RESEARCH ARTICLE OPEN ACCESS

Information Extraction System for Amharic Text

Sinatyehu Hirpassa
Research Scholar in Computer Science
Punjabi University
Punjab-India

ABSTRACT
The number of Amharic documents on the Web is increasing as many newspaper publishers started providing their services electronically. The unavailability of tools for extracting and exploiting the valuable information from Amharic text, which is effective enough to satisfy the users has been a major problem and manually extracting information from a large amount of unstructured text is a very tiresome and time consuming job, this was the main reason which motivates the researcher to engage in this work. The overall objective of the research was to develop an information extraction system for the Amharic vacancy announcement text. The system was developed by using Python and rule-based technique was applied to address the problem of automatically deciding the correct candidate texts based on its surrounding context words. 116 Amharic vacancy announcement texts which contain 10,766 words were collected from the "Ethiopian reporter" newspaper published in Amharic twice in a week. For this study, eight candidate texts are selected from Amharic vacancy announcement text, these are organization, position, qualification, experience, salary, number of people required, work agreement and deadline.. The experiments have been carried out on each component of a system separately to evaluate its performance on each component, this helps us to identify drawbacks and give some clue for future works. The experimental result shows, an overall F - measure of 71.7% achieved. In order to make the system to be applicable in this domain, which is Amharic vacancy announcement, further study is required like incorporating additional rules, improving the speed of the system by modifying the algorithm, a well-designed user interface and integrating other NLP facilities.

Keywords:- Information Extraction, Natural Language Processing, Feature Extraction, Extraction Patterns, Name Entity Recognition

I. INTRODUCTION

Rapid expansions in Information and Communication Technology are making available vast amount of data and information. Much of these data are in electronic forms (like more than a billion documents in the Web. Usually these data are unstructured or semi-structured and can generally be considered as a text database. Likewise, the recent decades witnessed a rapid production of Amharic textual information available in digital form in a numerous of repositories on the Internet and intranets. As a result of this growth, a huge amount of valuable information, which can be used in education, business, health and many other areas are hidden under unstructured representation of the textual data and is thus hard to search in. This resulted in a growing inquiry for an effective and efficient method for analyzing free-text data and find out valuable and relevant knowledge from this text in the form of structured information, and led to the emergence of Information Extraction technologies.

IE is one of the NLP applications that aim to automatically extract structured factual from unstructured text. Riloff [1] discusses, the task of automatic extraction of information from text involves identifying a predefined set of concepts and deciding whether a text is relevant for a certain domain, and if so extracting a set of facts from that text. During the last ten years, IE has become an increasingly researched field. As [1] stated, "unfortunately, during this time most of the known IE systems have been developed for texts written in the English language. In comparison to the achievement registered for English IE systems for most of other languages are still lacking essential components".

In Ethiopia most of Amharic news such as, science and technology, sport and business are available online. Most of this news is presented in unstructured and semi-structured text forms, then reader's looks relevant information from the text manually, according to Cowie and Wilks [2], manually extracting information from such an often unstructured or semi-structured text is a very tedious and lingering task.

Thus, getting the accurate information for decision-making from existing abundant unstructured text is a big challenge. In addition, the unavailability of tools for extracting the valuable information which is efficient enough to satisfy the users of Amharic language has also been a major problem.

It is hoped that the availability of an IE system can ease the information searching process. IE, in contrast to other research domains is language and domain dependent [8]. The IE system developed for English text with specific domain is not work or applicable for Amharic language even if its domain is similar.

There are different language specific, issues which may not be handled by the system developed for English. Thus, this work was aimed to develop suitable model and algorithms for Amharic news text information extraction and finally, evaluate the performance and usability of the system.

1.2. METHODOLOGY

1.2.1. Data sources and data set preparation for the experiment

The researcher collected different Amharic vacancy announcement texts that were required for training and testing the system from the "Ethiopian Reporter" newspaper published in Amharic twice in a week. For the purpose of this study, 116 Amharic vacancy announcement texts that contain in 10,766 words were selected purposefully with different range of vacancy announcements. Their dissimilarity is based on the organization of who is posting the vacancies and the type of vacancies. The newspaper was chosen as a data source since it has a large collection of Amharic vacancy announcement texts in its database.

1.2.2. Design and implementation of Amharic vacancy announcement texts

The designing phase contains the document pre-processing, learning and extraction, and post processing as the three main components. In order to develop a prototype system, different appropriate tools have been selected and employed, and different data pre-process IE modules, such as Tokenizer, and Normalizer, which are mostly language specific algorithms are developed using python programming language. This programming language was used for developing candidate text identifier, and tagger and candidate text extractor. The POS which is developed by Gebrekidan [6] is used as one of the features in IE component. Also, Microsoft SQL server 2008 was used to store up extracted information or candidate words.

II. BUILDING IE SYSTEM

In principle, designing IE has two approaches: (1) the learning approach or Automatic Training Approach., and (2) the Knowledge Engineering approach.

The Knowledge Engineering (KE) approach needs a developer, who is an expert on both the requirements of the application area and the function of the IE system. The developer is concerned with the definition of rules used to identify and extract the appropriate information. Therefore, a corpus of domain-relevant texts will available for this task [5].

Building a high performance system is usually an iterative process whereby a set of rules is written [1], the system executes over a training corpus of texts, and the output is examined to see where the rules under and over generate. The knowledge engineer then makes appropriate modifications to the rules, and iterates the process [3]. Thus, the performance of the IE system depends on the skill of the knowledge engineer.

The Automatic Training Approach is quite different from the knowledge engineering approach, because in this approach, it is not necessary to have someone on hand with detailed knowledge of how the IE system works, or how to write rules for it. It requires only someone who knows enough about the domain and the task to take a corpus of texts, and annotate the texts appropriately for the information being extracted. Typically, the annotations focus on one particular aspect of the system's processing. For example, a name recognizer would be trained by annotating a corpus of texts with the domain-relevant names. A co reference module would be trained with a corpus representing the co reference equivalence classes for each text. Once a suitable training corpus has been annotated, a training algorithm is executed, and resulting in information that a system can employ in analyzing candidate texts. Another approach to obtaining training data is to interact with the user during the processing of a text. The user is permitted to designate whether the system's hypotheses about the text [4, 9]. The above mentioned approaches for IE can be applied on the free text or semi structured or structured text which is used as an input for IE system [7].

III. ARCHITECTURE of IE SYSTEM

Different scholars use different steps for designing extracting information system for different language and different domain. The research work in [3] mainly categorizes IE into six different tasks.

1. Part-of-Speech (POS) Tagging
2. Named Entity Recognition (NER)
3. Syntax Analysis
4. Co-references and Discourse Analysis
5. Extraction Patterns
6. Bootstrapping

1. Part-of-speech tagging (POS) it is the act of conveying each word in sentences of tag that describes how that word is used in the sentences. That means POS tagging assigns, whether a given word is used as a noun, adjective, verb, etc.

As Pal and Molina [10] acknowledges, one of the most well-known disambiguation problems is POS tagging, because many words are ambiguous: they perhaps assigned more than one POS tag (E.g., the English word round may be a noun, an adjective, a preposition or an adverb, or a verb).

POS tagger finds the possible tags or lexical category for each word provided that the word is in a lexicon and guess possible tags for unknown words. It also chooses possible tag for each word that is ambiguous in its part-of-speech. If certain words is assigned more than one tag, this means that the word can have different meanings or function in different context.

2. Named entity recognition (NER) named entities are one of the most often extracted types of tokens during extracting information from documents. Named entity recognition is classification of every word in a document as being a person-name, organization, location, date, time, monetary value, percentage, or "none of the above". Some approaches use a simple lookup in predefined lists of geographic locations, company names, person names and name of animals and other things from the gazetteers, while some others utilize trainable Hidden Markov Models to identify named entities and their type.

3. Syntax analysis, in contrast to POS tagging, syntax analysis, also called syntax parsing, looks beyond the scope of single words. During syntax analysis we attempt to identify syntactical parts of a sentence (verb group, noun group and prepositional phrases) and their functions (subject, direct and indirect object, modifiers and determiners). Simple sentences, consisting, for instance, of a main clause only, can be parsed using a finite state grammar. Simple finite state grammars are often not sufficient to parse more complex sentences, consisting of one or more subordinate clauses in addition to the main clause, or containing syntax structures, such as prepositional phrases, adverbial phrases, conjunction, personal and relative pronouns and genitives in noun phrases [11].

4. Co-references and Discourse Analysis it is a process of finding multiple references to the same object in a text. It refers to the task of identifying noun phrases that refer to the same extra linguistic entity in a text. This is especially important since the same thing about a single entity is expressed in different sentences using pronouns [3].

5. Extraction Patterns the resulting output of IE consists of single data items filled into the slots of data tuples templates. The data tuples populate the result database, one tuple for each relevant document of the input text corpus. The data items are pieces of information which have to be located in the text. Extraction patterns are used for this task.

6. Bootstrapping, as Johannes [3] notes that, newer systems use various bootstrapping algorithms to improve the results of the pattern matching, or do unsupervised named entity recognition. Some systems require a test corpus to evaluate the results of the pattern matching and bootstrapping process. During the bootstrapping the following steps are iterated:

- Apply all seed patterns in the whole text corpus and split the text corpus into two categories, so that one category contains all relevant texts in which one or more seed patterns scored and the other category contains all the other texts.
- Score all the patterns gained from the text corpus based on their density of distribution in relevant documents in comparison to their density of distribution in all texts.
- Use the highest scoring patterns to generate concept classes by merging those pairs which appear in the correlated text.

IV. PROPOSED MODEL

Johannes [3] acknowledged that, every IE system has three basic components which are the linguistic preprocessing, learning and extraction and post processing regardless of the approach, language and domain on which the IE system is

developed for. The model which is designed in this work is has three major components and each component also contain different subcomponents which are language specific and general subcomponents that are required in IE.

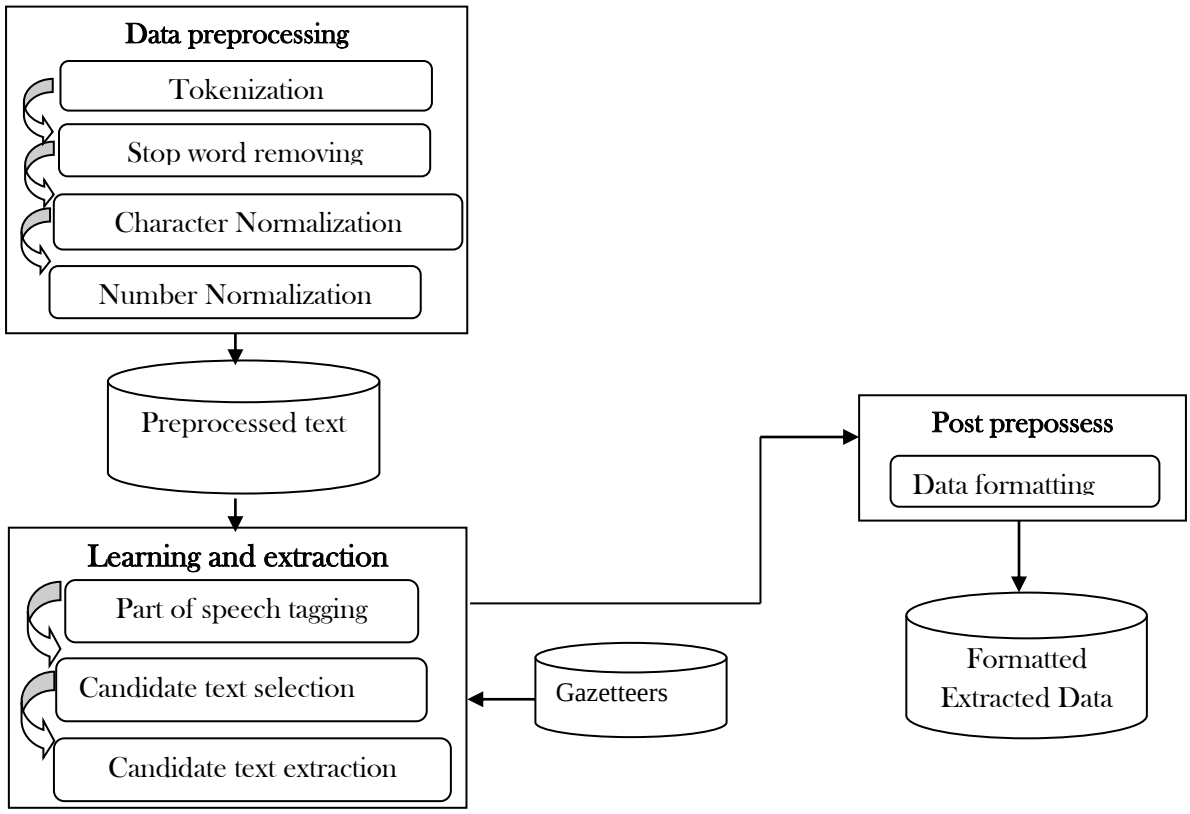

Figure1. Model of Amharic texts information extraction

4.1. Data preprocessing

In the preprocessing stage file formats, character sets, and variant forms can be converted, so that all text, regardless of its source, is in the same format. In later stages all further processing can then be consistently applied to all the data. In this stage a language specific issue such as tokenization, normalization, and stop word removal are addressed in this study.

1. Tokenization

As it is defined by Siefkes and Siniakov [6], it is the process of splitting the text into sentences and tokens. It Start with a sequence of characters to identify the elementary parts of natural language such as words, punctuation marks and separators. It is generally known that; tokenization is an important step in NLP, particularly for information extraction system. As, we know there is no single right way to do tokenization. The right algorithm depends on the application.

In this work, words are taken as tokens. All punctuation marks (except "/"), control characters, and special characters are removed from a text before the data is transferred for further process.

/ (ህዝባር) the Amharic slash has its own role during text normalization, due to this; it would not be removed during this process. The tokenizer, which is adopted for the purpose of this work is, used the following punctuation more prominently, such as **::** (አራት ነጥብ) the Amharic full stop and ፣(ነጠላሰረዝ) the Amharic comma for tokenization process, because they are the most commonly used punctuation marks in the Amharic texts.

።(አራትነጥብ) the Amharic full stop is used for identifying the sentence demarcation and (ነጠላሰረዝ) the Amharic comma is used to separate different text segments which mostly are

related. When these punctuations are found in a text a single space would be added between the word and punctuations by the system. The tokenizer then tokenizes all the text segments which have space between each other as independent token.

2. Normalization

A. Character normalization:

It is generally known that, in Amhric writing system different characters with the same sound are available.
These different symbols must be considered as equivalent because they do not cause a change in meaning in IE system. Though from the linguistics view this character variation might be have meaning, they need to be normalized when developing an IE system for Amharic language, because spelling variations of a word would unnecessarily increase the number of words representing a document, which could reduce the efficiency and accuracy of the system.
The letters such as ሀ,ኀ, ሃ, ኻ, ሐ,ኅ and ሐ, ዐ and አ, ሠ and ሰ, ፀ and ጸ, are the characters with the same meaning and pronunciation but different symbol. This character variation also exists in most of Amharic texts. These characters should be normalized to a single character like ሠ to ሰand ኀ, and ሐ to ሀand ፀto ጸas well as their orders (ሠ, ሡ, ሢ, etc. to ሰ, ሱ, ሲ, etc.) consequently.

B. Number normalization

Also in Amharic text there are different entities which are represented by number, but there is no standard for writing numbers in Amharic; someone may write by using only Arabic numerals, others may write by mixing both Ethiopic numerals with Arabic numerals. For example, in most of the Amharic texts the salary 8000birr is written as "8 ሺህ000 or 8,000.00 or ስምንትሺህbirr. This representation in causes problem during information extraction. The number Normalizer changes all above listed of number representation into their equivalent number representation.

3. Construction of Stop-word List

Like any other language, Amharic writing system also contain different stop-words include prepositions, conjunctions, and articles. Even though these words are important in writing a document, they haven't advantage in designing the NLP system. Using these words in a dataset as it is will have an impact on the performance the system, which mean, they would degrade the speed and takes much memory space. Due to this, removing stop-words from a dataset is necessary to reduce file size and processing time.

In this work, in order to generate most frequent stopwords from Amharic text a new algorithm is developed. While establishing a general stop-word list, first we generate all the word forms appearing in a dataset are sorted according to their frequency of occurrence and the top most frequently occurring words are extracted, and then this list was examined manually to identify important word. Finally, some non-information-bearing words were included manually though they did not appear in the first top most frequent words.

4.2. Learning and extraction component

This component is the fundamental part of this model, which mainly deals with candidate texts. It uses the output of the document preprocessing component as an input. The extraction and learning component also comprises different sub components that are used to make the data ready for extraction.

a. Part of speech tagger (POS):

Part of speech tagging, or simply tagging is the task of labeling (or tagging) each word in a sentence with its appropriate part-of –speech. It is a technique for deciding whether each word is noun, verb, adjective, adverb, etc. [1]. POS tagger has been applied to assign a single best POS to every word in the corpus.
There are different part of speech tags of set in the Amharic writing system, but for the purpose of this study, we used 12 such as noun, noun phrase, verb, verb phrase, adjective, adverb, prepositions, punctuation, numeric, conjunction, verbal noun and noun consonant.
Since there is no POS tagged corpus available for specific domain Amharic text the dataset was selected; preprocessed and 20 % from total words are manually tagged to train the POS tagger.

b. Candidate text selection:

Once the POS tagging process completed the next activity is identifying the possible candidate texts that would be extracted from the Amharic texts. Since our domain of data set is vacancy announcement the name of the organization, job position, Qualification and location are considered as the candidate text. In order to identify and select the name of organization and job position in the Amharic vacancy announcement text, the gazetteer is incorporated with the system. The Gazetteer, which comprises lists of different organization and job position names under consideration. The other candidates such as Qualification Salary, Agreement, Year of Experience, and Number of people needed, Deadline, and Phone are selected by analyzing their feature words.

The new algorithm is developed, which used to extract the features of each candidate. The following features are going to find: The current candidate word, previous/following of candidate word, the word before/after the previous /following word, POS of the above listed words, and the token category of the candidate token. After the candidate texts are identified from the dataset, they are tagged accordingly to their attributes. Here are the tags; those are used to tag the candidates:

<ORG> "organization name"
<POS> "job position name"
<QUL> "expected qualification in that position"
<EXPER> " year of experience"
<SAL> "salary"
<AGRE"EMENT> "job Agreement"
<NEED> "number of people needed"
<DEAD> "deadline"

Candidate text identifier, and tagger Algorithm

Read raw of corpus
Read gazetteer which contain list of organization name
Read gazetteer which contain list of position name
vacancy = each vacancy in raw of corpus
sting = tokens in vacancy
org = each organization name in gazetteer
pos = each position name in gazetteer
For vacancy
For string
If org == string
Tag the organization name by <ORG> at the beginning and </ORG> at the end of the organization name
End if
End for
For string
If pos == string
Tag the position name by <POS> at the beginning and </POS> at the end of the position name
End if
End for
If string == "ብዛት" and string + 1 == "<ADJ>"
Tag at end of the next word by <NEED>
If string == "የቅጥር" and string +1 == "<NP>" and string + 2 == "ሁኔታ"
If string == "የቅጥር" and string +1 == "<NP>" and "አይነት"
Tag at end of the next word the by <AGREE>
If string =="ደመወዝ" and string+1 == "<VN>" or string == "ደሞዝ" and string+1 == "<VN>
Tag at end of the next word the by <SAL>
If string == "አመት" and string +1 == "<NUMP>" and string +2 == "ከዚያ" and "<PRONP>" and "በላይ"
Tag the word befor "አመት" by <EXPER>
Tag the word after "አመት" by </EXPER>
Elseif string =="አመት" and string +1 == "<NUMP>" and string +2 == "የስራ"and string +3 == "<NP>" and "ልምድ"
Tag the word befor "አመት" by <EXPER>
Tag the word after "አመት" by </EXPER>
If string == "ተከታታይ" and string +1 "<ADJ>" and "የስራ"and string +2 == "<NP>" and "ቀናት"
Tag the word befor " ተከታታይ" by <DEAD>
Tag the word after " ቀናት" by </DEAD>
Elseif string == "ተከታታይ" and string +1 "<ADJ>" and string+2== "ቀናት"
Tag the word befor " ተከታታይ" by <DEAD>
Tag the word after " ቀናት" by </DEAD>
Elseif string "የስራ"and string +1 == "<NP>" and "ቀናት"
Tag the word befor " ተከታታይ" by <DEAD>
Tag the word after " ቀናት" by </DEAD>
End if
If string =="ስልክ" and string+1 == "<N>" and "ቁጥር"
Tag at end of the next word the by <PHONE>
Elseif string == string =="ስልክ" and string+1 == "<N>" or "መረጃ"
Tag at end of the next word the by <PHONE>
Elseif string == "ለበለጠ" and string +1 =="መረጃ" and string+2 == "<N>" and string +4 ==<NUMCR>
Tag at end of the next word the by <PHONE>
End if string == "የትምህርት"and string + 1==<NP> or "ተፈላጊ" and string + 1 == <ADJ> and
If string == "ደረጃ"and string +1== <ADJ> or "ችሎታ" and string +1== <ADJ>
Tag the word after "ደረጃ" by <QUL> or
Tag the word after "ችሎታ" by <QUL>
Elseif string == "የተመረቀች" and "<PUNC>"
Tag at end word "የተመረቀች" by </QUL>
Elseif string == "ዲግሪ" and "<PUNC>"
Elseif string == "ዲፕሎማ" and "<PUNC>"
Tag at end word "ዲፕሎማ" by </QUL>
Elseif string == "ሰርተፊኬት" and "<PUNC>"
Tag at end word "ሰርተፊኬት" by </QUL>
End if
End for

c. Candidate text extraction:

Once the intended candidates are identified and tagged, in this phase, extraction of those candidate texts will be carried out in respect to their category. The other data those not selected by the system as a candidate text from texts would be discarded.

A rule-based algorithm is developed, which aided to extract such a tagged candidate text from a dataset.

Here is the algorithm:

```
Read raw of corpus
Vacancy = each Vacancy in raw of corpus
String = each tokens in vacancy
For string
    If string is tagged by <ORG>
        Hold the position

        While( string != </ORG>)
            Print the string
            Increament string
        End while
End if
If string is tagged by <POS>
        Hold the position
        While( string != </POS>)
            Print the string
            Increament string
        End while
End if
```

4. 3. Post processing

This is the last component of the model. After the relevant information has been founded by applying the extractor algorithm on the given dataset, the extracted candidate text fragments are assigned to the corresponding attributes of the target structure and store them in the database according to the predefined format of the database slots. In this work eight attributes those extracted are stored in the Database. Thus, the main function of the post processing component is to arrange the format and store the extracted data in a database, after that it will be flexible for data mining or any other application which want to use the data. The extracted candidate texts are also normalized according to the expected format, since, some identified facts may appear in the text more than once and there might be a violation the properties of the database.

V. RESULT AND EVALUATION

Information Extraction system is also expected to extract the right information from a text. What constitutes the correct output and how we can measure it is, however, not an easy task and so is an active area of research in IE. Therefore, raising one or more questions of accuracy, user-friendliness, efficiency, modularity, portability and robustness is important depending on the purpose.

5.1. Evaluation metrics

In this work, I will do mainly an intrinsic, black-box and automatic evaluation. We evaluate the different information extraction algorithms as isolated systems (intrinsic). Within the isolated system, we are going to do black-box evaluation as we will only compare the outputs of the system for given inputs with the gold standard. The most commonly used evaluation metrics in information extraction are precision, recall and f-measure.

Mathematically, $P = \frac{TP}{TP+FP}$ $R = \frac{TP}{TP+FN}$ $F = \frac{2PR}{P+R}$

5.2. The datasets

The dataset used for this work was Amharic vacancy announcement texts acquired from the "Ethiopian reporter" newspaper published in Amharic twice in week.116 Amharic vacancy announcement texts that contain in general 10,766 words were selected purposely with different range of vacancy announcements. Their dissimilarity is based on the organization of who is posting the vacancies and the type of vacancies.

Table 1. Statistics for dataset used

	Training data	Test data
Number of vacancies	82	34
Number of word (tokens)	8,046	2,720
Number of organization data	82	34
Number of job position data	82	34
Number of qualification	82	34
Number of salary data	78	32
Number of people needed data	62	26
Number of experience data	76	29
Number of deadline data	82	34
Number of phone data	69	28

5.3. Experimental results and evaluation of each component of our system

5.3.1. Result and evaluation of normalization

The performance of our system has been evaluated before and after document normalization. The experimental result showed that document normalization has a significant effect

on the performance of the system. Consider figures 2 and 3 to see the impact of document normalization:

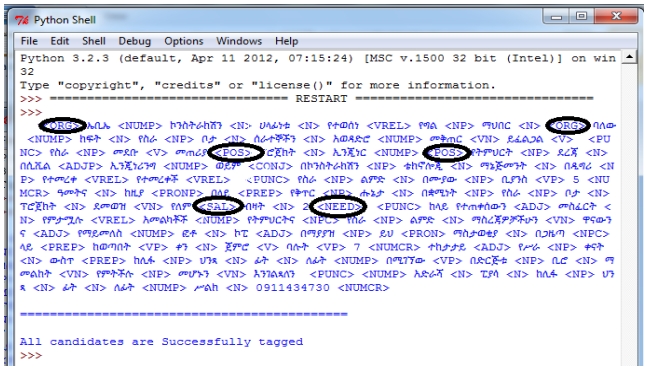

Figure 2. Before normalization.

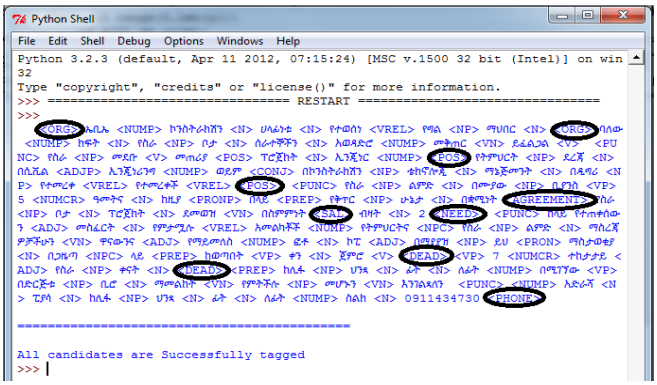

Figure 3. After normalization.

As the first result illustrated that, in a single Amharic vacancy announcement text, the system considered only four data as a candidate text, but it escalated into seven for the same texts as it is depicted in fig 2, due to applied of normalization. Thus, before providing any type Amharic texts to the Amharic information extraction system, it should be normalized.

5.3.2. Experimental results and evaluation of stop-word removal

As discussed in section 4, using stop-words in dataset as it is an impact on the performance of the system.

Table 2. Effects of stop-word removing on the running speed of the system

Datasets	Before stop-words are removed	After stop-words are removed
AVAT	Time	Time
	2:42 min	1:36 min

Table 5.2 shows that running speed of the system is increased by 1.06 minute than before stop-words were removed. Still, the running time indicated that it could improve, if all unnecessary words are removed.

Nevertheless, in this work, it is impossible to say all stop-words were included, when the stop-word list is constructed.

5.3.3. Experimental results and evaluation of part of speech tagger

Nowadays, there are different types of NLP tools are available, though, not all tools are fully relevant for Amharic language. Among this POS tagger is a one tool which is commonly used in designing most of natural language proceeding system [6].

For the purpose of this study two statistical POS taggers were tested, the first one is a Brill POS tagger for Amharic language, which was developed by Gebrekidan [6]. The Bigram POS tagger is another tagger that we have tasted in this study, which is developed by Abebe [12].

Table 3. Experimental results of Bigram POS

	Correctly tagged	Incorrectly tagged
Number of words	1565	640

Table 4 Experimental result of Brill POS tagger

	Correctly tagged	Incorrectly tagged
Number of words	1962	243

From the above table what we can understand is Brill POS tagger has 89.5% of accuracy and Bigram POS tagger has an accuracy of 71.4 %. Hence, the researcher selected and used the Brill POS tagger for tagging the dataset.

5.3.5. Experimental results and evaluation of organization and position extraction

Two algorithms have been tested to handle and extract organization and position candidate texts from Amharic vacancy announcement texts. The first algorithm is based on feature words or context information, which means extracting candidates based on the neighborhood features words those can express the name organization and position. Gazetteer based identification and extraction was another algorithm that the researcher had tested. I evaluate the performance of the system for identification and extraction by using two known evaluation mechanisms in NLP, they are Recall and precision. In this case, Recall is the proportion of candidate texts which are extracted correctly over the total number of extracted candidates for each slot in the test dataset.

Likely, precision is the proportion of candidate texts which are identified and extracted correctly over the number of identified and extracted for each slot in the test dataset.

Table 5 Experimental result of context information based algorithm for organization name and position extraction.

	Recall	Precision	F-measure
Organization	47.8	64.4	54.8
Position	36.9	56.2	43.19

Table 6. Experimental results of gazetteer based algorithm for organization name and Position extraction

	Recall	Precision	F-measure
Organization	100	100	100
Position	100	100	100

The experimental result showed that integrating gazetteer with organization and position extractor algorithm could have an ability to improve the performance of the system. The main reason why feature based organization and position identification and extraction algorithm was not good as gazetteer based algorithm is that: in different Amharic vacancy announcement texts, both organization and position presented in several ways, which means their presentation likeness from one Amharic vacancy announcement texts to another is very rare. Therefore, it is not possible to handle and discover all those various ways of representation based on feature words or context information. As a result of this, the second algorithm was not effective in identifying and extracting organization and position name as gazetteer based algorithm.

5.3.6. Experimental results and evaluation of other candidate text extraction

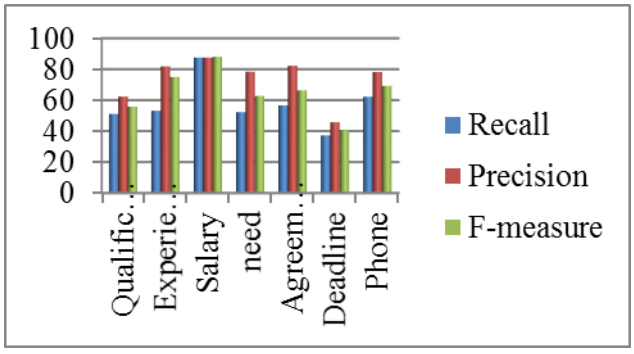

Fig 4. Experiment result of the rest candidate text extraction

The candidates such as “Salary”, “Number of needed people”, “Agreement” and “Phone” provide the best performance. It might be due to the facts those used for representing these candidate texts in Amharic vacancy announcement texts usually uses the same pattern: For example, job agreement is presented in most of the time in the following format: <የቅጥርሁኔታ>or <የቅጥርአይነት> expected word are “በቋሚነት”, “ቋሚ”, “በኮንትራት”, “ኮንትራት”. The worst performer was “Qulifcation” and “Deadline” slot. The main reason was over generalization in specific selcetion rule:"የትምህርትደረጃ" * "የተመረቀ ወይም የተመረቀች and <PUNC>" or “ተፈላጊ ችሎታ” *“ዲግሪ and <PUNC>". This rule is meant to match “የትምህርት ደረጃ ከዩኒቨርሲቲ በግብርና ሰፕላይስ ማኔጅመንት በኢኮኖሚክስ በቢኤ ዲግሪ የተመረቀ ወይም የተመረቀች <PUNC” and “ተፈላጊ ችሎታ በአካውንቲንግ የመጀመሪያ ዲግሪ <PUNC>” respectively, but it also matches the wrong sentence like “በአካውንቲንግ የመጀመሪያ ዲግሪ ያለው” or “ዲግሪ ያለው”. We need to inspect more vacancy announcement documents in order to refine the selection rule and to improve our system performance.

Generally, the result of the experiment shows that, our system can still be improved. Although this algorithm shows good result with precision, that is 79.56%, a cumulative recall is lower at 66.6% and F-measure was 71.7%. Low recall is common in most of IE. Using job domain document, RAPIER Calif [13] had precision84% and recall 53%.

VI. CONCLUSION

In this work, I presented the first rule-based IE system for Amharic text. The following conclusions are drawn from the experiments with regard to the research objectives:

- The results obtained from experiments shows 79.56% precision and 66.6% recall on 34 Amharic vacancy announcement texts test dataset.
- The Amharic information extraction system does not give similar accuracy on different datasets. Its accuracy depends on the features found in each candidate. Some candidates may contain similar feature words, that incorporated within the rule and other may contain features words out of these, therefore their accuracy may differ depending on the test dataset.
- The experiments have been carried out on each component of a system separately, in order to evaluate them individually. Based on the result, candidate text selector algorithm has shown less accuracy as compared with other components, this due to lack of adequate rules or feature words for each candidate text.
- Extracting candidates is a challenging task in rule-based algorithm, because one candidate text may appear in various ways in different Amharic texts. In this research, various rules were tried to incorporate in candidate text selector Algorithm, which used to identify candidate text in Amharic vacancy announcement texts. So, I can confidently say that it is promising to develop an IE system using the knowledge engineering approach.

6.1. Recommendation

The following recommendations are forwarded for future work:

- The dataset used in this system is only from one newspaper. Using a sizable dataset from different newspaper could possibly help to get diversified rules and improved performance.
- Further research is expected on different IE tools, such as sentence Parser, POS, NER, and Coreference Resolution for Amharic language to develop an effective information extraction system.
- It would be interesting to implement a statistical algorithm to identify and extract candidate texts and test to see how it performs for this model

REFERENCE

[1]. Ellen Riloff, Inducing Information Extraction Systems for New Languages via Cross-Language Projection, School of Computing, University of Utah, Salt Lake City, June 2004.

[2]. Jim Cowie and Yorick Wilks, Information extraction, Lecture notes on Information extraction, 2007.

[3]. Philipp Johannes, Multilingual Information Extraction, Department of Computer Science, University of Helsinki 15th, February 2004.

[4]. Cunningham H., Automatic Information Extraction , Encyclopedia of Language & Linguistics journal, Second Edition, volume 5, pp.665-677, Oxford, Elsevier, 2006.

[5]. Juan Antonio P´erez-Ortiz and Mikel L. Forcada, Part-of-Speech Tagging with Recurrent Neural Networks, Departament de Llenguatges SistemesInform`aticsUniversitatd'Alacant E-03071 Alacant, Spain, 2001.

[6]. Binyam Gebrekidan, Natural Language Processing & Human Language Technology, Part of Speech Tagging for Amharic, MA Thesis, School Of Law, Social Sciences and Communications, United Kingdom, June 2010.

[7]. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, An Introduction to Information Retrieval, Cambridge University, England, April 2009.

[8]. Ralph Grishman, Silja Huttunen and Roman Yangarber, Information extraction for enhanced access to disease outbreak reports, Journal of Biomedical Informatics 35(4): 236-246, 2002.

[9]. Katharina Kaiser and Silvia Miksch, Information Extraction, A survey, Institute of Software Technology & Interactive Systems, Vienna University of Technology, May 2005.

[10]. Ferran Pal and Antonio Molina, Natural language engineering: improving Part of speech tagging using lexicalized HMMs, Cambridge university press, united kingdom, 2004.

[11]. Kameyama M., Information Extraction across Linguisti c Barriers, AAAI Spring Symposium Series on Cross-Language Text and Speech Retrieval, Stanford, 1997.

[12]. Ermias Abebe, Bigram part-of-speech tagger, Addis Ababa University, school of Information Science, Addis Ababa, 2006.

[13]. Mary Elaine Califf and Raymond J. Mooney, relational learning pattern-matching rules for Information Extraction, Department of Computer Sciences, University of Texas at Austin, July 1987.

BIOGRAPHY

Sintayehu Hirpassa

Sintayehu Hirpassa was born in Butajira, Ethiopia in 1987. I received the BSC degree in Information Science from Jimma University, Ethiopia in 2010, and M.Sc in Information Science from Addis Ababa University, Ethiopia in 2013 and now computer science Ph.D. scholar in Punjabi University, India. I have served for three years in one public university of Ethiopia as a lecturer in department of Information Systems.