AMAZIGH NAMED ENTITY RECOGNITION: A NOVEL APPROACH

AMRI SAMIR, ZENKOUAR LAHBIB
LEC Laboratory, EMI School, Med V University, Rabat, Morocco
E-mail: amri.samir@gmail.com

ABSTRACT

Information Extraction (IE) is a sub discipline of Artificial Intelligence. IE identifies information in unstructured information source that adheres to predefined semantics i.e. people, location etc. Recognition of named entities (NEs) from computer readable natural language text is significant task of IE and natural language processing (NLP). Named entity (NE) extraction is important step for processing unstructured content. Unstructured data is computationally opaque. Computers require computationally transparent data for processing. IE adds meaning to raw data so that it can be easily processed by computers. There are various different approaches that are applied for extraction of entities from text. This paper elaborates need of NE recognition for Amazigh language and discusses issues and challenges involved in NE recognition tasks for Amazigh language. It also explores various methods and techniques that are useful for creation of learning resources and lexicons that are important for extraction of NEs from natural language unstructured text.

Keywords: *Amazigh, Corpus, Named Entity Recognition, Information Extraction, Challenges, NLP.*

1. INTRODUCTION

The progresses in information and communication technology have brought a great increase in the amount of data created and shared, techniques, technologies, and systems to extract value from the data. Data analytics are used for a variety of purposes (business, security and safety, scientific discovery, etc.), domains (biology, medicine, education, etc.), and stakeholders (businesses, governments, scientists, and consumers). Therefore, extracting information and value from data has become critical for academia, the industry, and governments.

Named Entity Recognition (NER) is one of the important parts of Natural Language Processing (NLP). NER is supposed to find and classify expressions of special meaning in texts written in natural language. These expressions range from proper names of persons or organizations to dates and often hold the key information in texts. NER can be used for different important tasks. It can be used as a self-standing tool for full-text searching and filtering. Also it can be used as a preprocessing tool for other NLP tasks. These tasks can take advantage of marked Named Entities (NE) and handle them separately, which often results in better performance. Some of these tasks are Machine Translation, Question Answering, Text Summarization, Language Modelling or Sentiment Analysis.

Furthermore, our work of Amazigh NER is considered crucial, it assist in improving the performance of Natural Language Processing (NLP) applications in Amazigh language. For instance, when executing tasks related to handling massive amounts of information, NER systems could help in Information Extraction (IE), Information Retrieval (IR) and Question Answering (QA) tasks.

In the general domain, NER focuses on identifying names of persons, locations, and organizations in news articles, reports, and even tweets. Thanks to the availability of annotated corpora, supervised learning methods have been widely adopted and prevail unsupervised ones. Such state-of-the-art NER systems have achieved performance as high as human annotators. On their side, NER systems are getting better with the advant of more annotated corpora to learn from. Traditional ways of tackling NER range from dictionary matching, heuristic rules, to supervised Hidden Markov Models (HMMs) / Conditional Random Fields(CRFs)-based sequence labeling. The first two approaches do not require training data, but usually involve ad-hoc rules and assumptions that may limit the type of entities and texts to which they could apply. CRF-based labelers have yielded high performance

in sequence learning tasks, and are the state of the art for some entity recognition tasks. However, the supervised nature of CRF entails a fairly large amount of training data which must be annotated by humans. As a result, it is only applicable in a limited number of settings.
The best NER systems for English produce near-human accuracy. One such system can do so at a level of roughly 93.39% accuracy, whereas a human would achieve roughly 97% accuracy. Amazigh language, however, provides some unique challenges to overcome.

This paper is organized as follows: Section 2 contains the details about language background; Section 3 presents NER approaches; Section 4 describes the challenges of Amazigh NER; Section 5 contains the description of our approach; Section 6 is devoted to the required linguistic resources for Amazigh NER; Section 7 describes and evaluates the system's performance. Finally, Section 8 draws conclusions from this work and presents suggestions for future research.

2. LANGUAGE BACKGROUND

2.1 Amazigh Language

Amazigh called also Berber belongs to the Hamito-Semitic "Afro-Asiatic" languages [1, 2]. Amazigh is spoken in Morocco, Algeria, Tunisia, Libya; it is also spoken by many other communities in parts of Niger and Mali. In Morocco, Amazigh language uses different dialects in its standardization (Tachelhit, Tarifit and Tamazight). The morphological word classes in Amazigh are the noun, the verb, and the particles (that includes all other morphosyntactic categories other than noun and verb) [3, 4]. Amazigh NLP presents many challenges for researchers.

2.2 Amazigh Script

In Morocco, IRCAM has developed an alphabet system called Tifinaghe-IRCAM. This alphabet is based on a graphic system towards phonological tendency. This system does not retain all the phonetic realizations produced, but only those that are functional. It is written from left to right and contains 33 graphemes which correspond to:

- 27 consonants including: the labials (ⴼ, ⴱ, ⵎ),dentals (ⵜ, ⴷ, ⵟ, ⴹ, ⵏ, ⵔ, ⵕ, ⵍ), the alveolars (ⵙ,ⵣ, ⵚ, ⵥ), the palatals (ⵛ, ⵊ), the velar (ⴽ, ⴳ), the labiovelars (ⴽⵯ, ⴳⵯ), the uvulars (ⵇ, ⵅ, ⵖ), the pharyngeals (ⵃ, ⵄ) and the laryngeal (ⵀ);
- 2 semi-consonants: ⵢ and ⵡ;
- 4 vowels: three full vowels ⴰ, ⵉ, ⵓ and neutral vowel (or schwa) ⴻ.

Correspondences between the different writing systems and transliteration correspondences are shown in Table 1.

Table 1. Mapping from existing writing system and the chosen writing system

Tifinaghe Unicode		Transliteration		Chosen writing system
Code	Character	Latin	Arabic	
U+2D30	ⴰ	A	ا	A
U+2D31	ⴱ	B	ب	B
U+2D33	ⴳ	G	گ	G
U+2D33 &U+2D6F	ⴳⵯ	Gw	گ	Gw
U+2D37	ⴷ	D	د	D
U+2D39	ⴹ	ḍ	ض	D
U+2D3B	ⴻ	E		E
U+2D3C	ⴼ	F	ف	F
U+2D3D	ⴽ	K	ک	K
U+2D3D &+2D6F	ⴽⵯ	Kw	گ +	kw
U+2D40	ⵀ	H	ه	H
U+2D43	ⵃ	ḥ	ح	H
U+2D44	ⵄ	E	ع	E
U+2D44	ⵅ	X	خ	X
U+2D45	ⵇ	Q	ق	Q
U+2D47	ⵉ	I	ي	I
U+2D47	ⵊ	J	ج	J
U+2D47	ⵍ	L	ل	L
U+2D47	ⵎ	M	م	M
U+2D47	ⵏ	N	ن	N
U+2D47	ⵓ	U	و	U
U+2D47	ⵔ	R	ر	R

U+2D47	ⵕ	ṛ	ر	R
U+2D47	ⵖ	Y	غ	G
U+2D47	ⵙ	S	س	S
U+2D47	ⵚ	ṣ	ص	S
U+2D47	ⵛ	C	ش	C
U+2D47	ⵜ	T	ت	T
U+2D47	ⵟ	ṭ	ط	T
U+2D47	ⵡ	W	ۋ	W
U+2D47	ⵢ	Y	ي	Y
U+2D47	ⵣ	Z	ز	Z

2.3 AMAZIGH MORPHOLOGY

The Amazigh language presents a rich morphology; the words can be classified into different grammatical classes which we cite: the noun, the verb and particles. In this paper, we are interested in noun morphology.

The Amazigh noun is always composed of one word between two spaces and formed from a root and a pattern. It is characterized by gender (masculine or feminine), number (singular or plural), and state (free or construct).

- Gender: the Amazigh noun is characterized by one of grammatical gender: masculine or feminine.
- Number: the noun, masculine or feminine, has a singular and plural. This latter has four forms: the external plural, broken plural, mixed plural and plural in ⵉⴷ [id]. The external plural: is formed by an alternation of the first vowel ⴰ/ⵉ [a/i] accompanied by a suffixation of ⵏ [n] or one of its variants. The broken plural: involves a change in the vowels of the noun. The mixed plural: is formed by vowels' change accompanied, sometimes by the use of the suffixation by ⵏ [n]. The plural in ⵉ ⴷ [id]: this kind of plural is obtained by ⵉⴷ [id] prefixing. It is applied to a set of nouns including: nouns with an initial consonant, proper nouns, parent nouns, compound nouns, numerals, as well as borrowed nouns.
- State: we distinguish between two states: the free state and the construct one. The free state: is unmarked. The noun is in free state if it is: a single word isolated from any syntactic context, a direct object, or a complement of the predictive particle ⴷ [d]. The construct state: involves a variation of the initial vowel. In case of masculine nouns, it takes one of the following forms: initial vowel alternation ⴰ [a] /ⵓ [u] or adding of ⵡ [w]; adding of ⵢ [y] to the nouns of vowel ⵉ [i]. For the feminine nouns, it consists to drop the initial vowel or maintaining of this vowel.

3. NER APPROACHES

3.1 Named Entity Types

Named Entities (NEs) play a central role in conveying important domain specific information in text, and good named entity recognizers are often required in building practical information extraction systems. There are no general types of NE that are commonly used across all languages. As a result of this, an NER system can recognize differs from language to language or from domain to domain. This feature is quite variable due to the ambiguity in the use of the term Named Entity depending on the different forums or events. There are conferences or contests that are organized to define the types of NEs and evaluate the performance of a given NER system developed for a language. The conferences are, for example, Message Understanding Conference (MUC) for English, Conferences on Natural Language Learning (CoNLL), a language independent NER task and Information Retrieval and Extraction Exercise (IREX) for Japanese. The corresponding NE types for MUC and CoNLL are shown in Table 2 & 3 respectively.

Table 2: Named Entity Types as defined by MUC

Named Entity	Example
PERSON	Smith, Obama
ORGANIZATION	IBM, General Motors
LOCATION	Rabat, New Jersey
DATE	20/02/2017, October 17
TIME	10:30 AM
PERCENTAGE	15%
MONETARY AMOUNT	$75.00, 500 DHs

Table 3: Named Entity Types as defined by CoNLL

Named Entity	Example
PERSON	Smith, Obama
ORGANIZATION	IBM, General Motors
LOCATION	Rabat, New Jersey
MISCELLANEOUS	50 Euro, 19:00 GMT

3.2 NER Approaches

It is important to acknowledge that the Named Entities Recognition approaches identify the class of the named entities present in the text. And by doing that we can search for references about the named entity identified in other resources available on the web for example. Also, if we find the named entity New York, the named entity recognition system would identify it as being of the class "Location". With that information we could look for a reference of the named entity found in the Google Maps and present it to the user. Another example can be made with the named entity "President Obama". The named entity recognition system would identify it as being of class "person" and by doing that; we can look for President Obama's reference in the Facebook or on Twitter. This kind of operation or behavior is illustrated on the Figure 1.

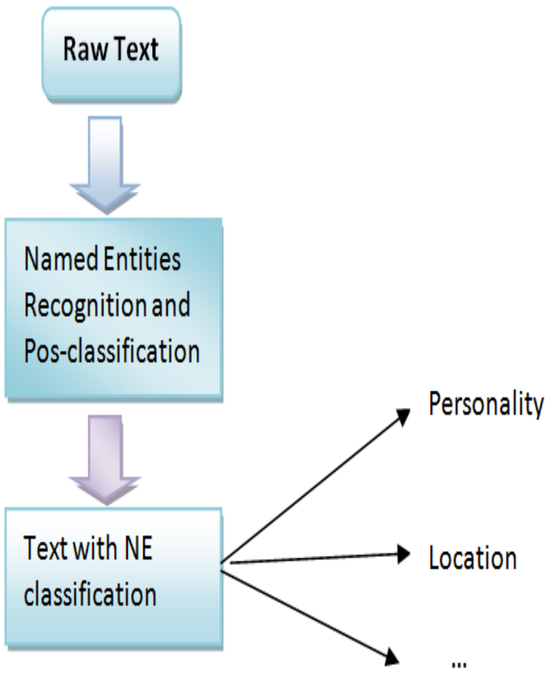
Raw Text
Named Entities Recognition and Pos-classification
Text with NE classification
Personality
Location
...

Figure 1. The generic approach of NE Recognition

Algorithms for named-entity recognition (NER) systems can be classified into three categories; rule-based, machine learning and hybrid [6].

- A Rule-Based NER algorithm detects the named entity by using a set of rules and a list of dictionaries that are manually pre-defined by human. The rule-based NER algorithm applies a set of rules in order to extract pattern and these rules are based on pattern base for location names, pattern base for organization name and etc. The patterns are mostly made up from grammatical, syntactic and orthographic features [5]. In addition to that, a list of dictionaries is used to speed up the recognition process. However, the types of dictionaries affect the performance of the NER systems and these dictionaries normally include the list of countries, major cities, companies, common first names and titles [6].

- A machine-learning NER algorithm normally involves the usage of machine learning (ML) techniques and a list of dictionaries. There are two types of ML model for the NER algorithms; supervised and unsupervised machine learning model. Unsupervised NER does not require any training data [7, 8]. The objective of such method is to create the possible annotation from the data. This learning method is not popular among the ML methods as this unsupervised learning method does not produce good results without any supervised methods. Unlike unsupervised NER methods, supervised NER methods require a large amount of annotated data to produce a good NER system. Some of the ML methods that had been used for NER algorithm include Artificial Neural Network (ANN) [9], Hidden Markov Model (HMM) [10], Maximum Entropy Model (MaxEnt) [11], Decision Tree [12], Support Vector Machine [13] and etc. ML methods are applicable for different domain-specific NER systems but it requires a large collection of annotated data. Hence, this might require high time-complexity to preprocess the annotate data.

- A hybrid named entity recognition algorithm implements both the rule-based and machine learning methods [14]. Such method will produce a better result. However, the weaknesses of the rule-based are still unavoidable in this hybrid system. A domain-specific NER algorithm may need to customize the set of rules used to recognize different types of named entity when the domain of studies is changed.

4. CHALLENGES AND GOALS OF AMAZIGH NER

We live in the Information Age. In every moment, an enormous amount of information is generated on the Internet, adding to its already gigantic size. Access to such a massive amount of information has totally changed the way we work and study. For organisations, possession and effective utilisation of information is deemed as a key part of strategic competitiveness. On the other hand, the scale and the scope of the information that one has to deal with at a time are also unprecedented, which makes locating useful pieces of information extremely difficult. The amount of accessible information would not be of much use if there were no suitable techniques to process it and extract knowledge from it. The answer to this challenge is the technology of Information Extraction (IE), the technique for transforming unstructured textual data into structured representation that can be understood by machines. IE has been an active research field for decades, involving many sub-topics that are addressed by rigorous communities. It originates from a set of earlier competitions organised within the Natural Language Processing (NLP) community.

So, this paper of Amazigh NER is dedicated to the important problems of NER and disambiguation. We use a hybrid method to identify named entities in Amazigh texts. The main contribution of this work to state-of-the-art is the experiments with different morphological features for machine learning as well as feature combination. We formulate four main goals of this work:

1- Develop new recognition methods and features to improve performance for Amazigh language.
2- Propose hybrid approach to improve the adaptability of Amazigh NER.
3- Experiment with disambiguation on small subset of selected named entities.
4- Create quality and reusable NER system.

This work is focused mostly on the Amazigh language and its peculiarities from the point of view of the NER task.

In Amazigh NE, we addressed many challenges posed by the particularities of the Amazigh language, which is significantly different from the other European languages. We review below some issues that need to be taken into consideration when building a NER system for Amazigh.

- Ambiguity: Many words can be interpreted in multiple ways, producing different meanings. In order to alleviate the impact of this issue, contextual information will be used in our system.

- Absence of capital letters. Unlike Latin script languages, Amazigh does not distinguish upper and lowercase letters (uppercase helps to identify the beginning and end of potential NEs in most Latin script languages).

- Complex morphology. The Amazigh language has a very systematic but complex morphological structure based on root-pattern schemes and is considered a highly inflectional language. Usually a given lemma in Amazigh could have more than one word form which includes a root, prefixes, suffixes, and clitics. This issue should be dealt with in order to detect correctly the NEs in the text.

- Lack of standardization of the Amazigh spelling. Amazigh text, like many other languages, has many spelling variants when it comes to proper names and especially foreign names, which may lack a standardized spelling.

- Lack of Linguistic Resources: We lead study on the Amazigh language resources and NLP tools (e.g., corpora, gazetteers, POS taggers, etc.). This led us to wrap up that there is a limitation in the number of available Amazigh linguistic resources in comparison with other languages. Many of those available are not relevant for Amazigh NER tasks due to the absence of NEs annotations in the data collection. Amazigh gazetteers are rare as well and limited in size. Therefore, we tend to build our Amazigh linguistic resources in order to train and evaluate Amazigh NER systems.

5. OUR APPROACH OF AMAZIGH NER

Named entity recognition is a challenging task which needs massive prior knowledge sources for better performance. Many researches works have been conducted in different domains with various approaches. Early studies focus on heuristic and handcrafted rules. By defining the formation patterns and context over lexical-syntactic features and term constituents, entities are recognized by matching the patterns against the input documents. Rule-based system may achieve high degree of precision. However, the development process is time-consuming and porting these developed rules from one domain to another is a major challenge. Recent research in NER tends to use machine learning approaches. The learning methods include various supervised, semi-supervised and unsupervised learning. The supervised learning tends to be the dominant technique for named entity recognition and classification. However, supervised machine learning methods require large amount of annotated documents for model training and its performance typically depends on the availability of sufficient high quality training data in the domain of interest. There are some systems which use hybrid methods to combine different rule-based and/or machine learning systems for improved performance over individual approaches. Hybrid systems make the best use of the good features of different systems or methods to achieve the best overall performance.

5.1 Typical NER System

A typical named entity recognizer has four core elements regardless of whether it is designed according to rule-based approach or automatic machine learning approach. The architecture of a typical NER system is shown Figure 2. The core elements are Tokenization, Morphological and Lexical processing, Identification and Classification. Tokenization is the first step in interpreting text by splitting up a string of words/characters (comprising a document, paragraph or sentence) into minimal parts of structured text that are useful to be used as a unit, referred to as a token. With regard to NER, tokenization can consist of sentence splitting and word segmentation as a subtask. After tokenization process is completed, morphological and lexical processing proceeds. During this step, word tokens in a document are sequentially tagged as being inside or outside of a given named entity. It mainly employs part-of-speech tagger and each word in a sequence of words is labeled with an inside or outside tag. In addition, it employs components like NP chunking and feature extraction. The morphological and lexical processing mainly helps for the detection of NEs which is depicted in the Figure 1 as identification. The identification component detects NEs with the aid of stored models or rules, based on the approach used. The detected NEs are then ready to be classified into their respective classes. The classification step takes the detected NEs and categorizes them into their corresponding categories. The classification is done by a classifier.

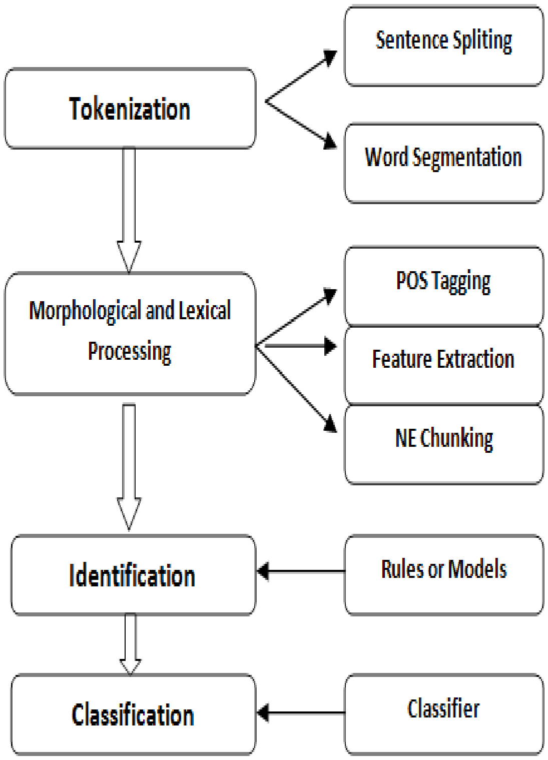

Figure 2. Architecture of a typical NER System

5.2 Our Approach

Our work followed a hybrid approach with the machine learning component based on a CRF algorithm. The system's architecture has two main processes (Figure 3): the learning and prediction processes.

- The learning process works on the training data and is used to generate the trained model.
- The prediction process is a process which works on the input text supplied by a user and is aimed at recognizing NEs from the text.

The system also has three main phases:

- The pre-processing phase is where both the training data and the plain text are pre-processed for the next task. Tokenization and segmentation is the main pre-processing tasks with the former applying to the corpus and the later to the plain input text.
- The training phase comprises essential components that are used to generate token/tag sequence, extract features, chunk the tokens and estimate the model with the training data.
- The recognition phase is where the pre-processed input text is an input and NE

recognized text is an output and it consists of components like the trained model, stored rules and NE Recognizer.

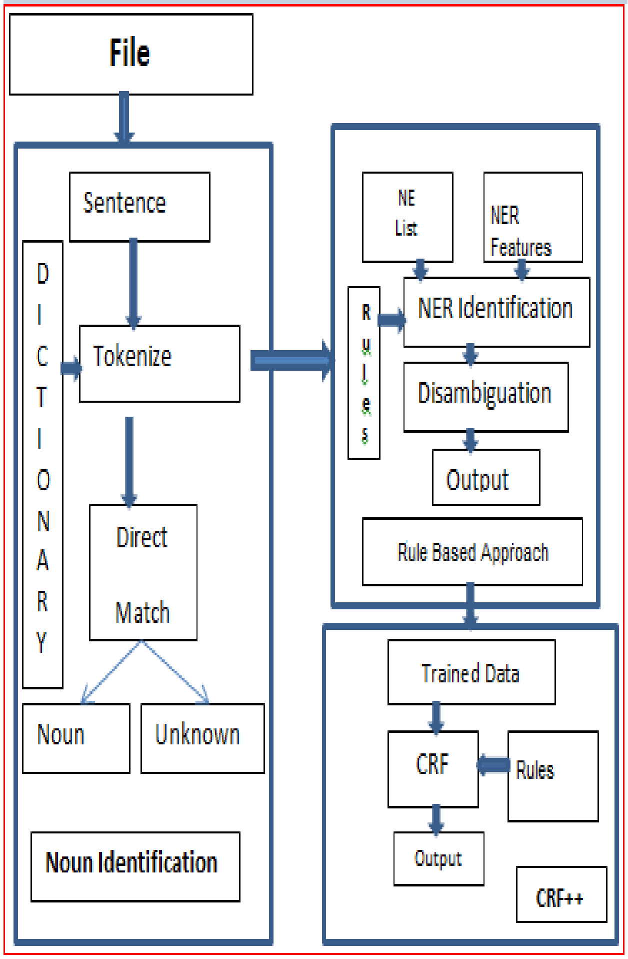

Figure 3. NER Architecture

Step 1: divide input file into sentences
Step 2: Tokenization
Step 3: If tokens directly match with dictionary, assign as a noun
Step 4: If the noun match with NER list, then assign its tag, otherwise use NER features and Disambiguation rules
Step 5: Still have ambiguity and unknown words, and then go for

5.3 Conditional Random Fields

Conditional Random Fields (CRF) were introduced in [15]. The idea of CRF is strongly based on ME. The difference is that ME classifies one instance after another while CRF classify the whole sequence at once. Mathematically written, ME estimates p(yi/xi) for i = 1; : : : ; n and CRF estimate p(y/x) where y and x are n-dimensional vectors. The probability p(y/x) can be computed using matrices and a variant of forward-backward algorithm.

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_j \lambda_j f_j(\mathbf{y}, \mathbf{x})\right)$$

The features are extended and can use the previous state in contrast to ME. Two types of features are used, state s and transition t. The state features can be considered as a subset of transition features, where the previous state is not used, and a general feature definition can be used.
Initial tests on the NER task were done in [16]. Since their introduction, many systems used them with very good results [17, 18]. CRF are considered to be the most successful classification method for NER.

5.4 Used Algorithms

5.4.1 Algorithm for Noun Identification

Our algorithm for noun identification is outlined below. We assume that we have a small amount of labeled data and a classifier that is trained on this Amazigh data. We exploit a large unlabeled corpus from the test domain from which we automatically and gradually add new training data, such that our corpus has two properties:

i) Accurately labeled, meaning that the labels assigned by automatic annotation of the selected unlabeled data are correct.

ii) Non redundant, which means that the new data is from regions in the feature space that the original training set does not adequately cover. Thus the classifier is expected to get better monotonically as the training data gets updated.

Step I: Read input file and break into sentences
Step2: Read each sentence and break into tokens
Step3: Read each token
Step4: For each (token) Loop check with Amazigh dictionary
Step5: If direct match with dictionary then assign noun
Step6: else if no match with dictionary then check with the suffix list of nouns
Step7: if suffixes are found and root is found in Amazigh dictionary then assign noun
Step8: else if suffix matches and root is not found then token may be a noun
Step9: else assign the category "unknown" End loop

5.4.2 Algorithm for NER Identification

Since the features of each token include the features copied from its neighbors, in addition to those extracted from the token itself, its neighbors need to be added to the training set also. If the confidence of the neighbors is low, the neighbors will be removed from the training data after copying their features to the token of interest. If the confidence scores of the neighbors are high, we further extend to the neighbors of the neighbors until low-confidence tokens are reached. We remove low-confidence neighbors in order to reduce the chances of adding training examples with false labels.

```
Step 1: Read list of nouns identified
Step2: Check gazetteer lists for NER features
Step3: For each (noun) Loop If suffix features found then assign NER tag
Step4: else if prefix features found then assign NER tag
Step5: else if context features found then assign NER tag
Step6: else if found in NER list then assign NER tag
Step7: else assign "Miscellaneous word"
Step8: If ambiguity is found then Call disambiguation rules Remove ambiguity
Step9: Else if still ambiguity and unknown words found then Call CRF End Loop
```

6. LINGUISTIC RESOURCES FOR AMAZIGH NER

6.1 Corpus

First of All, The building of a named entity extraction system requires collecting a sufficient number of texts that will serve, not only as a training corpus (to establish rules), but also as a test corpus. As we have mentioned before, there is non available Amazigh corpus for NER task. For this reason, we built our own corpus. It contains the Regional news (11 articles), Economics (27 articles), Social (31), Politics news (25), Sport (33), world activities (23 articles) and some general news (36 articles). Thus, we have collected 402 articles from these categories in html format and we have concatenated all of them in one text file. It contains 78 220 tokens.

6.2 Gazetteers Built

Gazetteers are lists of NEs. The opinions on gazetteers are mixed. Some authors stated that usage of gazetteers has not improved their results while other says it improved it significantly.

Our Amazigh NER system gathers four different manually built gazetteers:

- Person gazetteer: we have built a list of about 1120 entries of Amazigh names and foreign names transcribed in Amazigh, extracted from our corpus and Internet resources.
- Location gazetteer: We consider the type "location" or place name as: countries, cities, rivers, mountains, oceans and seas. Thus, we developed a lexicon containing 2083 entries, found in internet, and extracted from our corpus.
- Organization gazetteer: The organization lexicon is limited to a list of 330 company and organization names that we extracted from the web and our corpus.
- Miscellaneous (MISC): we have integrated in this class day and months (this contains 19 entries), numbers transcribed in Amazigh language (this contains 87 entries).

6.3 Trigger words

Trigger words are words which are not NEs, but are often in the neighbourhood of NEs. For example “rays” (‘president’ in English) can be a trigger word for Person NE.

List of trigger words can be automatically learned from corpora or can be made by hand.

Triggering properties of window words (e.g., W(3): trig=PER). Triggering properties of components of the NE being classified (e.g., for the entity “banka n lmaghrib” (Bank of Morocco) we could have a feature NE(1): trig=ORG). Context patterns to the left of the NE, where each word is marked with its triggering properties, or with a functional–word tag if appropriate (e.g., the phrase (rrays n marikan) “the president of United States”, would produce the pattern f ORG f for the NE “United States”, assuming that the word “rrays” (president) is listed as a possible trigger for ORG).

7. RESULTS AND DISCUSSION

The corpus was split into 90% for the training set and remaining set is for testing where training set represents the input values for the classification model of CRF. Moreover, the corpus represents the data entries in this model. The aim of the experiments presented here is to evaluate the performance of our hybrid approach.

The total corpus file is divided into four files out of which three files are to be used as per the rules and regulations of the CoNLL 2002 shared task [19]. The four files are named as the training file, the development file, the test file and the experimentation file. The learning methods are trained with the training data. The data in the development file is used for tuning the parameters of the learning methods. When the best parameters are found, the method can be trained on the training data and tested on the test data. Here, the split between development and test data has been chosen to avoid that systems are tuned to the test data. The experimentation file is used later in the experimentation. The statistics of the developed Amazigh NE corpus is given in Table 4.

Table 4: Statistics of Amazigh NE corpus

	Total Size	Total NE	PER	LOC	ORG	MISC
Training File	8000	1206	411	266	296	233
Development File	5000	745	167	248	183	147
Test File	4000	704	234	235	91	144
Experiment File	6100	920	252	346	221	101

NER systems are commonly evaluated using three evaluation metrics: precision, recall and F-measure. All of them are represented in the form of percentage:

- Precision (P) calculates the percentage of correctly recognized NEs out of the total recognized NEs.
- Recall (R) calculates the percentage of the recognized NEs from the reference set.

Three values can help us easily calculate the evaluation metrics for our system (Fig 4). These are the True Positive (tp), False Positive (fp) and False Negative (fn).

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Fig 4. NER evaluation metrics

- tp counts the number of NEs that are recognized by an NER system and are found in the test data.
- fp counts the number of NEs that are wrongly recognized by an NER system but are not in the test.
- fn counts the number of NEs that are left unrecognized by an NER system but are in the test data.

The developed NER system has been designed according to the CoNLL2002 and CoNLL2003 shared task tag set definition and is formed by tags falling into the following four categories

1- Person's Names: our annotated corpus contained 1120 occurrences of person's names. Of these, 830 person names were annotated correctly; 30 were partially correct annotations; 231 person names were not annotated and there were 29 false positives. The errors were in large part due to: i) Person‘s first and last names that appeared in the corpus, but were not included in word lists. ii) Some terms in Amazigh language are person names may also be city names.

2- Locations: There were 2042 location names in the corpus. The annotation method correctly identified 1905 of these. However, 51 location names were missed, 56 of the annotations were only partially correct, and there were 30 false positives. The lack of standards for writing location names involves the difficulties in recognizing location named entities.

3- Organizations: There were 622 organization names in the corpus. 295 organization names were correctly annotated, 272 were partially correct, 53 organization names were not recognized, and there were 5 false positives. This is mainly due to the fact that we use the delimitation of the NE’s using contextual information only which is not sufficient in NE task.

4- Miscellaneous Name: denotes the miscellaneous NEs which do not belong to any of the previous classes and include date, time, number, monetary expressions, measurement expressions and percentages. There are 828 date/Number expressions in the corpus. Of these, 320 were correctly annotated, 470 were partially correct, and 38 were missed. There were 0 false positives.

Using the three metrics, the results of our Amazigh named entity recognition for each type of entity are presented in Table 5 and Fig 5.

Table 5. System’s performance

Named Entity	Precision	Recall	F-Measure
Person	93	96	83
Location	97	97	97
Organization	74	77	76
Miscellaneous	65	68	67

A closer look at the erroneously identified examples showed the possibility of some improvements in the implementation of the algorithm, which would result in a further improvement in precision. However, this is beyond the aim of this paper and therefore, we did not do it. The precision results were very satisfactory: extraction precision for the Person category was 93 %, for the Organization category 74%, for the Location category 97% and for Miscellaneous category 65%.

Manually tagging entities of all types in such large set of documents and comparing these entities to those found automatically by the extractor would require a lot of work. Since this will only calculate an estimate of recall, and it was not the main goal of this project.

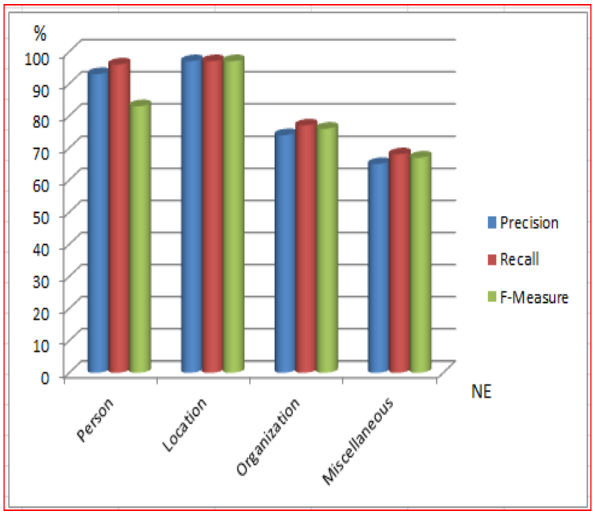

Fig 5. Amazigh NER Performance

The main goal of our work was to build very high-precision Amazigh entity extractors for the Person, Location, Organization and Miscellaneous categories that would minimize the noisy output (entities and their relationships). We used the four specific categories, they are among the top most heavily utilized categories in information retrieval systems across domains, thus they can be used to further improve the NER system's precision and expand its scope through machine learning.

Our proposed method ensures that each extractor is specialized in one and only one category: if the rules of a specific classifier do not recognize an entity, it will be ignored and not extracted at all instead of being misclassified. Even though missing a considerable number of potential entities will lower the extractor recall, we have designed a second step for ML by CRF technique that will counteract this weakness and improve recall.

The quality of our extraction system, however, depends on the quality of the NE lists. If the categories do not have a finite number of "members," our method would not achieve similar high-precision results. Creation of such lists requires research and time and could vary from one language to another. This method could be a challenge for very large data intensive systems. It would not be a very difficult task, however, to take an Amazigh list of NE and find the equivalent list in other languages

Our approach is simple enough that it can be used for different languages other than Amazigh language. Lastly, since our method does not require ML or training of any sort in the first stage, it can be applied across application areas without the need for any major changes. However, building separate, statistically based models for each application area would be needed.

8. CONCLUSION AND PERSPECTIVES

NER system for Amazigh language is difficult and challenging because of various issues like the inherent agglutinative and inflectional nature of Amazigh, ambiguities in named entity classes, non local dependencies, appearances of foreign words, spelling variations etc. This paper has explored various methodologies and techniques that may be used in designing Amazigh named entity recognition system.

The work has presented an attempt to develop the named entity recognition model for Amazigh language using hybrid technique. The aim of this model is to improve the precision of NER in Amazigh language introduced by different approaches in the literature. A hybrid approach has been adopted in this research. The corpus was split into 90% for the training set and the remaining set is used for testing where the training set represents the input values for the classification model of CRF. The result showed that our approach overcomes other methods in their performances and in terms of accuracy. The hybrid approach achieves 93%, 97%, 74% and 65% for precision in Person, Location, Organization and Miscellaneous respectively.

Future development will involves adding grammar rules in order to get a higher score. Another area which needs further research is unsupervised and semi supervised methods for Amazigh NER.

REFERENCES:

[1] A.Boukous, Société, langues et cultures au Maroc: Enjeux symboliques, Casablanca, Najah El Jadida, 1995.

[2] S. Chaker , “Le berbère”, Actes des langues de France, pp. 215-227, 2003.

[3] F. Boukhris, A. Boumalk, E. Elmoujahid, H. Souifi, La nouvelle grammaire de l'amazighe, Rabat, Maroc: IRCAM, 2008.

[4] M. Ameur, A. Bouhjar, F. Boukhris, A. Boukouss, A. Boumalk, M. Elmedlaoui, E. Iazzi, H. Souifi, Initiation à la langue amazighe. Rabat, Maroc: IRCAM, 2004.

[5] A. Mansouri, L. S. Affendy, and A. Mamat, “Named Entity Recognition Approaches,” International Journal of Computer Science and Network Security, vol. 8, no. 2, pp. 339-344, 2008.

[6] W. Takahiro, G. Robert, and W. Yoricks, “Evaluation of an algorithm for the recognition and classification of proper names,” in Proc. the 16th International Conference on Computational Linguistics (COLING), vol. 1, 1996, pp. 418-423.

[7] C. Micheal and S. Yoram, “Unsupervised models for named entity classification,” in Proc. the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999, pp. 100-110.

[8] J. H. Kim, I. H. Kang, and K. S. Choi, “Unsupervised name entity classification models and their ensembles,” in Proc. the 19th International Conference on Computational Linguistics (COLING), 2002, vol. 1, pp. 1-7.

[9] F. M. Naji and O. Nazlia, “Arabic named entity recognition using artificial neural network,” Journal of Computer Science, vol. 8, issue 8, ISBN 1549-3636, Science Publications, pp. 1285-1293, 2012.

[10] M. B. Daniel, M. Scott, S. Richard, and W. Ralph, “Nymble: a high-performace learning name-finder,” in Proc. the Fifth Conference on Applied Natural Language Processing (ANLC), pp. 194-201, 1997.

[11] B. Yassine, R. Paolo, and M. B. Jose, “ANERsys: An Arabic Named Entity Recognition System Based on Maximum Entropy,” in Proc. the 8 th International Conference on Computational Linguistics and Intelligence Text Processing (CICLing), 2009, pp. 143-153.

[12] B. Frederic, N. Alexis, and G. Franck, “Tagging Unknown Proper Names using Decision Trees,” in Proc. the 38th Annual Meeting on Association for Computational Linguistics (ACL), 2000, pp. 77-84.

[13] Y. C. Wu, T. K. Fan, Y. S. Lee, and S. J. Yen, “Extracting named entities using support vector machines,” in Proc. Knowledge Discovery in Life Science Literature, PAKDD 2006 International Workshop, KDLL 2006, vol. 3886, Springer Berlin Heidelberg, pp. 91-103, 2006.

[14] S. Rohini, N. Cheng and L. Wei, “A Hybrid Approach for Named Entity and Sub-Type Tagging,” in Proc. the 6th Applied Natural Language Processing Conference, 2001, pp. 247-254.

[15] John D. La_erty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random _elds: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, pages 282{289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[16] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random _elds, feature induction and web-enhanced lexicons. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4, pages 188{191, Morristown, NJ, USA, 2003. Association for ComputationalLinguistics.

[17] Yassine Benajiba, Mona Diab, and Paolo Rosso. Arabic named entity recognition using optimized feature sets. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08, pages 284{293, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[18] Georgi Georgiev, Preslav Nakov, Kuzman Ganchev, Petya Osenova, and Kiril Simov. Feature-rich named entity recognition for bulgarian using conditional random _elds. In Proceedings of the International Conference RANLP-2009, pages 113{117, Borovets, Bulgaria, September 2009. Association for Computational Linguistics.

[19] F. Erik and Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 Shared Task: Language Independent Named Entity Recognition. In: Proceedings of CoNLL-2002, Taipei, Taiwan, pp. 155-158.