information

Article

Enhancing the Performance of Telugu Named Entity Recognition Using Gazetteer Features

SaiKiranmai Gorla *, Lalita Bhanu Murthy Neti and Aruna Malapati

Department of Computer Science and Information Systems, Birla Institute of Technology and Science Pilani, Hyderabad Campus, Telangana 500078, India; bhanu@hyderabad.bits-pilani.ac.in (L.B.M.N.); arunam@hyderabad.bits-pilani.ac.in (A.M.)

* Correspondence: p2013531@hyderabad.bits-pilani.ac.in

Received: 11 November 2019; Accepted: 20 January 2020; Published: 2 February 2020

Abstract: Named entity recognition (NER) is a fundamental step for many natural language processing tasks and hence enhancing the performance of NER models is always appreciated. With limited resources being available, NER for South-East Asian languages like Telugu is quite a challenging problem. This paper attempts to improve the NER performance for Telugu using gazetteer-related features, which are automatically generated using Wikipedia pages. We make use of these gazetteer features along with other well-known features like contextual, word-level, and corpus features to build NER models. NER models are developed using three well-known classifiers—conditional random field (CRF), support vector machine (SVM), and margin infused relaxed algorithms (MIRA). The gazetteer features are shown to improve the performance, and theMIRA-based NER model fared better than its counterparts SVM and CRF.

Keywords: information extraction; named entity recognition; Telugu language; gazetteer; support vector machine; conditional random field; margin infused relaxed algorithm

1. Introduction

Named entity recognition (NER) is a sub-task of information extraction (IE) to identify and classify textual elements (words or sequences of words) into a pre-defined set of categories called named entities (NEs) such as the name of a person, organization, or location, expressions of time, quantities, monetary values, percentages, etc. The term *named entity* was first coined at the 6th Message Understanding Conference (MUC-6) [1]. NER plays an essential role in extracting knowledge from the digital information stored in a structured or unstructured form. It acts as a pre-processing tool for many applications, and some of these applications are listed below:

- **Information retrieval** (IR) is the task of retrieving relevant documents from a collection of documents based on an input query. A study by Guo et al. [2] states that 71% of the queries in search engines are NEs and thus IR [3] can benefit from NER by identifying the NEs within the query.
- **Machine translation** (MT) is the task of automatically translating a text from a source to a target language. NEs require a different technique of translation than the rest of the words because, in general, NEs are not vocabulary words. If the errors of an MT system are mainly due to incorrect translation of NEs, then the post-editing step is more expensive to handle. The research study by Babych and Hartley [4] showed that including a pre-processing step by tagging text with NEs achieved higher accuracy in the MT system. The quality of the NER system plays a vital role in machine translation [5,6].

- **Question answering** (QA) systems are tasked with automatically generating answers to questions asked by a human being in natural language. The answers to questions starting with the wh-words (What, When, Which, Where, Who) [7]) are generally NEs. So, incorporating NER in QA systems [8–11] makes the task of finding answers to questions considerably easier.
- **Automatic text summarization** includes topic identification of where the NEs are as an essential indication of a topic in the text [12]. It is shown that integrating named entity recognition significantly improves the performance of resulting summaries [13,14].

The problem of the identification and classification of NEs is quite challenging because of the open nature of vocabulary. There has been a significant amount of work on NER in English, wherein the earlier work on NER is based on rule-based and dictionary-based approaches.

Rule-based NER relies on hand-crafted rules for identifying and classifying NEs. These rules can be structural, contextual, or lexical patterns [15]. For example, the following list shows two rules for recognizing organization and person names:

- $\langle \textit{proper noun} \rangle + \langle \textit{organization designator} \rangle \longrightarrow \langle \textit{organization name} \rangle$
- $\langle \textit{capitalized last name} \rangle , \langle \textit{capitalized first name} \rangle \longrightarrow \langle \textit{person name} \rangle$

The first rule detects organization names that consist of one or more proper nouns followed by an organization designator such as “Corporation” or “Company”. The second rule recognizes person names written in the order of family name, comma, and given the name. The first limitation of the rule-based approach is in the design of generic rules with high precision by the domain expert/linguist. This process takes a significant amount of time and often needs many iterations to improve the performance. Secondly, the rules obtained for a given domain may not be applliccable to other areas for some languages. For example, NEs for the health domain may not be suitable for finance.

Dictionary-based NER uses dictionaries of target entity types (e.g., dictionaries of the names of people, companies, locations, etc.) and identifies the occurrences of the dictionary entries (e.g., Bill Gates, Facebook, Madison Square, etc.) in text [16]. This approach looks very straightforward at first glance but has difficulties due to the ambiguity of natural language. Firstly, the entities can be referred to by different names. For example, Thomas Alva Edison can also be written as Thomas Edison or Edison. It is not practically possible to create a comprehensive dictionary that enumerates all of these variations. Secondly, the same name might represent different entities like a person or location. For example, “Washington” is the name of the first president of the U.S. as well as the name of a state in the U.S. [17]. Since NER systems have to deal with these issues, machine learning approaches have been adopted for NER.

The state-of-the-art of NER systems are machine learning techniques, which can automatically learn to identify and classify NEs based on the data. Supervised learning techniques like hidden Markov model (HMM) [18], maximum entropy model (ME) [19], decision tree [20], conditional random fields [21], neural networks [22], naïve Bayes [23], and support vector machines [24] has been explored to build NER models. There have been few attempts to solve the problem using semi-supervised [25] and unsupervised learning techniques [26]. NER for the English language has been widely researched. However, for South-East Asian languages (especially Telugu) there has not been much progress. Though we may get some insights from the learning models developed for NER in English or other languages, the language-dependent features make it difficult to use similar models for the Telugu language.

Telugu (తెలుగు) is a Dravidian language mostly spoken in the states of Andhra Pradesh, Telangana, and other neighboring states of Southern India. Telugu [27] ranks fourth in terms of the number of people speaking it as a first language in India. The main challenges for Telugu NER are listed below:

1. Telugu is a highly inflectional and agglutinating language: The way lexical forms get generated in Telugu are different from English. In Telugu, words are formed by inflectional suffixes added

to roots or stems. For example: in the word హైదరాబాదులో (*haidarAbAdlo* (transliteration in English)) (in Hyderabad) = హైదరాబాద్ (*haidarAbAd*) + లో (*lo*) (root word + post-position).

2. The absence of capitalization: In English, named entities start with a capital letter and this capitalization plays an important role in identifying and classifying NEs, whereas there is no concept of capitalization in Telugu. For example: పూజ (*puja*) could be the name of a person or the common meaning "worship". In English, we write "Puja" when it is name of a person and "puja" when it refers to the common noun. In Telugu, we write పూజ (*puja*) in both cases. Thus, capitalization is an important feature to distinguish proper nouns from common nouns.
3. Resource-poor language: For the Telugu language, resources like annotated corpora, name dictionaries (gazetteers), morphological analyzers, part-of-speech (POS) taggers, etc. are not adequately available.
4. Relatively free order: The primary word order of Telugu is SOV (subject–object–verb), but the word order of subject and object is largely free. For example, in the sentence: "Ramu sent necklace to sita" can be written as రాము సీతకు హారాన్ని పంపాడు (*rAmu sItaku hArAnni oampADu*) or రాము హారాన్ని సీతాకు పంపాడు (*rAmu hArAnni sItaku pampADu*) in Telugu. Internal changes or position swaps among words in sentences or phrases will not affect the meaning of the sentence.

NER for Telugu has been receiving increasing attention, but there are only a few articles in the recent past. Most of the previous works on NER for Telugu [28–31] build NER models using language-independent features like contextual information, prefix/suffix, orthogonal and POS of current words. The language-dependent features help in improving the performance of the NER task [32] and gazetteers (entity dictionaries) or entity clue lists are part of the language-dependent features. In one of the previous works on Telugu NER [33] the model is built using both language-independent and language-dependent features, but the language-dependent-feature gazetteers are generated manually. However, building and maintaining high-quality gazetteers by hand is time-consuming. Many methods have been proposed for the automatic generation of gazetteers [34]. However, these methods require patterns or statistical methods to extract high-quality gazetteers. The exponential growth in information content, especially in Wikipedia, has made it increasingly popular for solving a wide range of NLP problems across different domains. Wikipedia has 69,450 (https://meta.wikimedia.org/wiki/List_of_Wikipedias) articles in the Telugu language as of July 2018. Each article in Wikipedia is identified by a unique name known as an "entity name". These articles have many useful structures for knowledge extraction such as headings, lists, internal links, categories, and tables. In this work, we used category labels for the dynamic creation of gazetteer features. The process is explained in Section 3.3.3.

The major contributions in this work are listed below:

1. Morphological pre-processing is proposed to handle the inflectional and agglutinating issues of the language.
2. We propose to use language-dependent features like clue words (surname, prefix/suffix, location, organization, and designation) to build an NER model.
3. We present a methodology for the dynamic generation of gazetteers using Wikipedia categories.
4. We extract the proposed features for the FIRE data set and make it publicly available to facilitate future research.
5. We perform a comparative study of NER models built using three well-known machine learning algorithms—support vector machine (SVM), conditional random field (CRF), and margin infused relaxed algorithm (MIRA).
6. We study the impact of gazetteer-related features on NER models.

The rest of this article is organized as follows: The related work on NER in Indian languages is discussed in Section 2. Section 3 explains the NER corpus, tag-set with potential features, and briefly explains the three different classifiers used to build the models. The experimental results are discussed in Section 4 followed by the conclusion of the article in Section 5.

2. Related Work on NER

In this section, we first discuss NER-related studies in the Telugu language, followed by some studies of other Indian languages—Hindi, Bengali, and Tamil.

Srikanth and Murthy [33] were some of the first authors to explore NER in Telugu. They built a two-stage classifier which they tested using the LERC-UoH (Language Engineering Research Centre at University of Hyderabad) Telugu corpus. In the early stage, they built a CRF-based binary classifier for noun identification, which was trained on manually tagged data of 13,425 words and tested on 6223 words. Then, they developed a rule-based NER system for Telugu, where their primary focus was on identifying the name of person, location, and organization. A manually verified NE-tagged corpus of 72,157 words was used to develop this rule-based tagger through boot-strapping. Then, they developed a CRF-based NER system for Telugu using features such as prefix/suffix, orthographic information, and gazetteers, which were manually generated, and reported an F1-score of 88.5%. In our work we present a methodology for the dynamic generation of gazetteers using Wikipedia categories.

Praneeth et al. [28] proposed a CRF-based NER model for Telugu using contextual word of length three, prefix/suffix of the current word, POS, and chunk information. They conducted experiments on data released as a part of the NER for South and South-East Asian Languages (NERSSEAL) (http://ltrc.iiit.ac.in/ner-ssea-08/) competition with 12 classes. The best-performing model gave an F1-Score of 44.91%.

Ekbal et al. [31] proposed a multiobjective optimization (MOO)-based ensemble classifier using a three-base machine learning algorithm (maximum entropy (ME), CRF, and SVM). The ensemble was used to build NER models for Hindi, Telugu, and Bengali languages. The features used to construct the Bengali NER were contextual words, prefix/suffix, length of the word, the position of the word in the sentence, POS information, digital information, and manually generated gazetteer features. They reported an F1-Score of 94.5%. To build an NER model for Hindi and Telugu, they used the contextual words, prefix/suffix, length of the word, the position of the word in the sentence, and POS information, and reported F1-Scores of 92.80% and 89.85% for Hindi and Telugu, respectively.

Sriparna and Asif [30] extended the above work by building an ensemble classifier using base classifiers ME, Naïve Bayes, CRF, Memory-Based Learner, Decision Tree (DT), SVM, and hidden Markov model (HMM) without using any domain knowledge or language-specific resources. The proposed technique was evaluated for three languages—Bengali, Hindi, and Telugu. Results using a MOO-based method yielded the overall F1-Scores of 94.74% for Bengali, 94.66% for Hindi, and 88.55% for Telugu.

Arjun Das and Utpal Garain [29] proposed CRF-based NER systems for the Indian language on the data set provided as a part of the ICON 2013 conference. In this task, the NER model for the Telugu language was built using language-independent features like contextual words, word prefix and suffix, POS and chunk information, and first and last words of the sentence. The model obtained an F1-Score of 69%.

SaiKiranmai et al. [35] built a Telugu NER model using three classification learning algorithms (i.e., CRF, SVM, and ME) on the data set provided as a part of the NER for South and South-East Asian Languages (NERSSEAL) (http://ltrc.iiit.ac.in/ner-ssea-08/) competition. The features used to build the model were contextual information, POS tags, morphological information, word length, orthogonal information, and sentence information. The results show that the SVM achieved the best F1-Score of 54.78%.

SaiKiranmai et al. [36] developed an NER model which classifies textual content from on-line Telugu newspapers using a well-known generative model. They used generic features like contextual words and their POS tags to build the learning model. By understanding the syntax and grammar of the Telugu language, they introduced some language-dependent features like post-position features, clue word features, and gazetteer features to improve the performance of the model. The model achieved an overall average F1-Score of 88.87% for person, 87.32% for location, and 72.69% for organization identification.

SaiKiranmai et al. [37] attempted to cluster NEs based on semantic similarity. They used vector space models to build a word-context matrix. The row vector was constructed with and without considering the different occurrences of NEs in a corpus. Experimental results show that the row vector considering different occurrences of NEs enhanced the clustering results.

In the Hindi language, Li and McCallum [38] built a CRF-based NER model by making use of 340k words with three NE tags, namely person, location, and organization, and reported an F1-score of 71.5%. Saha et al. [39] developed a Hindi NER model using maximum entropy (ME). They developed the model using language-specific and context pattern features, obtaining an F1-score of 81.52%. Saha et al. [40] proposed a novel kernel function for SVM to build an NER model for Hindi and bio-medical data. The NER model achieved an F1-score of 84.62% for Hindi.

In the Bengali language, Ekbal and Sivaji [41] developed an NER model using SVM. The corpus consisted of 150k words annotated with sixteen NE tags. The features used to build the model were context word, word prefix/suffix, POS information, and gazetteers, and it achieved an average F1-score of 91.8%. Ekbal et al. [42] developed an NER model for Bengali and Hindi using SVM. These models use different contextual information of words in predicting four NE classes, such as a person, location, organization, and miscellaneous. The annotated corpora consist of 122,467 tokens for Bengali and 502,974 tokens for Hindi. This model reported an F1-score of 84.15% for Bengali and 77.17% for Hindi. Ekbal et al. [43] developed an NER model using CRF for Bengali and Hindi using contextual features with an F1-score of 83.89% for Bengali and 80.93% for Hindi. Banerjee et al. [44] developed an NER model for Bengali using the margin infused relaxed algorithm. They used IJCNLP-08 NERSSEAL data, which are annotated with twelve NE tags, and obtained an F1-Score of 89.69%.

Vijayakrishna and Sobha [45] developed a Tamil Named Entity Recognizer for the tourism domain using CRF. It handles nested NEs with a tag-set consisting of 106 tags, and reported an overall F1-Score of 80.44%. Abinaya et al. [46] present a NER model for Tamil using the random kitchen sink (RKS) algorithm, which is a statistical and supervised approach. They also implemented the NER model using SVM and CRF and reported overall F1-Scores of 86.61% for RKS, 81.62% for SVM, and 87.21% for CRF.

3. Proposed Methodology for Telugu NER

NER in Telugu is comparatively challenging as it is highly inflectional and agglutinating in nature. Telugu is morphologically rich language [47]. The significant portion of grammar is managed by morphology in Telugu. Each inflected word starts with a root and has many suffixes. The word suffix used here refers to inflections, post-positions, and markers which indicate tense, number, person and gender, negatives, and imperatives. In English, phrases generally include several words, and in most cases, such phrases are mapped to a single word in Telugu. For example, గెలవలేదనుకొన్నావా (*vacciveLLADu*) (do you think he will not win?) and రాజమండ్రివైపు (*rAjamaMDrovaipu*) (towards rajahmundary) are single words in Telugu, which makes the NER task complex.

The application of stochastic models to the NER problem requires a large annotated corpus to achieve a reasonable performance. Stochastic models have been applied to English and other languages due to the availability of sufficiently large annotated corpora. The problem is difficult for Telugu due to the absence of such annotated corpora. HMMs [48] do not work well when small amounts of annotated corpus are used to estimate the model parameters, and the incorporation of diverse features is difficult. In contrast, CRF, SVM, and MIRA learning algorithms can efficiently deal with the diverse and overlapping features of the Telugu language. We implemented these learning algorithms to identify NEs and classify them into predefined NE classes—*Name*, *Location*, *Organization*, and *Miscellaneous*.

In this section, we describe the corpus and tag-set of NEs with potential features and classifiers used to build the NER models.

3.1. Corpus and Named Entity Tag-Set

The different corpora that have been used so far in literature for NER in Telugu are listed below:

- IJCNLP-Workshop on NER for South and South-East Asian Languages-2008 (http://ltrc.iiit.ac.in/ner-ssea-08/index.cgi?topic=5): This data set consists of 64,026 tokens. The tag-set for the task has 12 tags. The reason they opted for these tags was that they needed a slightly finer tag-set for machine translation (MT) and certain domains like health and tourism.
- ICON-NLP Tools Contest on Named Entity Recognition in Indian languages, 2013 (http://ltrc.iiit.ac.in/icon/2013/nlptools/: The data set has four NE classes, and it is not publicly available.

In this work, we used the bench-marked data set (http://fire.irsi.res.in/fire/2018/home) provided by the Forum of Information Retrieval and Evaluation (FIRE-2018). The main advantage is that the corpus is large enough as compared to other available data sets. The data consists of 767,603 tokens, out of which 200,059 are NEs. The size of the data set is given in Table 1.

Table 1. Size of the data set.

Dataset	No. of Tokens	No. of Entities
Train	537,510	139,999
Test	230,093	60,060
Total	767,603	200,059

The data set is annotated with nine named entity tags. A tag conversion routine was implemented on the corpus to scale down the initial nine-member tag-set to the intended four-member tag-set—namely, name, location, organization, and miscellaneous as shown in Table 2.

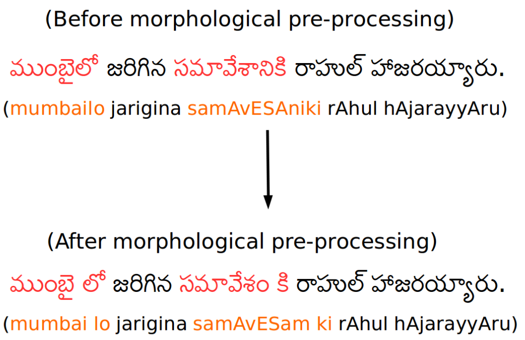
Table 2. Named entity tag-set.

Named Entity Tag	Meaning	Example
person	person name	నాగార్జున (Nagarjuna)
location	location name	హైదరాబాద్ (Hyderabad)
organization	organization name	గూగుల్ (Google)
miscellaneous	number, date, event, things, year, and occupation	1997

3.2. Morphological Pre-Processing

Telugu is a highly inflectional and agglutinating language, and hence it makes all sense to perform morphological pre-processing. Morphology is the study of word formation—how words are formed from smaller morphemes. A morpheme is the smallest part of a word that has grammatical information or meaning. For example, the word హైదరాబాదులో (*haidarAbAdlo*) in Telugu means "in Hyderabad" in English. The morphemes in this word are హైదరాబాద్(*haidarAbAd*) and లో (*lo*). After the morphological pre-processing, the word హైదరాబాదులో (*haidarAbAdlo*) will be split into two words హైదరాబాద్(*haidarAbAd*) and లో (*lo*). We propose this kind of morphological pre-processing to enrich the features of the NER model.

The morphological pre-processing was performed using the TnT (Trigrams'n'Tags) tool [49]. For the following example in Figure 1, with the morphological pre-processing the words ముంబైలో (*mumbailo*) will be split into ముంబై (*mumbai*) and లో (*lo*) similarly, సమావేశానికి (*samAvESAniki*) will be సమావేశం (*samAvESAm*) and కి (*ki*).



(Before morphological pre-processing)

ముంబైలో జరిగిన సమావేశానికి రాహుల్ హాజరయ్యారు.

(mumbailo jarigina samAvESAniki rAhul hAjarayyAru)

(After morphological pre-processing)

ముంబై లో జరిగిన సమావేశం కి రాహుల్ హాజరయ్యారు.

(mumbai lo jarigina samAvESam ki rAhul hAjarayyAru)

Figure 1. Morphological pre-processing.

3.3. *Features*

In this section, we present the features used for the recognition and classification of Telugu NEs. The extraction of features from a text corpus is an essential step in natural language processing (NLP) to apply machine learning (ML) techniques. We organized these features into the following different types: contextual, word-level, gazetteer, and corpus features.

3.3.1. Contextual Features

The neighboring words of a given word carry effective information in classifying whether that word is an NE or not. Hence, we considered words in a sliding window of size k as the contextual features. For example: Given the sentence నాగార్జునసాగర్ జలాశయానికి వరద పూర్తిస్థాయిలో తగ్గుముఖం పట్టింది (nAgArjunasAgar jalASayAniki varada pUrtisthAyilO taggumukham paTTimdi), for the current word వరద (varada) the contextual features for a sliding window of size of $k = 3$ are {జలాశయానికి (jalASayAniki), పూర్తిస్థాయిలో (pUrtisthAyilO)}. The the contextual features for the same word for a sliding window of size $k = 5$ are {నాగార్జునసాగర్ (nAgArjunasAgar), జలాశయానికి (jalASayAniki), పూర్తిస్థాయిలో (pUrtisthAyilO), తగ్గుముఖం (taggumukham)}. The optimal size (k) of the sliding window is decided by performing a sensitivity analysis.

The challenges of Telugu NER are detailed in Section 1. The contextual features in building NER models tend to address the following challenges:

- **Absence of capitalization:** Capitalization is not a distinguishing feature of Telugu script, which makes it difficult to differentiate between common nouns and proper nouns.
  For example: పూజ (*puja*) can be the name of a person or a common noun meaning "worship". The ambiguity between common and proper nouns is resolved using the contextual information of a named entity.
- **Relatively free order:** Internal changes or position swaps among words in sentences or phrases will not affect the meaning of the sentence. This is resolved using the contextual information of a word.
  For example:
  - రాము సీతకు హారాన్ని పంపాడు (*rAmu sItaku hArAnni oampADu*), for the current word సీతకు (sItaku) the contextual features for a sliding window of size of $k = 3$ are {రాము (rAmu), హారాన్ని (hArAnni)}.
  - రాము హారాన్ని సీతాకు పంపాడు (*rAmu hArAnni sItaku pampADu*), for the current word సీతకు (sItaku) the contextual features for a sliding window of size of $k = 3$ are {హారాన్ని (hArAnni), పంపాడు (pampADu)}.

3.3.2. Word-Level Features

Word-level features are related to the individual orthographic nature and structure of each word. They specifically describe word length, the position of a word, whether the word contains a number, and the POS tag of a word. Kumar et al. [50] found that short words are most probably not NEs and predefined the threshold to be less than or equal to three. So, we considered word length as a binary feature if the current word length ≥ 3. In a sentence, the position of a word acts as a good indicator for named entity identification, as NEs tend to appear in the first position of the sentence. In Telugu, verbs typically appear in the last position of the sentence, as it follows a subject–object–verb structure. So, we considered two binary features FirstWord and LastWord.

Previous works in Telugu NER used POS features as a binary feature (i.e., whether a word is a noun or not a noun). The study by SaiKiranmai et al. [51] suggests that other part-of-speech tags like postposition, quantifiers, demonstratives, cardinal/ordinal, NST (noun denoting spatial and temporal expression), and quotative are helpful in identifying whether a given word is a named entity or not. So, in our work we used the TnT [49] POS tagger, which classifies a Telugu word into one of 21 POS tags, and we considered the POS tag of the target word and surrounding words as features for NER.

The Named Entity of previous word(s) was also considered as a dynamic feature in the experiment.

3.3.3. Gazetteer Features

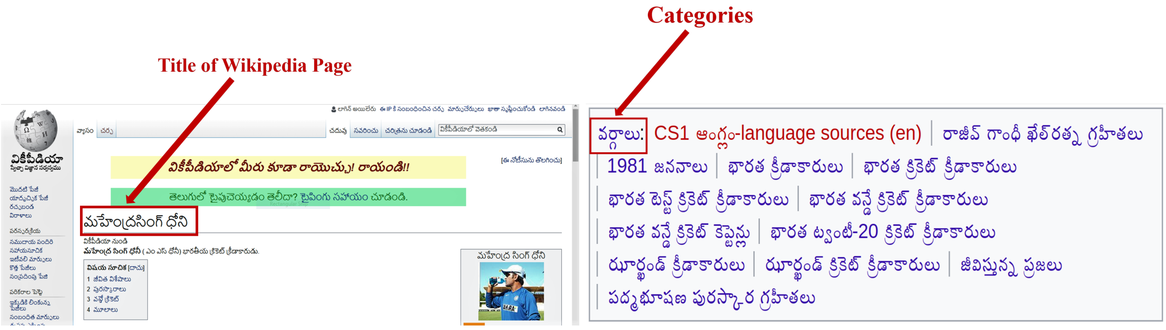
Gazetteers or entity dictionaries play an essential role in improving the performance of the NER task. However, building and maintaining high-quality gazetteers by hand is time-consuming. Many methods have been proposed for the automatic generation of gazetteers from a vast number of text documents [34]. However, these methods require patterns or statistical methods to extract high-quality gazetteers.

The exponential growth in information content, especially in Wikipedia, has made it increasingly popular for solving a wide range of NLP problems across different domains. Wikipedia had 69,450 (https://meta.wikimedia.org/wiki/List_of_Wikipedias) articles in the Telugu language as on July 2018. Each article in Wikipedia is identified by a unique name known as "entity names". These articles have many useful structures for knowledge extraction, such as headings, lists, internal links, categories, and tables. Further, new articles are added to Wikipedia every day. Hence, many recent studies have made use of Wikipedia as a knowledge source to generate gazetteers [52–54].

We explain the procedure of the gazetteer generation of person, location, and organization names by making use of Wikipedia articles in Section 3.3.4 and the generation of clue lists in Section 3.3.5.

3.3.4. Gazetteer Creation Using Wikipedia

Wikipedia maintains a list of categories for each of its title pages. The example Wikipedia page title "Mahendra Singh" and its categories in Telugu and English are as shown in Figures 2 and 3.



వికీపీడియాలో మీరు కూడా రాయొచ్చు! రాయండి!!

తెలుగులో వ్రాయడం తెలీదా? టైపింగు సహాయం చూడండి.

మహేంద్రసింగ్ ధోని

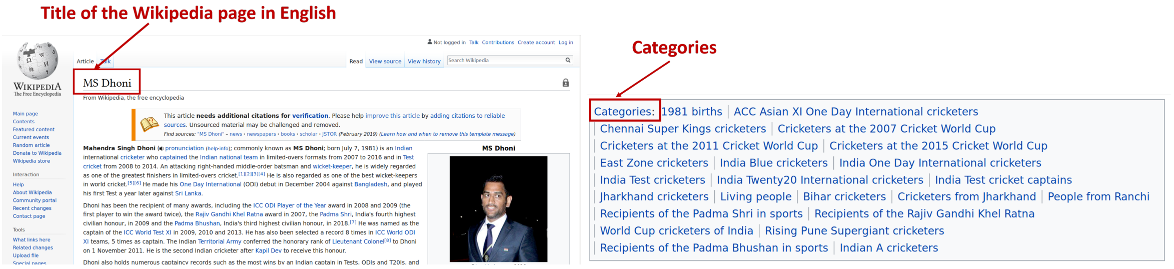
వికీపీడియా నుండి

మహేంద్ర సింగ్ ధోనీ (ఎం ఎస్ ధోనీ) భారతీయ క్రికెట్ క్రీడాకారుడు.

విషయ సూచిక [దాచు]
1 జీవిత విశేషాలు
2 పురస్కారాలు
3 వన్డే క్రికెట్
4 మూలాలు

వర్గాలు: CS1 ఆంగ్లం-language sources (en) | రాజీవ్ గాంధీ ఖేల్ రత్న గ్రహీతలు | 1981 జననాలు | భారత క్రీడాకారులు | భారత క్రికెట్ క్రీడాకారులు | భారత టెస్ట్ క్రికెట్ క్రీడాకారులు | భారత వన్డే క్రికెట్ క్రీడాకారులు | భారత వన్డే క్రికెట్ కెప్టెన్లు | భారత ట్వంటీ-20 క్రికెట్ క్రీడాకారులు | ఝార్ఖండ్ క్రీడాకారులు | ఝార్ఖండ్ క్రికెట్ క్రీడాకారులు | జీవిస్తున్న ప్రజలు | పద్మభూషణ పురస్కార గ్రహీతలు

Figure 2. Title and categories of a Wikipedia page in Telugu.



MS Dhoni

From Wikipedia, the free encyclopedia

This article **needs additional citations for verification**. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed.
Find sources: "MS Dhoni" – news · newspapers · books · scholar · JSTOR *(February 2019) (Learn how and when to remove this template message)*

Mahendra Singh Dhoni (pronunciation (help·info); commonly known as **MS Dhoni**; born July 7, 1981) is an Indian international cricketer who captained the Indian national team in limited-overs formats from 2007 to 2016 and in Test cricket from 2008 to 2014. An attacking right-handed middle-order batsman and wicket-keeper, he is widely regarded as one of the greatest finishers in limited-overs cricket.[1][2][3][4] He is also regarded as one of the best wicket-keepers in world cricket.[5][6] He made his One Day International (ODI) debut in December 2004 against Bangladesh, and played his first Test a year later against Sri Lanka.

Dhoni has been the recipient of many awards, including the ICC ODI Player of the Year award in 2008 and 2009 (the first player to win the award twice), the Rajiv Gandhi Khel Ratna award in 2007, the Padma Shri, India's fourth highest civilian honour, in 2009 and the Padma Bhushan, India's third highest civilian honour, in 2018.[7] He was named as the captain of the ICC World Test XI in 2009, 2010 and 2013. He has also been selected a record 8 times in ICC World ODI XI teams, 5 times as captain. The Indian Territorial Army conferred the honorary rank of Lieutenant Colonel[8] to Dhoni on 1 November 2011. He is the second Indian cricketer after Kapil Dev to receive this honour.

Dhoni also holds numerous captaincy records such as the most wins by an Indian captain in Tests, ODIs and T20Is, and

MS Dhoni

Categories: 1981 births | ACC Asian XI One Day International cricketers | Chennai Super Kings cricketers | Cricketers at the 2007 Cricket World Cup | Cricketers at the 2011 Cricket World Cup | Cricketers at the 2015 Cricket World Cup | East Zone cricketers | India Blue cricketers | India One Day International cricketers | India Test cricketers | India Twenty20 International cricketers | India Test cricket captains | Jharkhand cricketers | Living people | Bihar cricketers | Cricketers from Jharkhand | People from Ranchi | Recipients of the Padma Shri in sports | Recipients of the Rajiv Gandhi Khel Ratna | World Cup cricketers of India | Rising Pune Supergiant cricketers | Recipients of the Padma Bhushan in sports | Indian A cricketers

Figure 3. Title and categories of a Wikipedia page in English.

Zhang et al. [53] made use of these category labels for gazetteer creation for NER. For example, Wikipedia categories such as "Educational institutions established in 1926" and "Companies listed on the Bombay Stock Exchange" refer to organizations; "Living people" and "Player" refer to people; and "States and territories", "City-states" refer to locations.

In our work, NER experiments were conducted on a resource-poor language (Telugu). We devise a procedure for the dynamic creation of gazetteers using Wikipedia categories.

We manually collected most frequent NEs. These frequent NEs are seed list (SE) of each class C = {person, location, organization} and lists of 1049, 731, and 254 entities for persons, locations, and organizations, respectively. We collected category labels (CLs) for all the entities in the seed list. For a person-type named entity the category list may contain "actor", "engineer", and "famous" and for a location-type named entity the category may contain "city", "street", and "famous". Some of the category labels might be there in both category lists of two distinct NEs (e.g., person and location). In the example above the category label "famous" is in both lists. The next step in our algorithm is to remove the ambiguous category labels that are present in more than one list, and the result is a unique category list (UCL) for each class C. The procedure for extracting unique category labels for each NE class is shown in Algorithm 1.

Algorithm 1 Extracting unique category labels for each NE class.

```
Input : $SE_c$ – Seed lists of entities of class $C$ = {person, location, organization}
Output : $UCL_c$ – List of unique category labels of $C$

1.  for each $C$ in {person, location, organization}
    1.1.  for each entity $e$ in $SE_C$
          1.1.1.  retrieve corresponding Wikipedia article for $e$
          1.1.2.  extract the category labels of $e$ and add to $CL_C$
2.  for each C in {person, location, organization}
    2.1.  for each category label $CL_C$
          2.1.1.  if the category is not there in other category lists
                  2.1.1.1.  then add category to $UCL_C$ list.
```

We extracted the category list for each of the Wikipedia titles (WTs) from the Telugu Wikipedia dump (https://dumps.wikimedia.org/tewiki/) of 69,450 articles. We describe the procedure for the generation of gazetteer lists for each class in Algorithm 2. An example is explained below:

Consider the Wikipedia page of the famous Indian cricket player "Mahendra Singh Dhoni" (మహేంద్రసింగ్ ధోని) as shown in Figure 2 (https://te.wikipedia.org/wiki/మహేంద్రసింగ్ _ధోని) and its category labels (వర్గాలు) such as "జీవిస్తున్న ప్రజలు" (living people), "1981జననాలు" (births). Our algorithm searches the category labels of "Mahendra Singh Dhoni" (మహేంద్రసింగ్ ధోని) in the unique category list (UCL) and finds that maximum number of category labels correspond to the class *person*. Consequently, our algorithm classifies "Mahendra Singh Dhoni" (మహేంద్రసింగ్ ధోని) as a *person*.

Algorithm 2 Generation of Gazetteers from category labels

```
Input :List of Wikipedia titles $WT$
Output : $G_C$ – List of Gazetteer $G$ of class $C$

1.   for each $t$ in $WT$
     1.1.   Retrieve the category list ($CL$) of $t$
     1.2.   $C' = \underset{C \in \{person, location, organization\}}{\text{argmax}} (CL \cap UCL_C)$
     1.3.   $t \in G_{C'}$
```

After expansion, our list contains 7593 person names, 4791 location names, and 1254 organization names. Examples of NEs collected for each class are shown in Table 3.

Table 3. Example of named entity instances extracted from Wikipedia.

NE Type	Wiki-Extracted NE
Person	అబ్దుల్(Abdul), చంద్రశేఖర్(Chandrasekhar)
Location	స్విట్జర్లాండ్(Switzerland), హంగరీ(Hungary)
Organization	ఐన్స్టీన్(Infosys), ఎల్ఐసి(LIC)

3.3.5. Gazetteer of Entity Clues

Clue words give some information about whether the current word is a named entity or not. The following are the lists of clue words that have been proposed.

1. **Surname gazetteer:** Surnames occur at the start of person names. We generated a gazetteer of surnames manually by making use of the person gazetteer list obtained from Algorithm 2. For example, in అర్జుల రామచంద్ర రెడ్డి (arjula rAmacmdra reDDi), అర్జుల (arjula) is the surname. If the current word (w_i) is present in the surname gazetteer, then the **Surname** feature is set to 1.
2. **Person suffix gazetteer:** The person suffix occurs at the end of a person's name. We generated a gazetteer of person suffixes manually by making use of the person gazetteer list obtained from Algorithm 2. For example, in అర్జుల రామచంద్ర రెడ్డి (arjula rAmacmdra reDDi), రెడ్డి (reDDi) is the person suffix. If the current word (w_i) is present in the person suffix gazetteer, then the **PerSuffix** feature is set to 1 for the current (w_i) and previous two words (w_{i-1}, w_{i-2}).
3. **Designation gazetteer:** Designation words represent the formal and official status of a person. For example, రాష్ట్రపతి (rAshTapatri), ప్రధానమంత్రి(pradhAnama mtri). If the current word (w_i) is present in the designation gazetteer, then the **Desig** feature is set to 1 for the next word (w_{i+1}).
4. **Person prefix gazetteer:** Person prefixes help in identifying person names (e.g., శ్రీ (SrI), శ్రీమతి (SrImati)). If the current word (w_i) is present in a person prefix gazetteer, then the **PerPrefix** feature is set to 1 for the current (w_i) and next two words (w_{i+1}, w_{i+2}).

5. **Month gazetteer:** The month gazetteer consists of the names of months of both English and Telugu calendars. There are 24 entries in this list. If the current (w_i) word is present in the month gazetteer, then the **Month** feature is set to 1.
6. **Location clue gazetteer:** The location clue gazetteer consists of the words that give clues about location names—for example, clue words like: -pur, -puram, -gunTa, -nagar, -paTnam (తిరువంతపురం (tiruvamta**puram**), కాన్పూర్ (Kan**pur**), రేణిగుంట (rENu**gunTa**), శ్రీనగర్ (SrI**nagar**), మచిలీపట్నం (macilI**paTnam**)). If the current (w_i) word contains any of the suffixes listed in the location clue gazetteer, then the **LocClue** feature is set to 1.
7. **Organization clue gazetteer:** Organization names tend to end with one of a few suffixes, such as మండలి (Council), సంస్థ (Company), సంఘం (Community), సమఖ్యా (Federation), or క్లబ్ (Club). These were collected manually. The feature **OrgClue** is set to 1 for the current (w_i) and previous two words (w_{i-1}, w_{i-2}) if the current word (w_i) is present in the organization clue gazetteer.

The challenges of Telugu NER are specified in Section 1 and "Absence of capitalization" issues are handled by making use of gazetteers. Capitalization is not a discriminate feature for Telugu script, which makes it difficult to distinguish between common nouns and proper nouns. For example, పూజ (*puja*) can be the name of a person or the common meaning "worship". The ambiguity between common and proper nouns is resolved using the contextual information of a named entity. In general, a named entity is identified in context with a trigger word, clue word, and prefix/suffix information to the left and right of the NE.

3.3.6. Corpus Features

In any corpus, the NEs are not as frequent as other words, and hence a rare word is more likely to be a named entity. Therefore we considered a Boolean feature "RareWord" to specify whether a word is rare or not. We defined a word to be a rare word if its frequency was greater than or equal to some threshold value. The threshold frequency of words was tuned by considering different possible threshold values (i.e., 5, 10, 15, and 20). The model obtained the best results when we considered the frequency of 10 as an optimal number of rare words.

The description of all the features used to build NER models are shown in Table 4, where w_i represents the current word.

Table 4. Features. POS: part-of-speech.

Feature	Description
Context	$\text{Context}(w_i) = \{w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}\}$
POS Tag	$\text{POS}(w_i) = \{pos_{i-2}, pos_{i-1}, pos_i, pos_{i+1}, pos_{i+2}\}$
Length	$\text{Length}(w_i) = \begin{cases} 1, & \text{if } w_i \geq 3 \\ 0, & \text{otherwise} \end{cases}$
IsDigit	$\text{IsDisgit}(w_i) = \begin{cases} 1, & \text{if } w_i \text{ is a digit} \\ 0, & \text{otherwise} \end{cases}$
IsFirstWord	$\text{IsFirstWord}(w_i) = \begin{cases} 1, & \text{if } w_i \text{ is the first word of a sentence} \\ 0, & \text{otherwise} \end{cases}$
IsLastWord	$\text{IsLastWord}(w_i) = \begin{cases} 1, & \text{if } w_i \text{ is the last word of a sentence} \\ 0, & \text{otherwise} \end{cases}$
NE Tag	$\text{NE}(w_i) = \text{NE tag of } w_{i-1}$
Person-gazetteer	$\text{PerGaz}(w_i) = \begin{cases} 1, & \text{if } w_i \text{ is present in the person gazetteer list} \\ 0, & \text{otherwise} \end{cases}$
Location-gazetteer	$\text{LocGaz}(w_i) = \begin{cases} 1 & \text{if } w_i \text{ is present in the location gazetteer list} \\ 0, & \text{otherwise} \end{cases}$
Organization-gazetteer	$\text{OrgGaz}(w_i) = \begin{cases} 1, & \text{if } w_i \text{ is present in the organization gazetteer list} \\ 0, & \text{otherwise} \end{cases}$
Surname	$\text{SurName}(w_i) = \begin{cases} 1, & \text{if } w_i \text{ is present in the surname list} \\ 0, & \text{otherwise} \end{cases}$
Person Suffix	$\text{PerSuffix}(w_i) = \text{if } w_i \text{ is present in the person suffix list then } \begin{cases} 1, & w_i, w_{i-1}, w_{i-2} \\ 0, & \text{otherwise} \end{cases}$
Designation	$\text{Desig}(w_i) = \text{if } w_i \text{ is present in the designation list then } \begin{cases} 1, & w_{i+1} \\ 0, & \text{otherwise} \end{cases}$
Person Prefix	$\text{PerPrefix}(w_i) = \text{if } w_i \text{ is present in the person prefix list then } \begin{cases} 1, & w_i, w_{i+1}, w_{i+2} \\ 0, & \text{otherwise} \end{cases}$
Month	$\text{Month}(w_i) = \begin{cases} 1, & \text{if } w_i \text{ is present in the month list} \\ 0, & \text{otherwise} \end{cases}$
Location Clue	$\text{LocClue}_i = \begin{cases} 1, & \text{if } w_i \text{ is present in the location clue list} \\ 0, & \text{otherwise} \end{cases}$
Organization Clue	$\text{OrgClue}(w_i) = \text{if } w_i \text{ is present in the organization clue gazetteer then } \begin{cases} 1, & w_i, w_{i-1}, w_{i-2} \\ 0, & \text{otherwise} \end{cases}$
Rare Word	$\text{RareWord}(w_i) = \text{if } w_i \text{ is present in the rare word list then } \begin{cases} 1, & w_i \\ 0, & \text{otherwise} \end{cases}$

The methods applied to handle the challenges in the Telugu language are listed in Table 5.

Table 5. Methods to handle the challenges in the Telugu language for named entity recognition (NER).

Challenges in Telugu NER	Methods
Inflectional and Agglutinating nature	Morphological pre-processing
Absence of capitalization	Contextual features Clue words Prefix/suffix
Relatively free order	Contextual features

We extracted the proposed features for the FIRE data set and have made it publicly available to facilitate future research (https://github.com/gsaikiranmai/NER/).

3.4. Classifiers

In this section we briefly describe three different classifiers and the tools used to build the models.

3.4.1. Support Vector Machine (SVM)

The support vector machine was evaluated with polynomial kernels of different degrees, and we observed that the kernel with a polynomial of degree 2 fared better. We also observed that the pairwise multi-class decision method performed better than the one vs. rest method. We used the *YamCha* (http://chasen.org/~taku/software/yamcha/) toolkit and *TinySVM* (http://chasen.org/~taku/software/TinySVM/) to implement SVM. The results are shown in Section 4.2 for an SVM with a polynomial kernel of degree 2 and pairwise multi-class decision.

3.4.2. Conditional Random Field (CRF)

Conditional random field (CRF) is a probabilistic framework used for labelling and segmenting sequential data. We used the CRF++ (https://taku910.github.io/crfpp/) toolkit, which is an open source tool. We made use of L2 regularization and the regularization parameter C was set to the default value of 1. The number of iterations processed was 100 and the cut-off threshold for the features was set to the default value of 1.

3.4.3. MIRA

The margin infused relaxed algorithm [55] is a machine learning algorithm for multi-class classification problems. It learns a set of parameters (vector or matrix) by processing all training examples one-by-one and updating the parameters for each training sample. The change in parameters was kept as small as possible. MIRA is also called the *passive-aggressive algorithm (PA-I)*, and it is an extension of the online machine learning perceptron.

We used CRF++ (https://taku910.github.io/crfpp/), an open source tool kit which supports single-best MIRA.

4. Experiment and Results

In this section, we briefly illustrate the performance metrics used in our study to evaluate the models. The results obtained on test data using two different feature sets are explained in Section 4.2.

4.1. Evaluation Metrics

The standard evaluation measures like precision (P), recall (R), and F1-score ($F1$) were considered to evaluate our experiments.

$$Precision(P) = \frac{c}{r}$$

$$Recall(R) = \frac{c}{t}$$

$$F1 - Score = \frac{2 * P * R}{P + R}$$

where r is the number of NEs predicted by the system, t is the total number of NEs present in the test set, and c is the number of NEs correctly predicted by the system.

4.2. Experimental Results on the FIRE Competition Data Set

The data consisted of 767,603 tokens out of which 200,059 were NEs, and we trained the model with 70% of the data and tested on the remaining 30%. Ten sets of training and testing data were generated using the annotated corpus. This split was done randomly and sentences were not repeated in the training and testing data. We then used these 10 sets of test data to evaluate our classifier. The total number of NEs in the test set are shown in Table 6.

Table 6. Total number of named entities in the test set.

Named Entity	Number
location	28,856
name	18,141
misc	12,338
org	725
Total	60,060

We built two different models for three classifiers, with

- Contextual, word-level, and corpus features (Model A);
- Contextual, word-level, corpus, and gazetteer features (Model B).

The results provided below are the averages of the macro recall, precision, and F1-score for 10 runs.

4.2.1. Evaluation Based on Contextual, Word-Level, and Corpus Features (Model A)

We built three models using contextual, word-level, and corpus features using CRF, SVM, and MIRA. The evaluation results on the test set for each named entity class are presented in Table 7.

Table 7. Experimental results of each named entity (NE) class in the test set using contextual, word-level, and corpus features. CRF: conditional random field; MIRA: margin infused relaxed algorithm; SVM: support vector machine. P: precision; R: recall; F1: F1-score.

	MIRA			SVM			CRF		
Named Entity	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
name	**85.41**	**72.14**	**78.21**	81.45	69.45	74.97	72.56	71.24	71.90
location	**85.64**	81.23	**83.37**	81.74	78.45	80.06	74.123	**86.54**	79.85
organization	**61.24**	**58.64**	**59.91**	59.18	57.19	58.16	57.57	41.58	48.20
misc	**91.12**	**89.45**	**90.27**	90.11	88.94	89.52	89.56	84.12	86.75

Note: The higher values are in bold.

In terms of the F1-score, MIRA performed better than SVM and CRF, with relative percentage point improvements of 3.29% and 6.31% for "name", 3.31% and 3.52% for "location", 1.75% and 11.71% for "organization", and 0.75% and 3.52% for "misc", respectively.

The overall average precision, recall, and F1-score of different classifiers are shown in Table 8. Results show that the MIRA-based model performed best among all three models, with 80.85% precision, 75.36% recall, and an F1-score of 77.94%.

Table 8. Overall performance of each classifier.

Classifier	Precision	Recall	F1-Score
MIRA	**80.85**	**75.36**	**77.94**
SVM	78.12	73.51	75.68
CRF	73.45	70.84	71.67

Note: The higher values are in bold.

4.2.2. Evaluation Based on Contextual, Word-Level, Corpus, and Gazetteer Features (Model B)

We built three models for CRF, SVM, and MIRA using contextual, word-level, corpus, and gazetteer features. We strengthened the feature set by including gazetteer features to improve the NER performance. We generated gazetteers for name, location, and organization as explained in Section 3.3.3. We also created entity clues such as surname, person suffix and prefix, location clue, organization clue, designation, and month as explained in the same section. The results obtained by classifiers built using CRF, SVM, and MIRA for each class are presented in Table 9. In terms of precision, CRF performed better than SVM and MIRA for "location", "organization", and "misc" . For "name", MIRA performed better.

Table 9. Experimental results of each NE class on the test set using contextual, word-level, corpus, and gazetteer features.

	MIRA			SVM			CRF		
Named Entity	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
name	**93.05**	**83.47**	**87.95**	90.90	81.93	86.18	91.24	75.23	82.47
location	98.60	**96.45**	**97.51**	98.48	96.02	97.24	**98.78**	94.15	96.41
organization	92.67	**74.35**	**85.25**	92.03	69.78	79.35	**94.68**	57.97	71.91
misc	98.42	96.05	97.21	98.56	95.96	**97.24**	**98.55**	92.83	95.65

Note: The higher values are in bold.

In terms of the F1-score, MIRA performed better than SVM and CRF, with relative percentage point improvements of 1.77% and 5.48% for "name", 0.27% and 1.1% for "location", and 5.9% and 13.34% for "organization". For "misc", SVM performed slightly better than MIRA and CRF, with relative percentage point improvements of 0.03%, 1.59% respectively.

The overall average precision, recall, and F1-score of the three different classifiers are shown in Table 10. For precision, MIRA (96.05%) and SVM (95.45%) performed slightly better than CRF (95.87%), with MIRA showing relative percentage point improvements of 0.6% and 0.18%, respectively. In terms of recall, MIRA (89.91%) performed better than SVM (88.54%) and CRF (83.88%) with relative percentage point improvements of 1.37% and 5.03%, respectively.

Table 10. Overall performance of each classifier.

Classifier	Precision	Recall	F1-Score
MIRA	**96.05**	**89.91**	**92.66**
SVM	95.45	88.54	91.63
CRF	95.87	83.88	88.80

Note: The higher values are in bold.

The number of correctly classified NEs identified by the NER model implemented using CRF, SVM, and MIRA and the number of misclassifications for each classifier are listed in Table 11.

Table 11. Number of entities identified by different classifiers for Model A and Model B.

Entity	No. of Tokens in Test Data	No. of Entities Identified					
		MIRA		SVM		CRF	
		Model A	Model B	Model A	Model B	Model A	Model B
name	18,141	13,088	15,138	12,598	14,864	12,924	13,648
location	28,856	23,442	27,842	22,640	27,720	24,985	27,168
organization	725	425	538	414	506	301	421
miscellaneous	12,338	11,038	11,850	10,974	11,840	10,379	11,453
Total NEs	60,060	47,993	55,368	46,626	54,930	48,589	52,690
Misclassifications		12,067	4692	13,434	5130	11,471	7370

It can be seen that MIRA performed better than SVM and CRF with respect to performance measures like Precision, Recall and F1-score. The main reason for MIRA's superior performance can be attributed to two factors:

1. Its ability to handle overlapping features efficiently.

2. MIRA updates the parameters based on a single training instance at a time rather than updating parameters in a batch mode as in SVM.

4.2.3. Improvement of the Performance of NER by including Gazetteer Features

After including gazetteer features, the performance of the NER model increased, irrespective of the classifier. The results in Table 12 depict the percentage point increases in the performance of each NE class after including gazetteer features. The maximum percentage point increase for the NE class "name" was 11.21% by SVM, for "location" it was 17.18% by SVM, for "organization" it was 25.34% by MIRA, and for "miscellaneous"it was 8.9% by CRF. Out of the four NE classes, the organization NE class benefited most from gazetteer features as it is a multi-word entity and each word in an organization has a different POS tag.

Table 12. Increase in F1-score after including gazetteer features for each class.

	MIRA			SVM			CRF		
NE Class	**Model A**	**Model B**	**Difference**	**Model A**	**Model B**	**Difference**	**Model A**	**Model B**	**Difference**
name	78.21	87.95	9.74	74.97	86.18	11.21	71.90	82.47	10.57
location	83.37	97.51	14.14	80.06	97.24	17.18	79.85	96.41	16.56
organization	59.91	85.25	25.34	58.16	79.35	21.19	48.20	71.91	23.71
miscellaneous	90.27	97.21	6.94	89.52	97.24	7.72	86.75	95.65	8.9

The results in Table 13 show that the overall increases in the performance after including gazetteer features were 14.72% for MIRA, 15.95% for SVM, and 17.13% for CRF.

Table 13. Overall increase in F1-score after including gazetteer features.

Classifier	Model A (F1-Score)	Model B (F1-Score)	Percentage Increase in F1-Score
MIRA	77.94	92.66	14.72
SVM	75.68	91.63	15.95
CRF	71.69	88.80	17.13

Hence, we conclude that the gazetteer features improved the performance of our NER model.

4.2.4. Discussion and Error Analysis

An important characteristic of any data set is the variation in the data. The most common measure of variation, or spread, is the standard deviation. The standard deviation is a number that measures how far data values are from their mean. Table 14 shows the minimum, maximum, mean, median, and standard deviation of the three classifiers (MIRA, SVM, and CRF) using Model A and Model B. The median value of MIRA in both Model A and Model B was greater than that of other classifiers, so MIRA performed better than SVM and CRF.

Table 14. Measure of dispersion.

	MIRA		SVM		CRF	
	Model A	**Model B**	**Model A**	**Model B**	**Model A**	**Model B**
Minimum	77.70	92.35	75.12	91.21	71.48	88.01
Maximum	78.92	93.33	76.08	92.08	72.76	89.41
Standard Deviation	0.3552	0.3231	0.2771	0.2509	0.5026	0.440
Mean	78.01	92.88	75.47	91.63	72.10	88.83
Median	77.94	92.94	75.79	91.69	72.25	88.89

A two-tailed t-test was performed using the macro F1-score to check if there was a significant difference between Model A and Model B for MIRA, SVM, and CRF. The corresponding p-values are

3.77×10^{-15}, 1.88×10^{-16}, and 1.83×10^{-13}. Since the *p*-values are much less than 0.05, we conclude that there was a significant difference between Model A and Model B irrespective of the classifiers.

The procedure that we put forth to create dynamic gazetteers generated rich collections of gazetteer lists: 7593 person names, 4791 location names, and 1254 organization names. The corresponding gazetteer features contributed to the improvement of our NER model.

Further, we performed a pairwise *t*-test for MIRA–SVM, MIRA–CRF, and SVM–CRF to check if there was a significant difference between these pairs. The corresponding *p*-values are 5.096×10^{-6}, 5.327×10^{-9}, and 2.14×10^{-10}. As the *p*-values are less than 0.05, we conclude there was a significant difference between all pairs of classifiers.

We ran an error analysis to identify incorrect predictions for each class. The following are examples of false negatives incorrectly predicted by Model A and correctly predicted by Model B.

- **Organization**
  - ఐక్యరాజ్య సమితి (*aikyarAjya Samiti*) was misclassified as *aikyarAjya<other> Samiti<other>* by **Model A**. By including the organization suffix as a clue feature, Model B was able to classify correctly (i.e., *aikyarAjya<organization> Samiti<organization>*).
  - భారతీయ జనతా పార్టీ (*bhAratIya janatA pArtI*) was misclassified as *bharatiya<location> janata<other> pArtI<other>* by **Model A**. The dynamic gazetteers generated using Wikipedia enabled Model B to classify *bhAratIya<organization> janatA<organization> pArtI<organization>* correctly.
- **Name**
  - రేవురి ప్రకాష్ రెడ్డి (*rEvUri prakAsh reDDi*) was misclassified as *rEvUri<other> prakAsh<name> reDDi<other>* by **Model A**. By including person prefix/suffix as a clue feature, Model B was able to classify *rEvUri<name> prakAsh<name> reDDi<name>* correctly.
  - శేషరెడ్డి (*SeshAreDDI*) was misclassified as *SeshAreDDI<other>* by **Model A**. The dynamic gazetteers generated using Wikipedia enabled Model B to classify *SeshAreDDI<name>* correctly.

The following provides an example relevant to morphological pre-processing :

- **Location**
  - భారతదేశంలో (*bhAratadESamlo*) was misclassifed as *bhAratadESamlo<other>* before morphological pre-processing. After morphological pre-processing it was classified as *bhAratadESam<location> lo<other>*.

The following are examples of false positives incorrectly predicted by Model A and correctly predicted by Model B.

- **Others**
  - పార్టీ (*pArtI*):

    In the sentence ఈ ఎన్నికల్లో భారతీయ జనతా పార్టీ విజయం సాధించింది. (*I ennikallO bhAratIya janatA pArtI vijayam sAdhimcimdi*) the word పార్టీ (*pArtI*) is tagged as <*organization*>.

    In the sentence నేను పార్టీ కి వెళ్ళాను. (*nEnu pArTI ki vellEnu*) the word పార్టీ (*pArtI*) is tagged as <*other*> but Model A predicted it as <*organization*> as in the corpus most of the time *pArtI* was preceded by an organization name. Model B predicted it correctly as <*other*> as the organization gazetteer feature for the preceding words was zero, which helped it to classify correctly.

- నరసింహస్వామి (*narasimhasvAmi*):

  In the sentence మైదవోలు నరసింహస్వామి గవర్నర్‌గా ఉన్నారు. (*maidavOlu narasimhasvAmi gavarnrgA unnAru*) the word నరసింహస్వామి (*narasimhasvAmi*) is tagged as <*name*>.

  In the sentence నేను నరసింహస్వామి ఆలయానికి వెళ్ళాను. (*nEnu narasimhasvAmi AlayAniki vellEnu*) the word నరసింహస్వామి (*narasimhasvAmi*) is tagged as <*other*> but Model A predicted it as <*name*> as in the corpus most of the time *narasimhasvAmi* was a person's name. Model B predicted it correctly as <*other*> as in the person gazetteer, the person prefix/suffix features was zero for surrounding words, which helped to classify correctly.

4.3. Experimental Results on the NER for South and South-East Asian Languages (NERSSEAL) Competition Data Set

The Telugu NER data set was released as a part of the NER for South and South-East Asian Languages (NERSSEAL) (http://ltrc.iiit.ac.in/ner-ssea-08/index.cgi?topic=3) competition. The data set consists of 64,026 tokens out of which 10,894 are NEs and it is divided into training and testing sets. Characteristics of the data set are shown in Table 15.

Table 15. NER for South and South-East Asian Languages (NERSSEAL) data set characteristics.

Dataset	No. of Tokens	No. of NEs
Train	46,068	8485
Test	17,958	2409
Total	64,026	10,894

The tag-set as mentioned in the competition was based on AUKBC's ENAMEX (Named Entities tag), TIMEX (Temporal Expressions), and NUMEX (Number Expressions). It has 12 tags (i.e., NEP-Person, NED-Designation, NEO-Organization, NEA-Abbreviation, NEB-Brand, NETP-Title-Person, NETO-Tile-object, NEL-Location, NETI-Time, NEN-Number, NEM-Measure, NETE-Terms). In order to make consistency between FIRE and NERSSEA data sets we combined the tags. NEP, NED, and NETP were grouped to name; NEO and NEB were grouped to organization; NELis was grouped to location; and NEA, NETO, NETI, NETN, NETM, and NETE were grouped to miscellaneous.

We built a model with contextual word-level corpus features using the NERSSEAL (http://ltrc.iiit.ac.in/ner-ssea-08/index.cgi?topic=3) competition data set and refer to this model as Model A. We built the model with contextual, word-level, corpus, and gazetteer features using the NERSSEAL (http://ltrc.iiit.ac.in/ner-ssea-08) competition data set and refer to this model as Model B. Table 16 shows the per-class F1-score values for Model A (without gazetteer features) and Model B (with gazetteer features). The overall performances of each classifier with respect to precision, recall, and F1-score are shown in Table 17.

Table 16. Experimental results of each NE class on the test set for Models A and B in terms of F1-score.

	Model A (F1-Score)			**Model B (F1-Score)**		
NE Class\Classifier	**MIRA**	**SVM**	**CRF**	**MIRA**	**SVM**	**CRF**
name	79.25	75.45	75.96	89.45	85.43	85.32
location	82.13	79.48	77.95	92.13	91.45	91.21
organization	46.12	41.58	36.56	68.12	65.89	63.21
miscellaneous	75.69	73.65	72.99	80.68	79.84	80.12

Table 17. Experimental results of each classifier for Models A and B.

	Model A			**Model B**		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
MIRA	65.54	58.22	61.66	84.09	76.38	80.04
SVM	60.12	54.71	57.283	82.56	74.51	78.32
CRF	57.49	54.19	55.79	81.90	73.70	77.58

Table 18 shows the minimum, maximum, mean, median, and standard deviation of the three classifiers (MIRA, SVM, and CRF) using Model A and Model B. The median values of MIRA in both Model A and Model B were greater than for the other classifiers, and so MIRA performed better than SVM and CRF.

Table 18. Measure of dispersion.

	MIRA		**SVM**		**CRF**	
Minimum	61.19	79.90	56.95	78.09	55.58	77.32
Maximum	62.50	80.16	57.58	78.56	55.91	77.79
Standard Deviation	0.3552	0.0799	0.1747	0.1756	0.1048	0.1796
Mean	61.66	80.05	57.29	78.33	55.79	77.58
Median	61.65	80.06	57.30	78.37	55.83	77.62

A two-tailed t-test was performed using the macro F1-Score to check if there was a significant difference between Model A and Model B for MIRA, SVM, and CRF. The corresponding p-values are 2.523×10^{-17}, 2.056×10^{-17}, and 2.493×10^{-19}. Since the p-value is less than 0.05, we conclude that there was a significant difference between Model A and Model B, irrespective of the classifier.

Further, we performed pairwise t-tests for MIRA–SVM, MIRA–CRF, and SVM–CRF to check if there was a significant difference between these pairs. The reported p-values are 2.224×10^{-10}, 1.158×9^{-15}, and 1.184×10^{-5}. Since the p-values are less than 0.05, we conclude there was a significant difference between the pairs of classifiers.

5. Conclusions and Future Work

In this work, we put forth an approach to generate gazetteers dynamically for three named entities—person, location, and organization—and propose gazetteer-based features for Telugu NER. We also performed morphological pre-processing and used language-dependent features to enhance the performance of the NER models. NER models were built with MIRA, SVM, and CRF classifiers, and we demonstrated that MIRA was comparatively better than the other two classifiers. Our experimental results on two benchmark data sets show that the gazetteer features improved the performance of the NER models. With the proposed gazetteer features, the performance (F1-score) of the NER models built using MIRA, SVM, and CRF were increased by 14.72%, 15.95%, and 17.13%, respectively. There are not many open resources available to further the NER research in Telugu, and hence the two data sets along with language-dependent features have been made publicly available. We want to explore deep learning models using different word embeddings and state-of-the-art algorithms to build NER models in the future.

Author Contributions: Conceptualization, S.G., L.B.M.N. and A.M.; methodology, S.G.; software, S.G.; validation, L.B.M.N. and A.M.; formal analysis, S.G.; writing–original draft preparation, S.G.; writing–review and editing, L.B.M.N. and A.M.; supervision, L.B.M.N. and A.M. All authors have read and agreed to the published version of the manuscript

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Grishman, R. The NYU System for MUC-6 or Where's the Syntax? In Proceedings of the 6th conference on Message understanding. Association for Computational Linguistics, Center for Sprogteknologi, Copenhagen, Denmark, 5–9 August, 1995; pp. 167–175.
2. Guo, J.; Xu, G.; Cheng, X.; Li, H. Named Entity Recognition in Query. In Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in information retrieval, Boston, MA, USA, 19–23 July 2009, pp. 267–274.
3. Benajiba, Y.; Diab, M.; Rosso, P. Arabic Named Entity Recognition: A Feature-Driven Study. *IEEE Trans. Audio Speech Lang. Process.* **2009**, *17*, 926–934. [CrossRef]
4. Babych, B.; Hartley, A. Improving Machine Translation Quality with Automatic Named Entity Recognition. In Proceedings of the 7th International EAMT workshop on MT and other language technology tools, Improving MT through other language technology tools, Resource and tools for building MT at EACL, Budapest, Hungary, 12–17 April 2003; pp. 1–8.
5. Hálek, O.; Rosa, R.; Tamchyna, A.; Bojar, O. Named entities from Wikipedia for machine translation. In Proceedings of the CEUR Workshop Proceedings 584, Horský hotel Kralova studna, Harmanec, Slovakia, 25–29 September 2011; pp. 23–30.
6. Chen, Y.; Zong, C.; Su, K.Y. A joint model to identify and align bilingual named entities. *Comput. Linguist.* **2013**, *39*, 229–266. [CrossRef]
7. Indurkhya, N.; Damerau, F.J. *Handbook of Natural Language Processing*, 2nd ed.; Chapman & Hall/CRC: London, NY, USA, 2010.
8. Srihari, R.; Li, W. *Information Extraction Supported Question Answering*; Technical Report; Cymfony Net Inc.: Williamsville, NY, USA, 1999; pp. 1–13.
9. Toral, A.; Noguera, E.; Llopis, F.; Muñoz, R. Improving Question Answering Using Named Entity Recognition. In *Natural Language Processing and Information Systems*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 181–191.
10. Mollá, D.; Van Zaanen, M.; Smith, D. Named Entity Recognition for Question Answering. In Proceedings of the Australasian language technology workshop, Sydney, Australia, November 2006; pp. 51–58.
11. Álvaro Rodrigo.; Pérez-Iglesias, J.; Peñas, A.; Garrido, G.; Araujo, L. Answering questions about European legislation. *Expert Syst. Appl.* **2013**, *40*, 5811–5816. [CrossRef]
12. Nobata, C.; Sekine, S.; Isahara, H.; Grishman, R. Summarization System Integrated with Named Entity Tagging and IE pattern Discovery. In Proceedings of the European Language Resources Association (ELRA); Las Palmas, Canary Island, Spain, 29–31 May 2002; pp. 1–4.
13. Hassel, M. Exploitation of named entities in automatic text summarization for swedish. In Proceedings of the NODALIDA'03–14th Nordic Conference on Computational Linguistics, Reykjavik, Iceland, 30–31 May 2003; pp. 9–16.
14. Baralis, E.; Cagliero, L.; Jabeen, S.; Fiori, A.; Shah, S. Multi-document summarization based on the Yago ontology. *Expert Syst. Appl.* **2013**, *40*, 6976–6984. [CrossRef]
15. Mikheev, A.; Moens, M.; Grover, C. Named Entity Recognition Without Gazetteers. In Proceedings of the Ninth Conference on European Chapter of the Association for Computational, Association for Computational Linguistics, Bergen, Norway, 8–12 June 1999; pp. 1–8.
16. Gerner, M.; Nenadic, G.; Bergman, C.M. LINNAEUS: A species name identification system for biomedical literature. *Bmc Bioinform.* **2010**, *11*, 1–17. [CrossRef]
17. Cucerzan, S.; Yarowsky, D. Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence. In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, College Park, MD, USA, 21–22 June 1999; pp. 90–99.
18. Bikel, D.M.; Schwartz, R.; Weischedel, R.M. An algorithm that learns what's in a name. *Mach. Learn.* **1999**, *34*, 211–231. [CrossRef]
19. Borthwick, A.; Sterling, J.; Agichtein, E.; Grishman, R. Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In Proceedings of the Sixth Workshop on Very Large Corpora, Montreal, QC, Canada, 15–16 August 1998; pp. 152–160.

20. Sekine, S.; Grishman, R.; Shinnou, H. A decision tree method for finding and classifying names in Japanese texts. In Proceedings of the Sixth Workshop on Very Large Corpora, Montreal, Quebec, Canada, 15–16 August 1998; pp. 171–178.
21. McCallum, A.; Li, W. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003-Volume 4: Association for Computational Linguistics, Edmonton, Canada, 27 May–1 June 2003; pp. 188–191.
22. Kazama, J.; Torisawa, K. A New Perceptron Algorithm for Sequence Labeling with Non-Local Features. In Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Prague, Czech Republic, 28–30 June 2007; pp. 315–324.
23. Mohit, B.; Hwa, R. Syntax-based Semi-Supervised Named Entity Tagging. In Proceedings of the ACL Interactive Poster and Demonstration Sessions. Association for Computational Linguistics, Ann Arbor, MI, USA, 25–30 June 2005; pp. 57–60.
24. Isozaki, H.; Kazawa, H. Efficient support vector classifiers for named entity recognition. In Proceedings of the 19th international conference on Computational linguistics-Volume 1: Association for Computational Linguistics, Taiwan, 26–30 August 2002; pp. 1–7.
25. Collins, M.; Singer, Y. Unsupervised models for named entity classification. In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, College Park, MD, USA, 21–22 June 1999; pp. 100–110.
26. Nadeau, D. Semi-supervised named entity recognition: Learning to recognize 100 entity types with little supervision. Ph.D. Thesis, University of Ottawa, Ottawa, ON, Canada, 2007.
27. Available online: https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers_in_India (accessed on 20 January 2020).
28. Shishtla, P.; Gali, K.; Pingali, P.; Varma, V. Experiments in Telugu NER: A Conditional Random Field Approach. In Proceedings of the Third International Joint Conference on Natural Language Processing, IJCNLP, Hyderabad, India, 7–12 January 2008; pp. 105–110.
29. Das, A.; Garain, U. CRF-Based Named Entity Recognition @ICON 2013. *arXiv* **2014**, arXiv:1409.8008.
30. Saha, S.; Ekbal, A. Combining Multiple Classifiers Using Vote Based Classifier Ensemble Technique for Named Entity Recognition. *Data Knowl. Eng.* **2013**, *85*, 15–39. [CrossRef]
31. Ekbal, A.; Saha, S. A Multiobjective Simulated Annealing Approach for Classifier Ensemble: Named Entity Recognition in Indian Languages As Case Studies. *Expert Syst. Appl.* **2011**, *38*, 14760–14772. [CrossRef]
32. Nadeau, D.; Sekine, S. A survey of named entity recognition and classification. *Lingvisticae Investig.* **2007**, *30*, 3–26.
33. Srikanth, P.; Murthy, K.N. Named Entity Recognition for Telugu. In Proceedings of the Third International Joint Conference on Natural Language Processing, IJCNLP, Hyderabad, India, 7–12 January2008; pp. 41–50.
34. Ciravegna, F.; Chapman, S.; Dingli, A.; Wilks, Y. Learning to harvest information for the semantic web. In *European Semantic Web Symposium*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 312–326.
35. Gorla, S.; Murthy, N.L.B.; Malapati, A. A Comparative Study of Named Entity Recognition for Telugu. In Proceedings of the 9th Annual Meeting of the Forum for Information Retrieval Evaluation, Gandhinagar, India, 6–9 December 2017; pp. 21–24.
36. Gorla, S.; Velivelli, S.; Murthy, N.B.; Malapati, A. Named Entity Recognition for Telugu News Articles using Naïve Bayes Classifier. In Proceedings of the Second International Workshop on Recent Trends in News Information Retrieval co-located with 40th European Conference on Information Retrieval(ECIR), Grenoble, France, 26 March 2018; pp. 33–38.
37. Gorla, S.; Chandrashekhar, A.; Bhanu Murthy, N.L.; Malapati, A. TelNEClus: Telugu Named Entity Clustering Using Semantic Similarity. In Proceedings of the International Conference on Computational Intelligence: Theories, Applications and Future Directions, Indian Institute of Technology Kanpur, India, 6–8 December 2019; Springer: Singapore, 2019; Volume II, pp. 39–52.
38. Li, W.; McCallum, A. Rapid Development of Hindi Named Entity Recognition Using Conditional Random Fields and Feature Induction. *ACM Transactions on Asian Language Information Processing (TALIP)*; Association for Computing Machinery: New York, NY, USA, 2003; pp. 290–294.

39. Saha, S.K.; Sarkar, S.; Mitra, P. A hybrid feature set based maximum entropy Hindi named entity recognition. In Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I, Hyderabad, India, 12 January 2008; pp. 343–349.
40. Saha, S.K.; Narayan, S.; Sarkar, S.; Mitra, P. A composite kernel for named entity recognition. *Pattern Recognit. Lett.* **2010**, *31*, 1591–1597. [CrossRef]
41. Ekbal, A.; Bandyopadhyay, S. Bengali named entity recognition using support vector machine. In Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages, Hyderabad, 12 January 2008; pp. 51–58.
42. Ekbal, A.; Bandyopadhyay, S. Named entity recognition using support vector machine: A language independent approach. *Int. J. Electr. Comput. Syst. Eng.* **2010**, *4*, 155–170.
43. Ekbal, A.; Bandyopadhyay, S. A conditional random field approach for named entity recognition in Bengali and Hindi. *Linguist. Issues Lang. Technol.* **2009**, *2*, 1–44.
44. Banerjee, S.; Naskar, S.K.; Bandyopadhyay, S. Bengali named entity recognition using margin infused relaxed algorithm. In Proceedings of the International Conference on Text, Speech, and Dialogue, Brno, Czech Republic, 8–12 September 2014; pp. 125–132.
45. Vijayakrishna, R.; Sobha, L. Domain Focused Named Entity Recognizer for Tamil Using Conditional Random Fields. In Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages, Hyderabad, India, 12 January 2008; pp. 59–66.
46. Abinaya, N.; Kumar, M.A.; Soman, K. Randomized kernel approach for named entity recognition in Tamil. *Indian J. Sci. Technol.* **2015**, *8*, 1–7. [CrossRef]
47. Krishnamurti, B.; Gwynn, J.P.L. *A Grammar of Modern Telugu*; Oxford University Press: New York, NY, USA, 1985.
48. Robert, C. *Machine Learning, a Probabilistic Perspective*; MIT Press: Cambridge, CA, USA, 2014.
49. Reddy, S.; Sharoff, S. Cross language POS taggers (and other tools) for Indian languages: An experiment with Kannada using Telugu resources. In Proceedings of the Fifth International Workshop On Cross Lingual Information Access, Chiang Mai, Thailand, 8–12 November 2011; pp. 11–19.
50. Kumar, G.B.; Murthy, K.N.; Chaudhuri, B. Statistical Analyses of Telugu Text Corpora. *Int. J. Dravid. Lang.* **2007**, *36*, 1–20.
51. Gorla, S.; Velivelli, S.; Satpathi, D.K.; Murthy, N.L.B.; Malapati, A. Named Entity Recognition Using Part-of-Speech Rules for Telugu. In *Smart Computing Paradigms: New Progresses and Challenges*; Springer: Singapore, 2020; pp. 147–157.
52. Zesch, T.; Gurevych, I.; Mühlhäuser, M. Analyzing and accessing Wikipedia as a lexical semantic resource. In *Data Structures for Linguistic Resources and Applications*; Gunter Narr: Tübingen, Germany, 2007; pp. 197–205.
53. Zhang, Z.; Iria, J. A Novel Approach to Automatic Gazetteer Generation using Wikipedia. In Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources (People's Web), Suntec, Singapore, August 2009; Association for Computational Linguistics: Suntec, Singapore, 7 August 2009; pp. 1–9.
54. Attia, M.; Toral, A.; Tounsi, L.; Monachini, M.; van Genabith, J. An Automatically Built Named Entity Lexicon for Arabic. In Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta, 17–23 May 2010; European Language Resources Association (ELRA): Valletta, Malta, 2010; pp. 3614–3621.
55. Crammer, K.; Singer, Y. Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.* **2003**, *3*, 951–991.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).