by which the interaction between entities in a given Amharic problem text is depicted. The understanding process is said to be successful if it ends with matching a given Amharic arithmetic word problem to a specific schema stored in the database. Schema matching is mainly based on occurrences of one or more schema keywords and the notion of object transfer between owners in the problem text. The list of Amharic schemas used for knowledge representation in MASTER is listed in Table 4-1 and their associated keywords are given in Table 4-2. The Stemming NLP task is applied to the schema keywords and they are in their base form. Example 4-3 elaborates more on a typical scenario.

Example 4-3: [Knowledge Representation]

Referring to a simplified problem text presented as a set of sentences from Example 4-2, the element S_{SIM2} contains a keyword "ሰጠችው" [gave], which shows the transfer of objects ("ብርቱኳኖችን"[Oranges]) between owners ("በፀጋው"[Betsegaw] and "በእውነት"[Bewunet]). This piece of information would trigger JOIN Type Schema and the interaction is described as follow:

→ (ባለቤት1) (X) (ቁሶች) አሉት ፡ "(በፀጋው) (7) (ብርቱኳኖች) አሉት።
((owner1)has (X)(objects): Betsegaw has 7 oranges.)
→ (ባለቤት2)(Y) (ቁሶችን) ለ(ባለቤት1)አስተላለፈ ፡ (በእውነት) (3) ተጨማሪ (ብርቱኳኖችን) ለ(በፀጋው) ሰጠችው።
((owner2)(transfer)(Y)(objects) to (Owner1) ፡ "Bewunet Transfer 3 more oranges to Betsegaw")
→ (ባለቤት1) $(X+Y)$ (ቁሶች) አሉት፡"(በፀጋው) $(7 + 3)$ (ብርቱኳኖች) አሉት።
(owner1)has $(X+Y)$(objects): "Betsegaw (7+3) oranges")
→ (ባለቤት2) $(Z-Y)$ (ቁሶች) አሉት ፡ "(በእውነት) $(Z - 3)$ (ብርቱኳኖች) አሉት።
(owner2) has $(Z-Y)$(objects): "Bewunet has (Z-3) oranges"

As shown by the above scenario, every element sentence in a given set of sentences that make up the problem text is examined sequentially until a keyword is encountered. The keyword is then used to relate the problem to a specific schema in the database. For example, the word "ሰጠችው" [gave] is one of the words listed as a keyword in JOIN type schema. In the first sentence, Owner1 is identified to be "በፀጋው", the Object is "ብርቱኳኖች" and X =7. The second sentence provides

Owner2 as “በእውነት”, Y=3, and the object is transferred to Owner1. The third sentence, which is a question sentence, contains Owner1 which is “በፀጋው” and the Object “ብርቱኳኖች” which directs to the proposition $(owner1)\ has\ (X + Y)\ (objects)$ as a template for the final natural language answer compilation. To do so and come to successful knowledge representation, concepts like Owners, Objects, Numerals, etc. in the simplified problem text need to be first extracted. Then these concepts would be analyzed to initiate a particular schema stored in the database. The coming sections explain these processes in detail.

Table 4-1 Amharic Schema with translation in English

Schema Name	Schema	Result
Join	(ባለቤት1) (X) (ቁሶች) አሉት:: (ባለቤት2) (Y) (ቁሶች) አሉት፡፡ (Z) (ቁሶች) ከ(ባለቤት2)ወደ (ባለቤት1)ተላለፈ፡፡ (Owner1 has X objects. Owner2 has Y objects. Z objects were transferred from Owner2 to Owner1)	(ባለቤት1) $(X + Z)$ (ቁሶች) አሉት፡፡ (ባለቤት2) $(Y - Z)$ (ቁሶች) አሉት፡፡ (Owner1 has (X+Z) objects. Owner2 has (Y-Z) objects.)
Separate	(ባለቤት1) (X) (ቁሶች) አሉት:: (ባለቤት2) (Y) (ቁሶች) አሉት፡፡ (Z) (ቁሶች) ከ(ባለቤት1)ወደ (ባለቤት2)ተላለፈ፡፡Owner1 has X objects. Owner2 has Y objects. Z objects were transferred from Owner1 to Owner2	(ባለቤት1) $(X - Z)$ (ቁሶች) አሉት፡፡ (ባለቤት2) $(X + Z)$ (ቁሶች) አሉት፡፡ Owner1 has (X-Z) objects. Owner2 has (X+Z) objects.
Part-Part Whole	(ባለቤት1) (X) (ቁሶች) አሉት:: (ባለቤት2) (Y) (ቁሶች) አሉት፡፡ በጋራ (Z)ቁሶች አሉአቸው፡፡Owner1 had X objects. Owner2 had Y objects. Together, they have Z objects.	Z = X + Y
Compare	(ባለቤት1) (X) (ቁሶች) አሉት:: (ባለቤት2)(ከባለቤት1) በ(Y)የበለጡ ቁሶች አሉት፡፡ Owner1 had X objects. Owner2 had Y objects more than Owner1.	Owner2 has (X+Y) objects (ባለቤት2) $(X + Y)$ (ቁሶች) አሉት፡፡

Table 4-2 Amharic Schema keywords with translation in English

Schema	Keywords in their root form
Join	ሸጠ፣አካፈለ፣ጫነ፣ሰጠ፣ጨመረ፣አሰቀመጠ፣ተከለ፣ደመረ (Sell, distribute, load, give, put, place, plant, add)
Separate	ተዋሰ፣ሰረቀ፣አጠፋ፣አዋለ፣አስወገደ፣ቀነሰ፣ ወሰደ፣አገኘ፣አነሳ፣ገዛ (borrow, steal more, destroy, spend, remove, decrease, take from, get, pick, buy)
Part-Part Whole	በድምሩ ፣ በአጠቃላይ፣ በአንድ ላይ፣ በአንድነት (Combined, together, in all)
Compare	(ይልቅ ጥቂት፣ ይልቅ ያጠረ ፣ ይልቅ በለጠ፣ ይልቅ ጨመረ፣ ይልቅ ረዘመ፣ ይልቅ ያነሰ (Fewer than, shorter than, more than, taller than, longer than, less than)

4.2.4 Concept Extraction

A sentence set produced as a result of the SCR phase is input to the concept extraction phase. Here in this phase, a concept vector is generated which contains a set of concepts extracted from each simplified sentence as formally given by Definition 4-1. The set of concepts in the concept vector is used for schema instantiation which directs to a final answer.

Definition 4-1: [Concept Vector]

Given S_{SIM}, which is a set of simplified sentences from the SCR phase, its concept vector denoted as V_C, is a set of concept sets $\{V_{C1}, V_{C2}, V_{C3}, \ldots, V_{Cn}\}$ extracted from each sentence and each V_{Ci} contains a set of concepts $\{SentenceType,\ Keyword,\ Owner,\ E_{Name},\ E_{Quantity},\ E_{Transfer}\}$, where:

$$- SentenceType$$
$$= \begin{cases} OW, \textit{if it contains any of Amhraic Ownership terms like } \text{አለው፣አላት አላቸው ነበረው ነበራት ነበራቸው} \\ TR, \textit{if the sentence contains any of Amharic object trnasfering terms like } \text{ሰጠ፣ሰጠቸው፣አዋሰ፣አዋሰቸው} \\ QU, \textit{if the sentence denotes Amharic Question} \\ NULL, Otherwise \end{cases}$$

$$- Key = \begin{cases} \{\textit{a set of keywords}\}\ \textit{listed in the problem text stemmed to their base form} \\ NULL, Otherwise \end{cases}$$

$$- Owner = \begin{cases} Owner1 \text{ and } Owner2 \\ NULL, Otherwise \end{cases}$$

$$- E_{Name}\ \textit{is Entity name, N otherwise}$$

$$- E_{Quantity}\ \textit{is entity quantity,}$$

$$- E_{Transfer}\ \textit{is number of entities transfered}$$

Example 4-4: [Concept Vector]

Referring to a set of sentences S_{SIM} as a result of the simplification process, its concept vector is extracted as given in **Table 4-3**.

Table 4-3 Concepts extracted from sentence set S_{SIM} taken from **Example 4-2**.

Sentence	Sentence Type	Keyword	Owner		E_{Name}	$E_{Quantity}$	$E_{Transfer}$
			Owner1	Owner2			
S_{SIM1}	OW	NULL	በፀጋው [Betsegaw]	NULL	ብርቱኳኖች[Oranges]	7	0
S_{SIM2}	TR	ሰጠ [Gave]	በፀጋው [Betsegaw]	በእውነት [Bewunet]	ብርቱኳኖች [Oranges]	X	3
S_{SIM3}	QU	አጠቃላይ	በፀጋው [Betsegaw]	NULL	NULL	0	0

4.2.5 Schema Instantiation

Suppose that the concept extraction process has been completed successfully, i.e., all relevant concepts are identified for each sentence that makes up the problem text, in the schema instantiation process associations are made between the problem text and a specific schema in the database. The association is made based on the keyword used and the type sentence i.e., OW, TR, or QU according to **Definition 4-1**. As a result of schema instantiation, a set containing instances is produced and **MASTER** applies heuristics to select one as a natural language answer to a given arithmetic word problem. Besides this, the list of instances generated can be used to aid children to understand the problem and the way the answer is compiled.

To elaborate, consider a concept set generated for S_{SIM2} in Example 4-4. There appears a keyword "ሰጠ" [gave] which belongs to the keyword list in JOIN type schema and the sentence type is identified Transfer (TR) type. Hence, this triggers the instantiation of the Join Type schema. Table 4-4 gives a schema instantiation produced by MASTER for this typical example.

Table 4-4: MASTER output for schema instantiation for concept vector given in **Example 4-4**

Schema/Join	Result	Instances
(ባለቤት1) (X) (ቁሶች) አሉት:: Owner1 has X objects. **(ባለቤት2) (Y) (ቁሶች) አሉት::** Owner2 has Y objects. **(Z) (ቁሶች) ከ(ባለቤት2)ወደ (ባለቤት1)ተላለፈ::** Z objects were transferred from Owner2 to Owner1	Owner1 has (X+Z) objects. (ባለቤት1) (X + Z) (ቁሶች) አሉት:: Owner2 has (Y-Z) objects. (ባለቤት2) (Y − Z) (ቁሶች) አሉት::	በፀጋው (7 + 3) ብርቱኳኖች አሉት:: Betsegaw has (7+3) Oranges. በእውነት (Y − 3) ብርቱኳኖች አሉአት:: Bewunet has (Y-3) Objects.

4.3 Schema Production (SP)

The schema-producing module makes MASTER cope up with a new kind of problem that cannot be solved with the existing schema. It employs a rule-based approach to produce a new schema to aid the schema instantiation process. The schema production process either performs expansion of the basic schema to match the problem text or rewrites the problem text to match the existing basic schema. To do so, the schema producer first identifies the problem type and makes use of the context of the given problem in conjunction with the basic schema given in Table 4-1 to produce a new schema and its associated key.

SP Rule 1 [Rule of Schema Expansion]: Given a simplified Amharic AWP problem 'P' containing n number of receiving Owners with M Objects, and Q number of donating owners with R number of objects, the rule of expansion first disintegrates sentences into a set of sentences containing receiving owners S_R and a set of sentences containing donating owners S_D. Then the schema instantiation process applies the basic schema recursively. Example 4-5 explains the rule of schema expansion.

Example 4-5: [Schema Expansion]

Given a new kind of simplified Amharic AWP: -

"በፀጋው ሰባት 7 ብርቱኳኖች እና 5 ሙዞች አሉት። በእውነት 3 ተጨማሪ ብርቱኳኖችንና 2 ሙዞችን ለበፀጋው ሰጠችው::
በፀጋው በአጠቃላይ ስንት ብርቱኳኖችና ሙዞች ይኖሩታል?"

The new disintegrated sentences are: -

"በፀጋው ሰባት 7 ብርቱኳኖች አሉት። "በፀጋው 5 ሙዞች አሉት። በእውነት 3 ተጨማሪ ብርቱኳኖችን ለበፀጋው ሰጠችው::
በእውነት ተጨማሪ 2 ሙዞችን ለበፀጋው ሰጠችው:: በፀጋው በአጠቃላይ ስንት ብርቱኳኖች ይኖሩታል?"
በፀጋው በአጠቃላይ ስንት ሙዞች ይኖሩታል?"

The resulting schema is:

(ባለቤት1) (X) (ቁሶች) እና (Y) (ቁሶች) አሉት:: (ባለቤት2) (M) (ቁሶች) እና (N) (ቁሶች) አሉት፡፡
(A) (ቁሶች) እና (B) (ቁሶች) ከ(ባለቤት 2) ወደ (ባለቤት1)ተላለፈ፡፡
(Owner1 has X and Y objects. Owner2 has M and N objects. A and B objects were transferred from Owner2 to Owner1)

And the result according to the new schema is:

(ባለቤት1) $(X + A)$ (ቁሶች) እና $(Y + B)$አሉት፡፡ (ባለቤት2) $(M - A)$ (ቁሶች)እና $(N - B)$(ቁሶች) አሉት፡፡ (Owner1 has (X+A) and (Y+B) objects. Owner2 has (M-A) and (N-B) objects.

SP Rule 2 *[Rule of Problem Rewriting]:* the rule of problem rewriting, rewrites a given problem to match the basic schema.

Example 4-6: [Problem Rewriting]

Considering a join type problem given in Table 3-1:

ፌበን 5 እርሳሶች አላት፡፡ ስንት እርሳሶችን ብትጨምር በአጠቃላይ 7 እርሳሶች ይኖራታል? [Feben had 5 pencils. How many more pencils does she have to put with them so she has 7 pencils altogether?]

The Rewriting rule, rewrite the sentence as: -

ፌበን 5 እርሳሶች አላት፡፡ ያልታወቀ – ሰው ለፌበን X እርሳሶች ሰጣት፡፡ ፌበን አሁን በአጠቃላይ 7 እርሳሶች አላት። የ X ዋጋ ስንት ነው?[Feben had 5 pencils. UNKNOWN gave X Pencils to Feben. Feben has 7 pencils now. What is the value of X?

This results in a matching basic *Joint Type* schema, *Owner1 has X objects. Owner2 has Y objects. Z objects were transferred from Owner2 to Owner1).*

4.4 Answer Generator

After a set of instances are identified, the answer generation would use heuristics to select one of the instances as an answer to the problem given. Once, a specific instance is selected, the arithmetic operation included in it is converted to the equation of the form $R = [X\ \theta\ Y]$, where θ is the arithmetic operator [+ or -]. For example, the arithmetic operation in the first instance would be converted to $R = 7 + 3$. Finally, the result of the arithmetic operation is substituted instead of the arithmetic operation in the given instance, which automatically is a natural language answer for the problem given. Below is a list of heuristics used for instance selection and equation solving:

→ Use the owner mentioned in the question sentence to filter out candidate instances among the given set of instances. [*Result:* candidate instances containing the owner mentioned in the question sentence.]
→ Use the keyword mentioned and the type of sentence identified to specify an instance from the subset selected in step 1. [*Result*: an instance as candidate answer]

→ Convert the arithmetic operation in the candidate answer. If there is more than one instance that refers to the same owner, a combination of all the arithmetic operations is taken. [*Result*: Equation in the form $R = [X\ \theta\ Y]$]

→ Solve the equation, substitute in place of the arithmetic operation in the candidate answer, and return it. *[Result: Natural Language Answer]*

To finalize with elaboration, MASTER selects the instance "በፀጋው (7 + 3) ብርቱኳኖች አሉት፡፡" [Betsegaw has (7+3) oranges.] applying the procedures and returns a natural language answer "በፀጋው 10 ብርቱኳኖች አሉት፡፡" [Betsegaw has 10 oranges.]. The following sections present prototype and experimental results.

5 MASTER Amharic AWP Solver Prototype Implementation

To demonstrate the validity of the proposed strategy of the machine-led solution to the Amharic arithmetic word problem, a prototype was developed using python and the source code can be found with GitHub link "*https://github.com/andbeyes/MASTER-Amharic-Math-Word-Problem-Solver*". The prototype was used as a testing platform to test the performance of the proposed strategy. As part of the implementation, a command-based interface was developed to facilitate the interaction between the user and the proposed system.

As shown by the sample output of MASTER in **Figure 5-1**, the system takes the Amharic arithmetic word problem as input and returns a natural language solution.

```
>>> word_problem="በፀጋው 12 ብርቱኳኖች አሉት። በአውነት 3 ተጨማሪ ብርቱኳኖችን ለበፀጋው ሰጠችው። በፀጋው በአጠቃላይ ስንት ብርቱኳኖች ይኖሩታል?"
>>> ans=masterSolveAWP(word_problem)
  በፀጋው 12 ብርቱኳኖች አሉት::
  በአውነት 3 ብርቱኳኖች ሰጠችው::
  በፀጋው በአጠቃላይ ስንት ብርቱኳኖች ይኖሩታል?
  NLP Answer= በፀጋው 15 ብርቱኳኖች አሉት:: [JOIN TYPE-0 AWP]::
```

```
>>> word_problem="ሩት 12 ጌጦች አሉአት:: ለርብቃ 5 ጌጦችን ሰጠች:: ሩት ስንት ጌጦች ይቀሩአታል?"
>>> ans=masterSolveAWP(word_problem)
  ሩት 12 ጌጦች አሉአት::
  ለርብቃ 5 ጌጦችን ሰጠች::
  ሩት ስንት ጌጦች ይቀሩአታል?
  NLP Answer= ሩት 7 ጌጦች አሉአት:: [SEPARATE TYPE-0 AWP]::
```

```
>>> AWP="6 ወንዶችና  7 ሴቶች በአጅ ኳስ ቡድን ውስጥ አሉ። በአጠቃላይ ቡድኑ ስንት ልጆችን ይዞአል?"
>>> NLP_Res=masterSolveAWP(AWP)
>>> print(NLP_Res)
NLP Answer= ቡድኑ 13 ልጆችን ይዞአል::
```

```
>>> AWP="ከበደ  11 ኳሶች አሉት። ተሰማ   ከከበደ በ 4 የሚበልጡ ኳሶች  አሉት።  ተሰማ   ስንት ኳሶች አሉት።
>>> NLP_Res=masterSolveAWP(AWP)
>>> print(NLP_Res)
NLP Answer= ተሰማ 15 ኳሶች አሉት::
```

Figure 5-1 Screenshot of MASTER Amharic AWP solver prototype sample output using python IDLE command line interface

6 Experimental Results

Three experiments were conducted; the first to assess the accuracy of MASTER`s performance on sentence complexity resolution, then the second experiment focused on the accuracy of the proposed approach on schema instantiation, and the third experiment was conducted to check the accuracy of the answer generated by MASTER compared with a human answer. The experiments were conducted on a data set of 457 Amharic arithmetic word problems collected from Math textbooks and some generated by a human. The data set is categorized into two with varying difficulties. The first set (ADS1) contains 237 problem texts and the second (ADS2) contains 220 problem texts.

6.2 Sentence Complexity Resolution

As per the discussion made on the importance of sentence complexity resolution in section 4.2.2 above, the accuracy of the answer generated by MASTER is merely dependent on its understanding of the given problem text. This is possible only if sentences making the problem text are written in simple sentences, which is the concern of sentence complexity resolution sub-component in MASTER. Hence, the first experiment focuses on the accuracy of the proposed SCR on simplifying complex sentences into simpler sentences that could ease the process of schema instantiation for MASTER. Here for this experiment, the system was subject to the total dataset containing 457 Amharic AWPs among which 167 are AWPs with complex sentences.

The experiment focused to measure MASTER`s performance on identifying AWP with complex sentences and the accuracy of MASTER on sentence complexity resolution. Accordingly, MASTER identified 96% of the total 167 AWPs with complex sentences correctly as shown by Figure 6-1 and as Figure 6-2 shows MASTER resolved the sentence complexity of the total dataset provided with an accuracy of 97%.

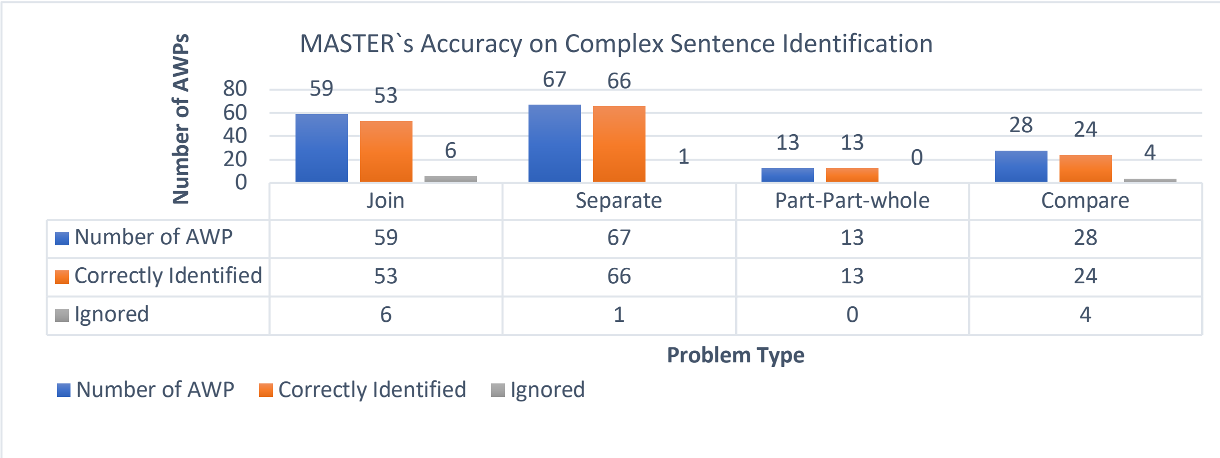

Figure 6-1 MASTER`s Accuracy on Complex Sentence Identification

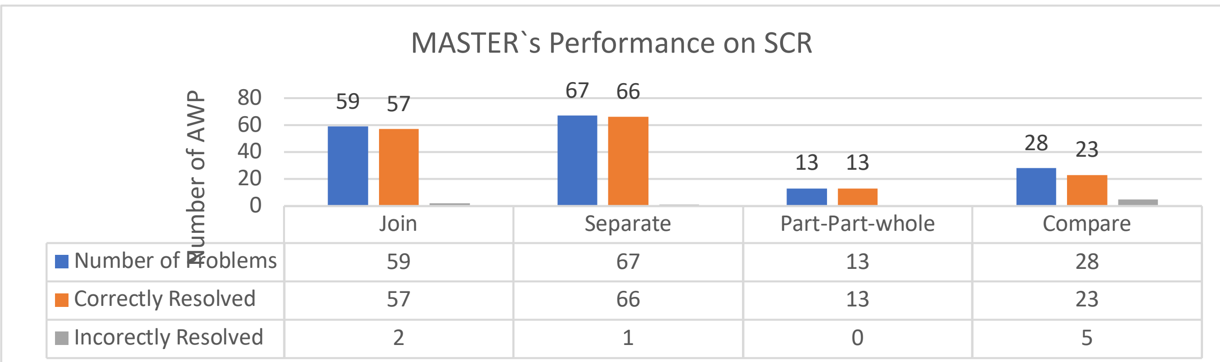

Figure 6-2 MASTER`s Performance on SCR

6.3 Accuracy of MASTER on Schema Instantiation

Schema instantiation is triggered by the appearance of keywords in a problem text as well as by a systematic identification of the type of sentences the problem text holds. Hence, a successful natural language answer for a given problem text is driven by a successful schema instantiation. Here, an experiment was conducted to check how accurate is MASTER to relate problems to their respective schema. For this experiment MASTER was subjected to the total dataset of 457 Amharic arithmetic word problems containing all four types of problems. To measure the performance of MASTER on schema instantiation, I computed the number of instantiations reported by the MASTER and the number of wrong instantiations reported by the MASTER per problem type. These numbers were then used to calculate the precision of MASTER on schema instantiation as follows.

$$Precision = \frac{Number\ of\ Correctly\ instantiated\ problems}{Total\ number\ of\ problems} * 100\%$$

Accordingly, Figure 6-3 shows the result of MASTER performance on schema instantiation for the given problems set represented by precision. MASTER has proven its robustness with an average precision of 89.48 %.

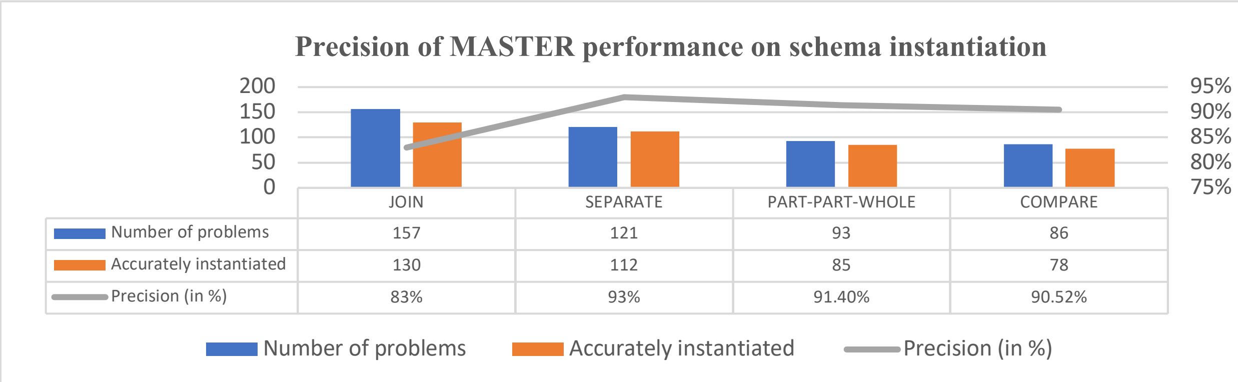

Figure 6-3: MASTER performance on schema instantiation

6.4 MASTER`s Overall Performance on AWP Solving

The performance of the proposed approach was measured on the datasets ADS1 and ADS2 briefed above. Table 6-1 shows the result of the experiment. MASTER has shown an average performance of 88.81%. With increased problem difficulty level presented by data set ADS2, MASTER retains its performance with only a 3.74% variation. A systematically designed Amharic schema with a list of keywords to represent the linguistic concepts used in the problem text and a strategy to simplify complex sentences in the problem text to simple sentences has helped MASTER to keep its robustness on difficult questions.

Table 6-1: MASTER`s performance on ADS1 and ADS2

Problem Type	Performance on ADS1 (in %)	Performance on ADS2 (in %)	Average Performance (in %)
JOIN	83	83	83
SEPARATE	94.67	89.75	92.21
PART-PART-WHOLE	95.5	90	92.75
COMPARE	89.54	85	87.27
Average Performance per dataset	**90.68**	**86.94**	**88.81%**

7 CONCLUSION

This paper presents a schema-based approach to solving Amharic arithmetic word problems. The proposed approach's solving process involves four main phases: preprocessing, simplification, Knowledge representation, and natural language answer generation. The preprocessing task mainly focuses on segmentation to identify individual sentences that make up the problem text, tokenization to identify individual tokens in each sentence of the problem text, and POS tagging to identify the part of speech that each token in a given sentence belongs to. This is followed by the problem simplification phase, which involves normalizing numeral values and converting complex sentences into more straightforward sentences. Then, the preprocessed problem text containing a set of simplified sentences with their linguistic information is passed on for the knowledge representation phase. Amharic schemas, together with keywords, are used for knowledge representation. Concept extraction from each sentence to make up a concept vector is the first task in knowledge representation which is followed by schema instantiation to identify a specific schema to apply to a given problem. The Natural Language Generation in the final phase employs a set of heuristics to compile the final answer. A software prototype was developed using python, and experimental results showed 89.48% accuracy on schema instantiation and 88.81% on overall performance. Finally, the approach can be extended to a new class of algebraic problems by introducing a new schema with keywords. This research work opens several avenues for researchers who want to research this area.

References

[1] Andinet A. (2020) Automatic Generation of Amharic Math Word Problem and Equation. Journal of Computer and Communications, 8, 59-77. https://doi.org/10.4236/jcc.2020.88006.

[2] Nandhini, K. and Balasundaram, S.R. (2011) Techniques for Generating Math Word Questions for Learning Disabilities. IJCA Proceedings on International Conference and Workshop on Emerging Trends in Technology (ICWET), No.4, 45-50.

[3] Amharic Language. http://www.lonweb.org/link-amharic.htm, Last Accessed on June 17, 2020.

[4] Dehadari, J., Tounsi L. and Genabith, J. (2011) Morphological Features for Parsing Morphologically-Rich Languages: A Case of Arabic. Proceedings of the 2nd Workshop on Statistical Parsing of Morphologically Rich Language (SPMRL 2011), Dublin, Ireland, October 2011, 12-21,

[5] Gasser, M. (2011) HornMorpho: A System for Morphological Processing of Amharic, Oromo, and Tigrinya. Conference on Human Language Technology for Development, Alexandria, 2-5 May 2011, 94-99.

[6] Alemu, B. (2013) A Named Entity Recognition for Amharic. MA Thesis, Addis Ababa University, Addis Ababa.

[7] Wanintorn Supap, Kanlaya Naruedomkul, and Nick Cercone, (2013) Mathmaster: an alternative math word problems translation. Computational Approaches to Assistive Technologies for People with Disabilities, 253:109.

[8] Y. Bakman. (2007) Robust understanding of Word Problems with Extraneous Information, HTTP:// lanl.arxiv.org/abs/math.GM/0701393.

[9] N. Kushman, L. Zettlemoyer, R. Barzilay, and Y. Artzi. (2014) Learning to automatically solve algebra word problems. In ACL, pages 271–281.

[10] Chiang, T. and Chen, Y. (2019) Semantically-Aligned Equation Generation for Solving and Reasoning Math Word Problems. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, June 2019, 2656-2668. https://doi.org/10.18653/v1/N19-1272.

[11] Yimam, B. (1995). Yamarigna Sewasiw (Amharic Grammar). Addis Ababa: EMPDA.

[12] Demeke, G.A. and Getachew, M. (2006) Manual Annotation of Amharic News Items with Part-of-Speech Tags and Its Challenges. Ethiopian Languages Research Center Working Papers, 1-16.

[13] Alemu, A.A. and Lars, A. (2007) An Amharic Stemmer: Reducing Words to their Citation Forms. Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources, Prague, June 2007, 104-110. https://doi.org/10.3115/1654576.1654594.

[14] Christian Liguda and Thies Pfeiffer. 2012. Modeling math word problems with augmented semantic networks. In Gosse Bouma, Ashwin Ittoo, Elisabeth Mtais, and Hans Wortmann, editors, Natural Language Processing and Information Systems, volume 7337 of Lecture Notes in Computer Science, pages 247–252. Springer Berlin Heidelberg.

[15] Oliphant, T.E. (2006) Guide to NumPy, Trelgol Publishing.