I.J. Information Engineering and Electronic Business, 2021, 5, 51-59
Published Online October 2021 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2021.05.05

Boosting Afaan Oromo Named Entity Recognition with Multiple Methods

Abdo Ababor
Department of Information Technology, Wolkite University, Wolkite, Ethiopia
Email: abdo.ababor@wku.edu.et

Received: 29 July 2021; Revised: 16 August 2021; Accepted: 28 August 2021; Published: 08 October 2021

Abstract: Named Entity Recognizer (NER) is a widely used method of Information extraction (IE) in Natural language processing (NLP) and Information Retrieval (IR) aimed at predicting and categorizing words of a given text into predefined classes of Named Entities like a person, date/time, organization, location, etc. This paper adopts boosting NER for Afaan Oromo by using multiple methods. Combinations of approaches such as machine learning, the stored rules, and pattern matching make a system more efficient and accurate to recognize candidates name entities (NEs). It takes the strongest points from each method to boost the system performance by voting a candidate NE which is detected in more than 1 entity category or out of context because of word ambiguity, it penalized by Word senses disambiguation. Subsequent NEs tagged with identical tags merged as a single tag before the final output. The evaluation shows the system is outperformed. Finally, the future direction is forwarded a hybrid approach of rule-based with unsupervised zero-resource cross-lingual to enhance more.

Index Terms: Afaan Oromo, Named Entity Recognition, Word Sense Disambiguation, NLP, Information Extraction.

1. Introduction

NLP is a computational method that automates the translation between human languages and computers (without being fed a clue). One of the NLP subfields is Information extraction (IE) that concerned with identifying predefined types of information from text by applying the concept of NLP. IE aims to extract more useful (structured) information from semi-structured or unstructured documents automatically.

NER is a widely used method of IE in NLP and Information Retrieval (IR) aimed at predicting and categorizing words of a given text into predefined classes of Named Entities (NEs). NER is standalone information extraction and filtering tool. The most common NEs classes a person, location, facilities, products, organization, temporal expressions, or numeric expressions [1]. NER is an important task toward more intelligent IE and in many of the language engineering applications such as IE (e.g. relation and event extraction [2, 3]), indexing for IR, question-Answering (e.g. answer selection) [4], empowering recommender systems, Machine Translation, Automatic Summarization, and Semantic Annotation [5, 6, 7]. Moreover, NER is used in other NLP applications like opinion mining [8], ontology population [9], and text clustering [10] semantic search [11], and so on.

Now a day the advancement of technology attracts many researchers in language engineering applications for enhancing human-computer interaction. A lot of research conducted on rich-resource languages like English, Spanish, etc. however, on low-resource languages like Afaan Oromo many NLP research are almost on initial. Afaan Oromo is Africa's second most commonly spoken indigenous language, with 40% of Ethiopians speaking it [12, 13, 14]. It is commonly spoken and used in most parts of Ethiopia, as well as parts of neighboring countries such as Kenya, Somalia, and Djibouti [15, 16, 17] as well as in Tanzania. I motivated to minimize gap of text processing applications in Afaan Oromo and related Ethiopian language, and also to increase attention of researchers in the area.

This paper focuses on boosting Afaan Oromo NER with the core component that takes plain text as input, then performs entity identification and classification, word disambiguation, entity chunking, and output labeled NEs. Identification is the first task that finds candidate tokens that can be NEs from the tokenized plain text. Based on the combined knowledge supplied from the hybrid approach detection is performed as follows. The model, invoked by the recognizer, does several tasks turn by turn. The features that are extracted and stored during the training phase are supplied to the recognizer to identify NEs from the text. The model then selects candidate tokens based on the calculated probability. To check if there are wrongly detected tokens and there are found ambiguities by the model, the stored rules are used to disambiguate.

The classification portion is also done by a joint work from the ML model, the rule-based and pattern matched. The recognizer starts the classification process by tagging the detected candidate tokens. Each token will be tagged with their possible NE tags or not. The next task is chunking which is combining two or more continuous NEs having similar entity categories to be considered as a phrase and assigned a single tag. Finally, the extraction phase outputs NEs as per their tag category.

I have prepared the Afaan Oromo dataset of size 44,120 words out of which around 7809 are NEs that were collected from 3 news websites. From this dataset, I have used 90% for training, 10% for testing. Finally, the result shows it is outperformed in Precision, Recall, and F1-measure that are 86.37, 85.66, and 86.01 respectively.

2. Related Work

The most recent NER approaches use deep neural networks to avoid the expensive steps of designing informative features and constructing knowledge sources to their success is the availability of large amounts of labeled training data. However, building large labeled datasets for low-resource languages is expensive and time-consuming [18, 19, 20, 21].

As many languages lack suitable corpora annotated with named entities, there have been efforts to design models for cross-lingual transfer learning. This offers an attractive solution that allows us to power annotated data from a source language (e.g., English) to recognize named entities in a target language (e.g., German) [18].

One possible way is building a cross-lingual NER system that translates knowledge to the target language. However, it requires a broad knowledge of the target language. Another way is performing cross-language projection that uses a sentence-aligned bilingual parallel corpus to project its token level predictions to the target language, and finally, train a NER tagger on them. This type of parallel corpora is often not available for low-resource languages, and building such corpora could be even more expensive than building the NER dataset [18, 22, 23].

Afaan Oromo is one of the low-resource languages, thus making the task of NER more challenging. One of the major challenges in identifying NE is the ambiguity of words (multiple meanings contextually). Another major challenge is classifying similar words from texts. Words will be abbreviated to make writing and comprehension easier [6]. The same words can be written in long forms. Spelling variations are common like single vowel and double vowel in Afaan Oromo that play an important role.

A few researchers worked on entity extraction of Afaan Oromo language with different approaches Mandefro [24] performed the first research, which is a hybrid model of Machine Learning (ML) that uses the Conditional Random Field (CRF) algorithm and a rule-based approach. This system can classify NEs based on their location, organization, person, and other parameters. On the other side, a rule-based NER system developed concentrating on the general structure of Afaan Oromo NEs, pattern matching with the task of disambiguation [25]. Furthermore, Abdi [1] has used a hybrid approach that incorporates Mandefro's [24] approach as well as rule-based methods using the GATE tool for NEs. According to Abdi [25], the rule base component generates NE labels from lists of NEs, keywords, and contextual rules, while the ML part processes the output of the rule base labeled NE to improve the NER's overall efficiency.

Mandefro [24] developed Afaan Oromo NER corpus of size 27588 words that annotated according to the CoNNL 2002 standard. The corpus classified for training 23,000 words out of which around 3600 are NEs and for the testing purpose 4588 words are used. The corpus annotated by Mandefro without any modification has been utilized [5, 25]. Part of speech (POS) tag used to extract noun which trained and tested on Afaan Oromo corpus size of about 5000 words [5, 24, 25].

3. Proposed Approach

This section presents the detail of the overall approach of Afaan Oromo Named Entity Recognizer (AOroNER). The first part of this section is a general overview of datasets used for training and testing is discussed. The second section illustrates an architecture of the proposed AOroNER system from the perspective of the system's flow of operations from taking input up to giving output. Then a detailed explanation of the study approach is presented. Finally, the paper discusses voting methods of an entity, chunking techniques along with subcomponents in each phase, and the performance measurement is presented.

A. Dataset Preparation and Preprocessing

Afaan Oromo does not have publicly available standard annotated corpus text for NLP tasks including NER [26]. As a result, I collected electronic text from 3 news websites (BBC Afaan Oromo, VOA Afaan Oromo, and Fana Afaan Oromo). The collected articles were further edited manually by removing those sentences which do not have NEs at all, and the corpus contains distinct sentences annotated by the researcher and validated by linguistic. The prepared dataset consists of 44,120 words out of which around 7809 are NEs. The dataset is divided into 2, 90% for training, and 10% for testing purposes.

For the training and testing dataset, an elementary preprocessing is performed. For rule-based, there is no preprocessing except parsing (converting text into a list of sentences) and tokenization will perform on each sentence via white space. For machine learning, all punctuation marks are separated except period for abbreviation and decimal

number as well as mandatory punctuation (e.g. hyphen, apostrophe (') known as "*hudhaa*" which is considered as part of word) if and only if it is written between alphabet. In Afaan Oromo text *Hudhaa* is frequently occurred to specify missed consonants in words. If this character is not considered a part of the word, a single word will be split into (three) tokens with no context. In Afaan Oromo, the word o'clock or hour is *sa'aa*.

The CoNNL 2002 standard is used to annotate datasets for machine learning. The training data are formatted into two columns separated by a tab, which is one token per line and a tab separating it from the line's label. The token is either labeled with a NE (like PER, ORG, etc.) or they will have a default label (of O) to show it's unlabeled. Each statement is separated by a blank line to mark the end of the previous sentence and the start of the next sentence, in both the training and testing dataset.

B. Architecture of AOroNER

The first task of AOroNER is to take input of Afaan Oromo plain text from a user. Once the text is fed preprocessing (like parsing and tokenization) is performed. As the figure shows, the preprocessed text is presented to NE recognizer which is assisted by the ML trained model, the stored rules, and the given pattern to recognize candidates NE before voting best NE type. Combinations of these approaches make a system more efficient and accurate. If a candidate NE is found in more than one entity form, voting is used. It takes the best features of each approach to boost the functionality. In comparison, if a candidate NE is identified out of context due to word ambiguity, Word sense disambiguation penalizes it. Before giving an output chunking is done on subsequent NEs tagged with identical tag merged as a single tag before the final output.

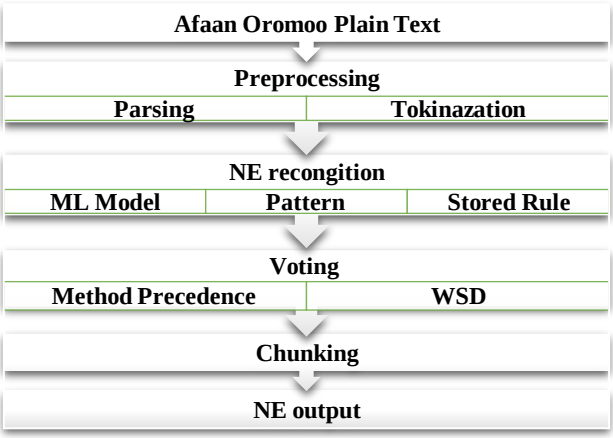

Fig. 1. Architecture of AOroNER

NER, systems have been proposed that use linguistic hand-crafted grammar-based techniques, statistical models (machine learning), and pattern similarity. In the next section, the detail is presented.

C. Rule-based Approach

A rule-based approach is a typical approach to NER. It extracts NEs using lots of human-made (manually written by linguists) rules using grammatical, syntactic, and Capitalization in combination with dictionaries. The rule-based have better capable of detecting complex entities that learning models have difficulty with. However, this approach is usually language and domain-specific and does not necessarily adapt well to new domains and languages . The main advantage associated with rule-based is the ability to extract logic of complex entities and it does not require a large amount of pre-annotated corpus. In contrast, the method needs furthermore the high cost of the maintenance of the rule increases even when the data is slightly changed. Under this section, the sample of rules for each entity category is described.

In Afaan Oromo text, capitalization of the initial letter of a word is an indication of NEs category for a pronoun (person, location, and organization) regardless of their position at the end, middle, or beginning of the sentence except for date/time [25]. The rule-based approach consists of many rules to recognize NEs.

A large number of NEs are classified by a clue prepared in dictionary form to lookup for all seven entities. Some of the most commonly used clue words are *aaddee* (Ms/Mrs), *obbo* (Mr.), *Doktoor* (Dr.), *koloneel* (colonel), etc. for PERSON. *Dhaabbata* (organization), *waajjira* (office), *biiroo* (office), *ejansii* (agency), *warshaa* (factory), *universitii* (university), *inistitiyuutii* (institute), *hoteela* (hotel), *baankii* (bank), *hospitaala* (hospitaala), etc. for ORGANIZATION. *Biyya* (country), *naannoo* (region), *godina*(zone), *aanaa*(woreda), *ganda*(kebele), *magaalaa* (city), etc. for LOCATION.

*Obbo Shimallis Abdiisaan godina Jimmaa aanaa Qarsaatti hojii misoosa garagaraa daawwatan. (*Mr. Shimalis Abdisa visited different development work at Jimma zone Karsa woreda.)

Typically date/time entity starts with words like *secondii(*second*), daqiiqaa(*minute*), sa'aa (*hour/o'cklock*), guyyaa (*day*), torbee (*week*), ji'a (*month*),* bara*(*year*), waggaa(*year*), jaarraa(*century*), ALA(E.C.), ALH(E.C), etc.* or starts with their plural form of thus listed and followed bay cardinal number or followed by multiplicity indicator words like *baay'ee/hedduu (*many*), guutuu (*full*), dheeraa (long), walakkaa (*half*)* or tense indicator words as *dura (*before*), durii, booda (*after*), darbe (last), dhufu (coming), kana* (this*).* If two or more subsequent word fit the above rule automatically thus words classified as date/time entity. Except the first list, all words are naturally either multiplicity indicator or ambiguous. Similarly time name words (*dheengadda* (a day before yesterday), *kaleessa* (yesterday), *har'a* (today), *bor* (tomorrow), *iftaan* (a day after tomorrow)) interchangeably subsequent with other time name (*ganama* (morning), *galgala* (evening), *guyyaa* (noon)). Thus words either in pair or single categorized as date/time entity. Name of day of the week (*Wiixata (*Monday*), Kibxata (*Tuesday*), etc*.) and month (*Fulbaana* (September), *Onkoloolessa (*October*),* etc.) are all part of date/time entity.

e.g. Guyyaa hedduu (many days)
torbee darbe (last week)
bara dhufu (next year)

If the monetary name (such as: *birrii* (birr), *maallaqa* (currency), *qarshii* (birr), *saantima* (cent), *doolara* (*dolar*), *yuuroo* (euro), *paawaundii* (pound), *etc.*) or momentary symbol (£, $, €, ¥) followed by any sort of number (either in form of text or digit) both monetary name and subsequent number detected as monetary NE. The word 'percent' is written in Afaan Oromo as: *parsantii (*percent*), dhibbeentaa (percentile), dhibbentaa (percentile),* or *dhibbarraa (from hundred).* Among these word if it followed by numeric value, the words are classified as percent NE. If the previous token is not detected as date/time, percent, or monetary and the current term is numeric it detect as cardinal.

e.g. Bu'aan Boeing dhibbeentaa 20'n hir'ate (Boeing's profit discount by 20 percent.)

D. Pattern Matching

This section mainly focuses on pattern matching of digits, widely used in various forms for dates, percentages, intervals, identifiers, etc. A digit can be time if the previous token is 'sa'aa' or 'sa'aatii' the successive word contains number-colon-number. If a token starts with a digit and its previous token categorized into date/time the current one is labeled a date/time type. Whenever a token starts with a monetary symbol such as $, €, ¥ and £ followed by number the word is classified as a monetary entity. A percentage is represented by '%' like English but, it comes before the number unlike of English. If the previous word is not detected as either of date/time, percent, or monetary a numeric value is categorized as a cardinal number. *e.g. Festivaalichi ganama keessaa sa'aa 4:30tti kan eegale (*The festival started yesterday morning at 4:30 o'clock). *Karoora qabamerraa %95 milkaa'uu ibsaniiru (Briefed that 95% of plan achieved).*

Table 1. Patterns of tokens

Pattern	Description	Sample
AlphaNumeric	Any sequence of digits followed by letters	10n (*by 10*) 2021tti (*by 2021*)
Numerical	Any sequence of numbers contains colon, comma, periods, or hyphens	2:30 5.5 6-7
Symbol-Numeric	Percentage/Monetary symbol and number	%95 (*95%*) $100 (*100$*)

E. Machine Learning Approach

In this work, a supervised machine learning has been implemented to learn and predict NEs based on probabilistic information of labeled data. It passes through preparing data for training, using probabilistic methods (determine feature weights), and using feature weights to predicts outcomes. Stanford NER is also known as a Conditional Random Field (CRF) Classifier because it is based on a linear chain CRF sequence model. The model is taught using supervised learning, which necessitates providing it with labeled training data. CRF algorithm performs continual feature extraction from the same context for different positions in the input.

Stanford NER properties file is one of a file that defines how the training file structure looks like, the mappings for the words, and its label in the training file. More precisely, a properties file is where parameters of NE of the training data, how the model should learn, and any features you want to specify are defined. Features are properties of a text that are used to provide necessary information associated with a given NE and increase the confidence level of predicting a token as a NE. Properties file determine feature weights using a probabilistic method. In the experiment, this properties file is customized by selecting different features as well as changing their values. Finally, the best features are identified with their parameter.

1) *Encoding*: Encoding is the task of detecting the token boundary and identifying its tag. The training data are formatted into two columns separated by a tab, which is one token per line and a tab separating it from the line's tag that is vital for feature extraction. Here is a sample from the dataset.

Ityoophiyaa	LOC
fi	O
Baankiin	ORG
Daanisk	ORG
waliigaltee	O
liqaa	O
Birrii	MON
biiliyoona	MON
4.49	MON
mallatteessan	O
.	O

2) *Recognition, Detectiona, and Classification of NEs*: Once AOroNER model is built, it is ready to extract NEs from a given Afaan Oromo plain text. Prediction is a process of identifying NEs from the given text. By using the knowledge acquired from the built model, NEs detection and identification are performed based on the calculated probability. The next classification process is preceded by tagging the detected candidate tokens. During classification, each token in the given plain text will be tagged with their possible NE tags using the Stanford scheme. Finally voting for the correct NEs type is performed by comparing the classified candidates from rule-based and pattern match, before chunking is performed.

F. Voting for Classified NEs Candidates

This section discusses how rule-based, machine learning, and pattern matching works together. The machine learning approach recognizes, three entity type (location, organization, and person name).

Pattern matching takes part in detecting date/time, monetary, percent, and cardinal numeric. Whereas the rule-based used to recognize all entity types listed above. Voting for candidate NE focuses on choosing one NE type at a time if a single word is classified as more than one type (for example the prediction of ML model is the difference from rule-based or pattern match) by voting it select one NE type only. For this purpose sense, disambiguation and method precedence are the way. The precedence is given to the rule-based in recognizing all entities and it performs the task of disambiguation for ambiguous words (that changed to different NE categories depending on neighbor words or removed from all category). However when rule-based couldn't filter out, but, machine learning or pattern matching output us used as the second option.

e.g. Baatii darbe (last month) ← date/time

e.g. Leencoo Baatii (Lencho Bati) ← person name

Word Sense Disambiguation WSD1 or WSD2 serve to perform the vote. WSD1—checks the previous word then approves or disproves the current word's tag. WSD2—checks the previous word then approves one tag only.

Word 'hanga' (until) and 'gaafa' () are ambiguous words to be date/time if it followed by date/time NE. Likewise, a word 'kana' and 'sana' not considered as date/time, unless a previous or next of this word is date/time.

e.g. *Sana booda* (After that)

e.g. *Bara kana* (This year)

A phrase *darbee darbee* translated to sometimes is another date/time NE.

Voting algorithm employ as follow:

```
For all lines in the file
    Read a list of words with a tag
        if a word is detected as NE
            looks whether it be by ML/rule-based /pattern
                if recognized by one only (e.g. ML)
                    checks for wsd1
                else
                    if it labeled one entity type (e.g. ORG)
                        WSD1 penalize it
                    else if it labeled two entity type (e.g. ORG & LOC)
                        WSD2 penalize it
        else
            skip and move to the next word
    end
```

G. *Chunking*

The next task is chunking which is combining two or more continuous NEs under a similar entity category to be considered as a phrase and assigned a single tag. Finally, the extraction phase outputs NEs as per their tag category.

4. Result and Discussion

Stanford NER is a Java implementation of a Named Entity Recognizer. It has a lot of features for specifying feature extractors. The feature extractor is responsible for identifying and extracting all the necessary features from the training data. It is one of the essential components that supply necessary information (features) during the building of the model. CRF algorithm holds the actual model to build a custom model from the training data file and outputs the model in a file. The feature extractor is designed to extract features from the training data and store them for future use. All the extracted features are supplied to the model builder which will predict the parameters of the model. In entity recognition, many features such as syntactic rule, capitalization, pattern match, the previous word, next word, dictionary lookup, Sequences, etc. are used. The experiment was conducted on test data of 4412 tokens. The evaluation is performed by comparing the system output to the human-annotated data in terms of three metrics: recall (R), precision (P), and the F1-measure (F) for each entity that the AOroNER recognizes. "These metrics have become standard evaluation method for IR systems" [27]. For each entity, the metrics values are calculated in percentage and averaged to give an overall score on the test data as in Table 2. About 19 parameters or features have participated in these experiments except one parameter that is *wideDisjunctionWidth*=6, the same value (true) used for all.

$$R = \frac{\text{Correct Entities Identified}}{\text{Total Correct Entities}} \tag{1}$$

$$P = \frac{\text{Correct Entities Identified}}{\text{Total Entities Identified}} \tag{2}$$

$$F1 = \frac{2\text{PR}}{\text{P} + \text{R}} \tag{3}$$

Table 2. Parameter used in different experimental of ML

useWord	e1, e2, e3, e4, e5, e6, e7, e8, e9, e10
useNGrams	e1, e2, e3, e4, e5, e6, e7, e8, e9, e10
usePrev	e1, e2, e3, e4, e5, e6, e7, e8, e9, e10
useNext	e1, e2, e3, e4, e5, e6, e7, e8, e9, e10
useDisjunctive	e6, e7, e8, e9, e10
useWideDisjunctive	e6, e7, e8, e9, e10
noMidNGrams	e5, e6, e7, e8, e10
usePosition	e3, e4, e5, e6, e7, e8, e9, e10
useSequences	e2, e3, e4, e5, e6, e7, e8, e9, e10
usePrevSequences	e3, e6, e7, e8, e9, e10
useNextSequences	e3, e6, e7, e8, e9, e10
useTypeSeqs	e5, e7, e8, e10
useTypeSeqs2	e5, e6, e7, e8, e9
useTypeySequences	e6, e7, e9, e10
wordShape	e7, e8, e9, e10
useBoundarySequences	e9, e10
useNeighborNGrams	e8, e9, e10
useWordPairs	e6, e7, e8, e9,e10
wideDisjunctionWidth	e10

Experimentally training a model for seven entity types makes it difficult to scale memory-wise. Three entity type (person, organization and location) that is relatively difficult to recognize than miscellaneous type by rule base and pattern matching approach.

Table 3. The performance of the person, organization, and location in ML top 10 experiments in %

Experiment		e1	e2	e3	e4	e5	e6	e7	e8	e9	e10
PER	P	71.04	72.24	75.05	73	77.25	74	77.2	79	81.03	83.3
	R	57.49	65.86	72	69.4	70.33	70	75.6	78	80.41	82.4
	F1	64.27	69.05	73.53	71.2	73.79	72	76.4	78.5	80.72	**82.85**
ORG	P	66.45	72.03	73.2	73.8	78	73.6	77	78	80	83.44
	R	57.6	66	69	68	70.6	71.3	74.5	75	76.02	78.62
	F1	62.03	69.02	71.1	70.9	74.3	72.45	75.75	76.5	78.01	**81.03**
LOC	P	76.32	78.05	79.5	77	79.68	76.05	80	82.05	82.3	85.37
	R	62.86	66	73.2	69.84	72	73.75	75.9	77	78.5	81.73
	F1	69.59	72.03	76.35	73.42	75.84	74.9	77.95	79.53	80.4	**83.55**

Experiment e1 performed on 4 features (such as: useWord, useNGrams, usePrev and useNext) in e2 to enhance e1 result other feature (useSequences) added. The experiment e3 extended features in e2 with usePosition, usePrevSequences, and useNextSequences that leads F1 result improve. However, in e4 performance decreased when usePrevSequences and useNextSequences removed from the feature take place in experiment e3. In experiment e5 noMidNGrams, useTypeSeqs and useTypeSeqs2 are added on by e4 parameter performed well. In e6 1 parameter value (that is noMidNGrams) changed to false as a result it leads that the performance is less than e5. Experiment e7 shows that the presence of wordShape (e.g. capitalization) plays an essential role in the recognizing of Afaan Oromo NEs. From experimentation e8 to e10 features like useNeighborNGrams, useBoundarySequences, and wideDisjunctionWidth highly increased the performance (F1) for location and organization, and person. Finally, the last experiment (e10) is relatively the best one that is taken from ML and used in the blended model outperformed as in Fig. 2.

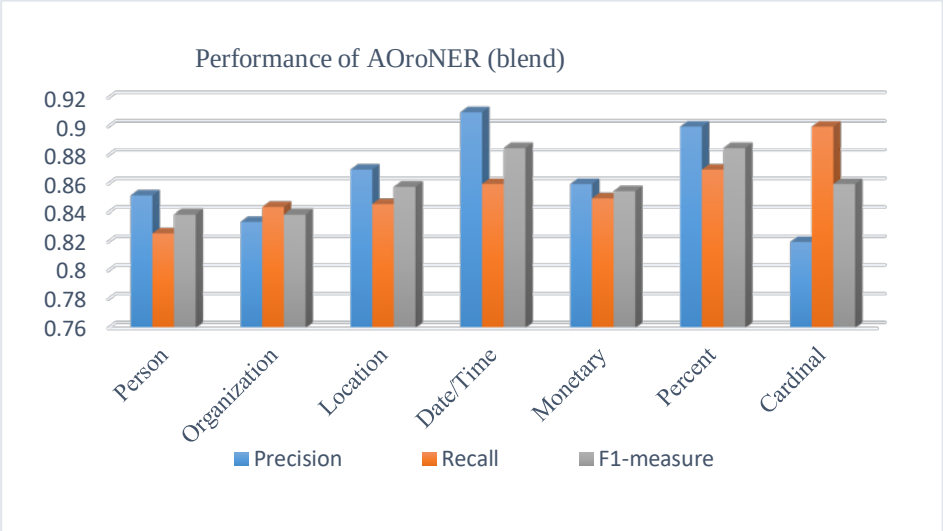

Fig. 2. Performance of AOroNER (blend)

The overall performance of AOroNER is shown in Table 4 Precision, Recall, and F1-measure is 86.37, 85.66, and 86.01 respectively. Clearly, this blend approaches outperform in overall F1-Measure (86.01) as the comparison is shown in Table 2 in percentage.

Table 4. Performance comparison with previous works in %

Researches	**Approaches**	**P**	**R**	**F1**
Ref.[24]	Hybrid	77.84	75.59	76.70
Ref. [25]	Hybrid	84.07	81.21	82.52
Ref. [5]	Rule based	83.33	83.33	83.33
AOroNER	*Blend*	*86.37*	*85.66*	*86.01*

5. Conclusion

A new architecture (AOroNER) is proposed to boost Afaan Oromo NER. The AOroNER has three main processes: the learning processes and prediction, voting and disambiguation of named entity candidates, chunking based on entity category, and merging subsequent words of the same category as a phrase.

The proposed approach integrates ML approaches based on a CRF algorithm, rule-based and pattern matching all together to improve the performance of Afaan Oromo NER during the learning and prediction process. It is capable of recognizing seven named entities location name, organization name, person name, currency, date time, percentage, and cardinal number. Voting and disambiguation are performed by comparing the classified candidates to select the most correct NEs type from the contemporary approaches and to penalize as well. The chunking combining two or more subsequent NEs having a similar entity category to be considered as a phrase and assigned a single tag.

A wide-range experiment has been conducted on a token of around 40,000 on different features to achieve state-of-the-art performance [6]. To summarize the performance of AOroNER in F1-measure: Person, Organization, and Location name entity category are 83.9%, 83.9%, and 85.8% respectively. Likewise, the numeric and temporal expression such as Date/time, Currency, Percent, and Cardinal number is 88.5%, 85.5%, 88.5%, and 86% performed respectively.

From the finding, the addition of deep linguistic knowledge to recognize NER is a significantly rise accuracy of the results. It is especially interesting that using the voting of classified named entity performs better result. I can conclude that rule base and pattern matching is performs well on numeric entities.

Generally, the system outperformed but, there is room to enhance more. I plan a hybrid of unsupervised "zero-shot cross-lingual NER model (assuming no parallel bi-texts, no labels in the target language, no cross-lingual dictionaries, and no comparable corpora)" [18] and rule-based NER to enhance more.

References

[1] C. S. Malarkodi and S. L. Devi, A Deeper Study on Features for Named Entity Recognition, Proc. of the WILDRE5– 5th Workshop on Indian Language Data: Resources and Evaluation, pp. 66–72, 2020.

[2] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, and D. S. Weld, Knowledge-Based Weak Supervision for Information Extraction of Overlapping Relations, In Proc.of the 49th Annu. Meeting of the Assoc. for Comput. Linguistics: Human Lang. Techno., 1, 541-550, 2011.

[3] S. Riedel, L. Yao, and A. McCallum, "Modeling Relations and their Mentions without Labeled Text", In Joint European Conf. on Machine Learn. Knowl. Discovery in Databases, Springer Berlin Heidelberg, 2010.

[4] A. Thomas and S. Sangeetha, "Deep Learning Architectures for Named Entity Recognition: A Survey", Advan. Computing and Intelligent Eng, pp. 2015-2025, 2020.

[5] N. K. Raja, N. Bakala, S. Suresh, "NLP: Rule Based Name Entity Recognition", IJITEE, Vol. 8, no. 11, Sep. 2019.

[6] A. Goyal, M. Kumar, V. Gupta, "Named Entity Recognition: Applications, Approaches and Challenges", Int. J. of Adv. Res. in Sci. and Eng. vol. 6, no. 10, pp. 1902-1919, 2017.

[7] M. Gupta, "Review of Named Entity Recognition (NER) Using Automatic Summarization of Resumes" https://towardsdatascience.com/a-review-of-named-entity-recognition-ner-using-automatic-summarization-of-resumes-5248a75de175 (accessed apr.15, 2021).

[8] A. M.Popescu, and Etzioni, O., Extracting Product Features and Opinions from Reviews, In Natural language processing and text mining, SPRINGER, pp. 9-28, 2007.

[9] O. Etzioni, et al. "Unsupervised Named-Entity Extraction from the Web: An Experimental Study, Artificial intelligence", 165(1), ELSEVIER, pp. 91-134, 2005.

[10] Cao, T. H., Tang, T. M. and Chau, C. K., Text Clustering with Named Entities: A Model, Experimentation and Realization, In Data mining: Foundations and intelligent paradigms, 267-287. Springer Berlin Heidelberg, 2012.

[11] I. Habernal, and M. KonopíK, SWSNL: "Semantic Web Search using Natural Language. Expert Systems with Applications, vol. 40(9), pp. 3649-3664, 2013.

[12] W. Tegegne "The Development of Written Afan Oromo and the Appropriateness of Qubee, Latin Script, for Afan Oromo Writing", Int. Journ. of Computer Appl. Techn and Res., pp 8-14, Vol.28, 2016.

[13] M. Hassen, "A Brief Glance at the History of the Growth of Written Oromo Literature in Cushitic and Omotic Languages" 3rd, Int. Symp., Berlin, 1996.

[14] T. Gamta, "The Oromo language and the latin alphabet", Journal of Oromo Studies, 1992. http://www.africa.upenn.edu/Hornet/Afaan_Oromo_19777.html last visited on Friday, October 31, 2014.

[15] Ws. Li and A. McCallum, "Rapid Development of Hindi Named Entity Recognition using Conditional Random Fields and Feature Induction", 2003.

[16] I. Bedane, "The Origin of Afan Oromo: Mother Language," Glob. J. Hum. Soc. Sci. G Linguist. Educ., vol. 15, no. 12, 2015.

[17] W. Tesema and D. Tamirat, Investigating Afan Oromo Language Structure and Developing Effective File Editing Tool as Plug-in into Ms Word to Support Text Entry and Input Methods.

[18] M. S. Bari, S. Joty, and P. Jwalapuram, Zero-Resource Cross-Lingual Named Entity Recognition, The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20), 2020.

[19] J. Devlin, M. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding, 2018

[20] A. Akbik, D. Blythe, and R. Vollgraf. Contextual string embeddings for sequence labeling. In COLING, pp. 1638–1649, 2018

[21] M.E. Peters, W. Ammar, C. Bhagavatula, and R. Power, Semi-supervised sequence tagging with bidirectional language models, 2017.

[22] J. Xie, Z. Yang, G. Neubig, A. Smith, and G. Carbonell; Neural cross-lingual named entity recognition with minimal resources. 2018

[23] Y. Lin, S. Yang, V. Stoyanov, and H. Ji. A multi-lingual multi-task architecture for low-resource sequence labeling. Association for Computational Linguistics, In ACL, pp. 799–809. Melbourne, Australia: 2018.

[24] M. Legesse, “Named Entity Recognition for Afan Oromo”, M.S. thesis, Addis Ababa Univ., 2012.
[25] A. Sani, “Afan Oromo Named Entity Recognition using Hybrid Approach”, M.S. thesis, Addis Ababa Univ., 2015.
[26] M. Oljira, et al. Sentiment analysis for Afaan Oromo using combined covolutional neural network and bidirectional long-short memory, IJARET, pp. 101-112, 2020.
[27] A. D. Sitter, Calders, T. and W. Daelemans, “A formal framework for evaluation of information extraction”, 2004.

Author’s Profile

Abdo Ababor Abafogi: Received his BSc and MSc in Information Technology, from Jimma University, Ethiopia in 2012 and 2017 respectively. His research interest areas are Artificial Intelligence, Data Science, Deep Learning, Machine Learning and Natural Language Processing, SEO.

How to cite this paper: Abdo Ababor Abafogi, "Boosting Afaan Oromo Named Entity Recognition with Multiple Methods", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.13, No.5, pp. 51-59, 2021. DOI: 10.5815/ijieeb.2021.05.05