Natural Language Processing (2024), 1–17
doi:10.1017/nlp.2024.14

ARTICLE

A bidirectional LSTM-based morphological analyzer for Gujarati

Jatayu Baxi and Brijesh Bhatt

Department of Computer Engineering, Dharmsinh Desai University, Nadiad, India
Corresponding author: Jatayu Baxi; Email: jatayubaxi.ce@ddu.ac.in

(Received 27 November 2022; revised 13 January 2024; accepted 13 February 2024)

Special Issue on '**Natural Language Processing Applications for Low-Resource Languages**'

Abstract

Morphological analysis is a crucial preprocessing stage for building the state-of-the-art natural language processing applications. We propose a bidirectional LSTM (long short-term memory)-based approach to develop the morphological analyzer for the Gujarati language. Our morph analyzer predicts a root word and the morphological features for the given inflected word. We have experimented with two different methods for label representation for predicting morphological features: the monolithic representation method and the individual label representation method. We have also created the gold morphological dataset of 16,234 unique words for the Gujarati language. The dataset contains morpheme splitting and grammatical feature information for each inflected word. Due to the change in the label representation technique in the proposed model, the accuracy of the present baseline system is improved by a large margin. The proposed system performs very well across the POS categories without the knowledge of language-specific suffix rules.

Keywords: Morphology; machine learning; language resources

1. Introduction

In this section, we introduce the problem of morphological analysis first and understand how low-resource scenario such as the Gujarati language can make this problem more challenging. We discuss the motivation behind using deep learning-based methods for Gujarati morph analysis and provide the key highlights of this paper.

A morph analyzer performs two key tasks: analyzing an inflected word and separating root and suffix parts and assigning grammatical features to the inflected word. For the given inflected word, a morph analyzer produces corresponding root word and set of grammatical features associated with an inflected word. Based on the type of morphology, the morph analysis is classified into two types: inflectional morph analysis and derivational morph analysis. Inflectional morph analyzer can handle the words where the inflections affect the grammatical feature (e.g., go → going) and derivational analyzer can handle the words where the POS category of the word changes due to derivational suffix (e.g., drive → driver). For the English language, significant work has been done in the area of morphological analysis due to the availability of a large corpus and wide coverage of the language (Byrd *et al.* 1986; Minnen, Carroll, and Pearce 2001; Tang 2006). English morphological analyzers are publicly available in the popular natural language processing (NLP) libraries such as Natural Language Toolkit (NLTK).[a]

[a] https://aclweb.org/aclwiki/Morphology_software_for_English

The existing research in the field of morphological analysis is focused only on a few resource-rich languages. Many low-resource languages are at the risk of systematic disappearance due to absence of the computational models for the analysis of the language characteristics (Magueresse, Carles, and Heetderks 2020). The lack of documentation of word formation rules, lack of standardized grammar, and unavailability of the resources such as corpus and datasets present unique challenges in developing efficient morphological analysis tools for the low-resource languages. The complex and ambiguous word formation rules and attachment of multiple suffixes in the morphologically rich languages add an extra challenge in developing an efficient morphological analyzer. The standard rule-based systems often fail to capture language-specific properties. Due to the above issues, the need arises to build more sophisticated models for the morphological analysis. Our work in this paper is focused on the Gujarati language. Gujarati is a member of the Indo-Aryan language family. It is derived from Sanskrit—an ancient Indian language. Gujarati grammar is very similar to that of other Indo-Aryan languages. Compared to other Indian languages, less work has been reported in the field of NLP for the Gujarati language due to the unavailability of the standard word formation rules and annotated datasets for the various NLP tasks. Some existing notable work for the Gujarati language are the development of stemmer (Patel, Popat, and Bhattacharyya 2010; Suba, Jiandani, and Bhattacharyya 2011), POS tagger (Patel and Gali 2008), Morph analyzer (Baxi, Patel, and Bhatt 2015; Baxi and Bhatt 2021), and automatic speech recognition (Raval *et al.* 2020). We believe that there is a need for the research in Gujarati language in terms of language resources and the development of linguistic tools. Once the basic linguistic tools are developed and are made available for use, the higher-level applications like question answering, text summarization, and machine translation can be developed more efficiently. Also, the development of basic linguistic tools will lead to a better understanding of the language and its grammatical structure.

Many researchers have pointed out that the traditional rule-based methods often fail to capture the word formation process and do not give promising results for the morphological analysis task. This is our primary motivation behind using a deep learning-based model for this task. In general, the rule-based methods often suffer from the problem of ambiguity in grammatical feature tagging task due to the complex and ambiguous word formation rules. It is difficult to use rule-based methods for the Gujarati language as the documentation containing such rules is not publicly available. Traditional machine learning methods are also not good choice for this task due to the requirement of heavy feature engineering. The main advantage of deep neural models is that they do not require feature engineering. We choose Bi-LSTM model for our task because it takes into account both the past and future contexts while learning. This fact may be useful for the identification of the inflection pattern and morpheme boundary detection task. The results obtained through Bi-LSTM model are compared with vanilla Recurrent Neural Network (RNN) model and also the unsupervised model.

In this article, we focus on our work on the creation of Bi-LSTM-based morphological analyzer for the Gujarati language. Our morphological analyzer mainly performs two tasks: morpheme boundary detection and grammatical feature tagging. For Gujarati, POS tagged dataset is available but morphologically tagged dataset was not available so we have studied the Gujarati language morphology and created a morphological dataset for the evaluation of the proposed system. The dataset is annotated as per the standard format and included in the latest release of the UniMorph dataset (Batsuren *et al.* 2022; Baxi and bhatt 2022). The proposed system improves the accuracy of the baseline system (Baxi and Bhatt 2021) by a large margin.

The major highlights of our research are as follows:

- The study of Gujarati linguistics and associated challenges
- The creation of morphological dataset for Gujarati

https://doi.org/10.1017/nlp.2024.14 Published online by Cambridge University Press

- The proposed change in label representation technique for the grammatical feature prediction task to improve the performance of the baseline system
- The result analysis from the linguistic perspective

The remaining article is organized as follows. In Section 2, we discuss related work from the literature in the field of computational morphology and morph analyzer. In Section 3, we describe the proposed model. We demonstrate the experiment details, result analysis, and observation in Section 4 of this article.

2. Related work

In this section, we survey about various approaches for developing morphological analyzer in chronological order in Subsection 2.1. In Subsection 2.2, we give details about Gujarati grammar and morphology along with unique linguistic challenges. We survey about existing morphology related work for the Gujarati and investigate the limitations of the baseline system in Subsection 2.3.

2.1 Morphological analysis development landscape

Since morphological analysis serves as foundation for almost all higher-level NLP tasks, significant research is done in this field since the early eighties. Majority of the research in this area was focused only on a few resource-rich languages. The first computational model for the morphology was developed by Kimmo Koskenniemi (Koskenniemi 1984) which had a support for word form recognition and generation. It is considered as the pioneer work in the analysis of morphologically complex languages. In Beesley and Karttunen (1992), the authors developed current C version of two-level compiler, called TWOLC. Another popular methodology known as finite-state morphology was evolved inspired from the idea of two-level morphology. XFST—tthe tool for implementing finite-state morphology—was introduced by Beesley and Karttunen (2003). For the recognition of the valid word form and generation of various grammatical features, finite-state automata and finite-state transducer were used, respectively. Notable research works in the field of finite-state morphology are Beesley (1998, 2003), Megerdoomian (2004). In the early 2000s, the paradigm-based model for the development of the morph analyzer became very popular. It was first proposed by Akshar Bharati (Bharati *et al.* 2002). A paradigm table captures rules for constructing all the possible word forms that can be derived from the given root form along with the grammatical features of each word form. Different paradigm tables are created, and inflected input word is mapped with the corresponding paradigm table to perform the analysis (Bapat, Gune, and Bhattacharyya 2010). The paradigm of research in the field of computational morphology has shifted from the conventional methods to the machine learning-based methods in the last decade. The reason is the requirements of handcrafted rules, dictionaries, paradigm tables, and suffix tables in the standard rule-based and paradigm-based methods. In Anand Kumar *et al.* (2010), the authors treat the problem of morphological analysis as classification task. SVM architecture is used which captures the nonlinear relationship of the grammatical features. A statistical morphological analyzer is developed in the work (Malladi and Mannem 2013; Srirampur, Chandibhamar, and Mamidi 2015). Few unsupervised models have also been experimented for the morphological tagging task. The pioneer work in this field is done by Goldsmith (2005). Another popular unsupervised model is the morfessor (Creutz 2005). Ak and Yildiz (2012) and Shambhavi *et al.* (2011), Narasimhan, Barzilay, and Jaakkola (2015) have also proposed unsupervised morphological analyzer using Trie-based approach and log-linear methods.

https://doi.org/10.1017/nlp.2024.14 Published online by Cambridge University Press

Morphology in general is classified into two types: inflectional and derivational. The inflectional morphology makes a different form of the same word such as plural or past tense. The derivational morphology forms a new word by changing the POS category of the word. The common morphology types are as follows:

- Affixation: Adding an affix (suffix or prefix) (go→goes)
- Compounding: Combining two or more roots in a single word (greenhouse)
- Internal change: Changing part of the root (sit→sat)
- Suppletion: Changing the root completely (good→better)

During the last few years, many deep neural network-based architectures have emerged and also given promising results for numerous challenging NLP tasks. For the computational language processing tasks, researchers have used deep learning models (Yu, Falenska, and Vu 2017; Malaviya, Gormley, and Neubig 2018). In Liu (2021), the authors have done the brief survey on various neural models for the computational morphology related tasks. The paper gives good insights on different architectures and how to leverage them for the tasks such as lemmatization, morphological tagging, and joint learning of lemma and tags. The deep learning models require a huge amount of training data for accurate feature learning. The requirement of large data becomes a challenge for the low-resource languages. In the work (Chakrabarty, Chaturvedi, and Garain 2016), a neural lemmatizer for the Bengali language is discussed. The proposed lemmatizer makes use of contextual information of the surface word to be lemmatized. The work (Heigold, Neumann, and van Genabith 2016) proposes neural morphological tagger modules for morphologically rich languages. This work is extended in Heigold, Neumann, and van Genabith (2017) by obtaining results on fourteen different languages. In Chakrabarty, Pandit, and Garain (2017), a deep neural model is described which identifies the correct word lemma transformation. RNN-based morpheme segmentation model is discussed in Premjith, Soman, and Kumar (2018) for the Malayalam language. In Tkachenko and Sirts (2018), various neural models for morphological tagging are discussed by keeping the same encoder discussed in Lample *et al.* (2016). In Gupta *et al.* (2020), the results of various neural morphological taggers are compared for the Sanskrit language.

In the recent times, transformer-based approaches have become popular for almost all NLP tasks. The core of transformer-based models such as BERT are multilingual pre-trained models. Such models are trained on the large corpus, and then they can be fine-tuned for the specific task. For the task of morphological analysis and lemmatization, transformer-based approaches have been experimented (Kondratyuk 2019; Singh, Rutten, and Lefever 2021; Acikgoz *et al.* 2022).

2.2 Gujarati morphology

Gujarati belongs to the Indo-Aryan language family. The language is derived from Sanskrit—an ancient Indian language. The grammar of Gujarati is similar to other Indo-Aryan languages such as Hindi, Bengali, etc. Gujarati is an agglutinative language. Various suffixes are added to the root to create new word forms expressing grammatical relations. The word order for Gujarati is Subject-Object-Verb (SOV). There are three genders and two numbers (Chauhan and Shah 2021). We now discuss Gujarati noun, verb, and adjective POS categories with their important grammatical features.

Gujarati nouns inflect for the gender, number,and case. Gujarati has three genders: male, female, and neutral; two numbers: singular and plural; and six cases. Table 1 shows various cases along with case markers.

Gujarati verbs take inflections for gender, number, person, tense, aspect, and mood features. Tables 2 and 3 show the examples of verb features.

The features identified for adjectives are type, gender, and number. The type feature describes the nature of the adjectives. In Gujarati, some adjectives inflect with for the gender and number features and other adjectives do not inflect at all. Table 4 shows the examples.

Table 1. List of case markers for Gujarati noun

Case	Suffix
Nominative	ϕ
Genitive	નો,ની,નું,નાં (*Nō,nī,nuṁ,nāṁ*)
Ergative	એ (*ē*)
Objective/Dative	ને (*nē*)
Ablative	થી (*thī*)
Locative	માં (*māṁ*)

Table 2. Examples of moods in Gujarati verb

Mood	Example	Transliteration	English translation
Indicative	રાહુલ ક્રિકેટ રમે છે	*Rāhula krikēṭa ramē chē*	*Rahul is playing cricket*
Imperative	રોજ થોડી કસરત કરજો	*Rōja thōḍī kasarata karajō*	*Do some exercise every day*
Conditional	જો રામ ત્યાં હોત તો સીતાને બચાવી લીધા હોત	*Jō rāma tyāṁ hōta tō sītānē bacāvī līdhā hōta*	*If Rama was there, Sita would have been saved*
subjunctive	કૃણાલ અત્યારે પુસ્તક વાંચતો હશે	*Krṇāla atyārē pustaka vāñcatō haśē*	*Krunal must be reading a book now*

Table 3. Examples of Gujarati verb aspects

Mood	Example	Transliteration	English equivalent
Simple	પારસ મુંબઈમાં રહે છે	*Pārasa mumba'īmāṁ rahē chē*	*Paras lives in Mumbai*
Progressive	કાજલ અત્યારે પૂજા કરી રહી છે	*Kājala atyārē pūjā karī rahī chē*	*Kajal is worshiping right now*
Perfect	મહારાજ એ પાઠ કરી લીધા	*Mahārāja ē pāṭha karī līdhā*	*Maharaj recited it*
Perfect progressive	પરીક્ષા પાસ કરવા માટે સાગર એક વર્ષથી મહેનત કરી રહ્યો હતો	*Parīkṣā pāsa karavā māṭē sāgara ēka varṣathī mahēnata karī rahyō hatō*	*Sagar had been working hard for a year to clear the exam*

Table 4. Gujarati adjective inflection

Type of adjective	Example
Non-inflected	કોમળ (*Kōmaḷa*)
Inflected	કાચો, કાચી , કાચું , કાચા (*Kācō, kācī, kācuṁ, kācā*)

Below are some of the challenges in performing morphological processing for the Gujarati language:

- The standard dataset in which each word is mapped with the corresponding root form along with the grammatical features is not available for the Gujarati language. For the other

https://doi.org/10.1017/nlp.2024.14 Published online by Cambridge University Press

resource-rich languages, such data can be found in the UD Treebank. The lack of dataset created hurdle in experimenting with modern deep neural network-based models.

- Due to the high degree of inflections, the standard rule-based and unsupervised systems cannot be generalized and often fail to handle the ambiguities in the word formation process.
- Due to the differences in the fundamental grammatical rules, the existing models for other languages of a similar family such as Hindi cannot be directly used. For example, Hindi has two genders, while Gujarati has three genders.

As discussed in the literature review section, the approaches for developing morph analyzer can be classified into rule-based and machine learning-based approaches. Due to rich morphology and high degree of ambiguities present in the word formation rules, it is not preferable to use the rule-based methods such as the paradigm approach and Finite State Transducer (FST). The work done in Baxi *et al.* (2015) supports this argument. The standard machine learning approaches require manual feature engineering which is challenging because identifying word-level features from highly inflectional language is a difficult task. By studying various unsupervised models of morphology in the literature, we observe that even though attempts are made to develop morph analyzer using this approach, the best results for the inflectional morphology are obtained only through standard rule-based methods (Hammarström and Borin 2011; Creutz 2005; Goldsmith 2005).

2.3 Baseline system

We consider Gujarati morphological analyzer (Baxi and Bhatt 2021) as baseline system. Their model is also Bi-LSTM based, but the labels are represented using monolithic scheme. The system has following limitations:

- Due to the monolithic label representation, a number of output classes are very high which leads to inefficient training and low accuracy especially for verb POS category.
- We observe incorrect output when two words with similar structure belong to different POS categories. For example, consider the following sentences:
  Sentence 1: મારા નાની ગામડે રહે છે. (Transliteration: Mārā nānī gāmaḍē rahē chē. English Translation: My grandmother lives in a village). Here, the word નાની means grandmother, and the POS category is noun.
  Sentence 2: મેં નાની બચત યોજનામાં રોકાણ કર્યું છે. (Transliteration: Mēṁ nānī bacata yōjanāmāṁ rōkāṇa karyuṁ chē. English Translation: I have invested in a small saving scheme). Here, the word નાની means small, and the POS category is adjective. Such ambiguities are not resolved using monolithic label representation.

3. Proposed approach

In this section, we discuss the motivation and overall idea of the proposed approach in Subsection 3.1. The dataset for the experiments is manually created by us. In Subsection 3.3, we describe the dataset creation process and overall structure of the dataset.

3.1 Outline of the approach

In this section, we discuss the architecture and working of the proposed model used to create Gujarati morph analyzer. Our morph analyzer has two modules: morpheme boundary detection

https://doi.org/10.1017/nlp.2024.14 Published online by Cambridge University Press

Table 5. List of features along with corresponding labels

Feature	Labels
Gender	Male, female, neutral, no gender
Number	Singular, plural, no number
Case	Nominative, dative, ergative, genitive, ablative, locative
Tense	Future, present, past, no tense
Aspect	Simple, perfect, progressive, perfect progressive, no aspect
Person	1st, 2nd, 3rd, no person
Type (adjective)	Inflective, non-inflective

module which separates the root and inflection part from the word and grammatical feature prediction module that assigns morphological features to an inflected word. The morpheme boundary detection model is similar to Baxi and Bhatt (2021) with the enhancement in the dataset. The authors observed that the primary limitation of the system in Baxi and Bhatt (2021) is low accuracy for the morph feature prediction task especially for verb POS category due to the monolithic representation of the labels. We use the different label represent technique in the proposed approach and observe the improvements in the overall results.

We propose the change in the label representation technique through which the number of target classes is reduced and there is improvement in the results for the grammatical feature prediction task. Given any inflected word, the task of grammatical feature prediction module is to tag the word with corresponding grammatical feature values. As describes in the previous section, for the Gujarati language, we have identified various features for the noun, verb, and adjective categories. Each grammatical feature has multiple possible labels. Table 5 shows labels associated with each grammatical feature. A number of labels determine the number of output classes in case of multiclass classification model. In the monolithic label representation used in Baxi and Bhatt (2021), each combination of labels was assigned a class label. For example, the combination male–singular–nominative is one class, the combination female–singular–nominative is another class, and so on. Due to many different possible combinations, a number of output classes are very large due to which the model is not trained properly and results into poor accuracy. The similar observations are also reported by Tkachenko and Sirts (2018) and Chakrabarty *et al.* (2017). To overcome this limitation, we use different label representation technique. In the proposed technique, we represent each feature as separate label instead of combining them together. In this way, a number of classes are equal to number of possible labels for a feature. For example, consider noun POS category which has gender, number, and case feature. In the proposed model, the total number of output classes will be 11 (gender: male, female, and neutral; number: singular, plural; and case: nominative, dative, ergative, genitive, ablative, and locative). Each label can have binary value indicating whether it is present or not. Table 6 shows the reduction in number of classes compared to the previous monolithic representation.

3.2 Model selection

For the task of morphological analysis, we consider the following neural architectures: RNN, Bi-LSTM, Convolutional Neural Network (CNN), and transformer. While CNNs have been successful in image processing tasks, their application in NLP has been relatively limited. Transformer-based approaches are based on large pre-trained models like BERT. Such models are

https://doi.org/10.1017/nlp.2024.14 Published online by Cambridge University Press

Table 6. Comparison of number of classes in baseline and proposed individual label representation technique

POS category	# of classes in monolithic representation (Baxi and Bhatt 2021)	# of classes in Individual feature representation
Noun	36	11
Verb	198	21
Adjective	16	9

fine-tuned for the specific task. For the Gujarati language, such robust models are not available. In the task of morphological analysis, each label may be dependent on the previous predicted label, so we treat it as sequence labeling problem. RNN and Bi-LSTM models are well-suited for the sequential tasks. Due to this, we select Bi-LSTM as our implementation model and also compare the results with RNN and unsupervised model.

3.3 Dataset

A morphological analyzer is a program which takes the inflected word as input and gives corresponding root word and grammatical feature information as an output. In order to implement the morphological analyzer model, we require a dataset to train the system. The dataset must contain inflected words for various POS categories along with their morpheme boundary segmentation and grammatical feature information. Most of the existing models discussed in the previous section uses either Universal Treebank or UniMorph datasets. Universal Dependencies (UD) is a framework for consistent annotation of grammar across different human languages. UD is an open community effort with over 300 contributors producing nearly 200 treebanks in over 100 languages including Hindi (Nivre *et al.* 2016, 2020). The Universal Morphology (UniMorph) project (Kirov *et al.* 2018; McCarthy *et al.* 2020) is a collaborative effort to improve how NLP handles complex morphology in the world's languages. The goal of UniMorph is to annotate morphological data in a universal schema.

Being a low-resource language, the dataset for the Gujarati language was not yet available in the UD treebank or UniMorph datasets so we have created our own dataset and annotated it as per the UniMorph schema. The dataset creation process is published in Baxi and bhatt (2022), Batsuren *et al.* (2022).

3.3.1 Summary of dataset creation steps

For the dataset creation, we first studied the morphology and word formation process of Gujarati. After that, we surveyed about various available corpus for Gujarati and selected TDIL-ILCI-II corpus which contains 30,000 POS tagged sentences. We prepared wordlist files for each POS category. After that, with the help of linguists, for each inflected word, we identified corresponding root morpheme and also associated grammatical feature lists such as gender, person, case, etc. We have also developed the annotation tool using which each word was annotated and the data was prepared in the standard UniMorph format.

Our dataset is divided into two parts: dataset for morpheme boundary segmentation and the dataset for grammatical feature tagging task. In the morpheme boundary segmentation dataset, each word is represented in its root and suffix part using binary encoding. The second part is the dataset for capturing various grammatical features of an inflected word. We included noun, verb, and adjective POS categories and identified various grammatical features associated with them. Table 7 shows the details about the features and number of the words in our dataset.

https://doi.org/10.1017/nlp.2024.14 Published online by Cambridge University Press

Table 7. Details about dataset

POS category	Features	Number of words
Noun	Gender, number, case	6,847
Verb	Gender, number, tense, aspect, person	10,128
Adjective	Gender, number	3,346

Our dataset contains 16,327 unique inflected words with their correct root word mappings. The proposed Gujarati dataset is included in the upcoming version of the UniMorph schema (Batsuren *et al*. 2022). The dataset is publicly available.[b]

4. Experiments and observations

This section gives details about the problem formulation and experiments. Subsections 4.1 and 4.2 describe the experiment setup and working of the model, respectively. In Subsection 4.3, we perform the result analysis of the proposed model and also compare the results with the baseline system.

4.1 Experiment setup

In our morphological analysis, we solve two distinct problems: morpheme boundary detection and grammatical feature tagging. We first discuss how both problems can be formulated as ML classification problems and provide details of data points and labels as described in Jung (2022a).

- Morpheme boundary detection: This problem involves breaking down words into individual characters, which are then represented as character embeddings. Each character serves as a data point, and the goal is to classify whether a given character represents the boundary between two morphemes or not. In other words, it's a binary classification task where the labels are binary, indicating whether there is a morpheme split at a particular character position.
- Grammatical feature tagging: In this task, each word is treated as a data point, and the objective is to assign grammatical features, such as gender, case etc., to that word. This is a multiclass-multilabel problem, meaning that for each word, there can be multiple grammatical features(labels) to assign, and each of these features can have multiple possible classes.

Figure 1a and b shows the architecture of the system for both tasks.

We use the same architecture used in Baxi and Bhatt (2021) for the morpheme boundary detection task. The data is represented in the binary encoded format such that 1 in the string represents the morpheme splitting location. Fig. 2 shows the example of the data representation for the morpheme boundary detection task. Table 8 shows the results obtained for the morpheme boundary detection task. In this example, an inflected word સવારે (English meaning in the morning) is represented. The corresponding root word would be સવાર (English meaning morning).

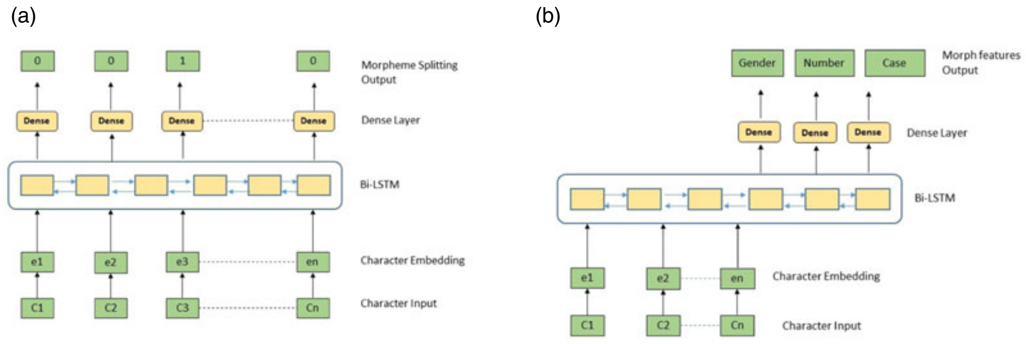
For the grammatical feature prediction task, the format of the training data is such that each word is associated with a binary string. Each value in the string corresponds to individual feature labels as mentioned in Table 5. Once the training data is prepared, we first apply character level tokenization and then apply padding and generate sequences for the input. We split the data for

[b] Gujarati UniMorph dataset: https://github.com/unimorph/guj

https://doi.org/10.1017/nlp.2024.14 Published online by Cambridge University Press

Table 8. Morpheme boundary detection results—POS category wise

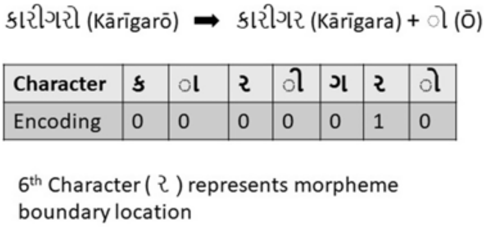
POS category	# of words	Correctly predicted words	Accuracy (%)
Noun	1,369	1,240	90.57
Verb	2,025	1,761	86.96
Adjective	669	645	97.49



System Architecture for the morpheme segmentation task

System Architecture for the grammatical feature prediction task

Figure 1. System architecture for the morpheme segmentation and grammatical feature prediction task, respectively.



કારીગરો (Kārīgarō) ➡ કારીગર (Kārīgara) + ો (Ō)

Character	ક	ા	ર	ી	ગ	ર	ો
Encoding	0	0	0	0	0	1	0

6th Character (ર) represents morpheme boundary location

Figure 2. Example of encoding for the identification of morpheme boundary.

the training and testing and define the model as per the model architecture mentioned in the previous section.

We use the following model architecture:

- Layer 1: Character level embedding layer, dimensions—50.
- Layer 2: Bidirectional LSTM: Hidden layer size: 128. For the morpheme segmentation task, return _sequences argument is True.
- Layer 3: Dropout layer for the prevention of overfitting.
- Layer 4: Dense layer: activation sigmoid. Numbers of output neurons are equal to number of features for the grammatical feature prediction task and is 1 for the morpheme boundary detection task.
- Loss = binary cross entropy.

https://doi.org/10.1017/nlp.2024.14 Published online by Cambridge University Press

Table 9. Comparison of accuracy F1 scores—monolithic and individual feature representation

	Monolithic representation		Individual feature representation	
POS category	Accuracy	F1 score	Accuracy	F1 score
Noun	70.64	0.68	99.95	1
Verb	16.18	0.12	78.76	0.84
Adjective	85.85	0.68	99.84	0.99

4.2 Working of the model

As highlighted in the previous sections, the morphological analyzer performs two tasks. The morpheme boundary detection task is treated as sequence labeling problem in which each character of the input word is assigned a label indication whether the character is split point or not. Once the split point is identified, the root and suffix part can be identified. Each character is converted into correspondingly character embeddings before the Bi-LSTM layer. The final dense layer assigns binary label 0 or 1 indicating the split point.

The second task of grammatical feature prediction is treated as multiclass-multilabel classification problem. Depending on the number of features for the particular POS, output classes are decided. Similar to the morpheme boundary detection task, the character embeddings are given as an input to the Bi-LSTM layer, and the dense layer predicts the output classes for each feature.

4.3 Evaluation

In this section, we discuss the results obtained through the proposed individual feature representation technique for the grammatical feature prediction task and compare the obtained results with the monolithic label representation. Table 9 shows the comparison of F1 scores and accuracy between monolithic and proposed individual feature representation technique. We have also performed the grammatical feature tagging experiments individually for each POS category—feature combination. The purpose of this analysis is to observe how each feature and POS category contributes to the overall results. The results are shown in Table 10. We observe from the results that for the noun and adjective POS categories, the proposed system gives good results for all the grammatical features. However, for the verb POS category, the system does not give good results specifically for gender and person features. In this way, we are able to identify the exact features which are not learned properly by the system. This may be due to ambiguities at the word formation level or insufficient examples related to the particular grammatical feature. In this table, accuracy denotes the percentage of examples for which all the features are classified correctly. Because the training data is slightly unbalanced, the precision, recall, and F1 measures are used. Figure 3 shows the visual representation of improvement in the result accuracy due to proposed system. We note that there is huge improvement in the results due to the proposed change in the label representation. From the machine learning perspective, we identify the following reasons for the performance improvement in the proposed approach compared to the baseline model:

- Feature representation: As highlighted in Table 6, due to individual label representation, a number of output classes are reduced. This reduction makes the model more powerful and allows better feature interaction allowing the model for learning more complex patterns.
- Better generalization: Because of the proposed change in the label representation, the model is better generalized for the unseen data.

https://doi.org/10.1017/nlp.2024.14 Published online by Cambridge University Press

Table 10. Results for morphological feature tagging task for individual features using proposed label representation technique—POS category and feature wise

POS category	Feature	Accuracy (%)	Precision	Recall	F1 score
Noun	Gender	100	1	1	1
	Number	100	1	1	1
	Case	99.85	1	1	1
Verb	Gender	63.93	0.76	0.72	0.73
	Number	83.9	0.89	0.89	0.89
	Person	57.22	0.7	0.65	0.67
	Tense	100	1	1	1
	Aspect	88.79	0.91	0.91	0.91
Adjective	Type	100	1	1	1
	Gender	99.54	0.98	0.98	0.98
	Number	100	1	1	1

Table 11. Analysis of training and validation errors for baseline and present system

Approach/loss	Baseline	Present
Training loss	0.18	0.096
Validation loss	0.19	0.079

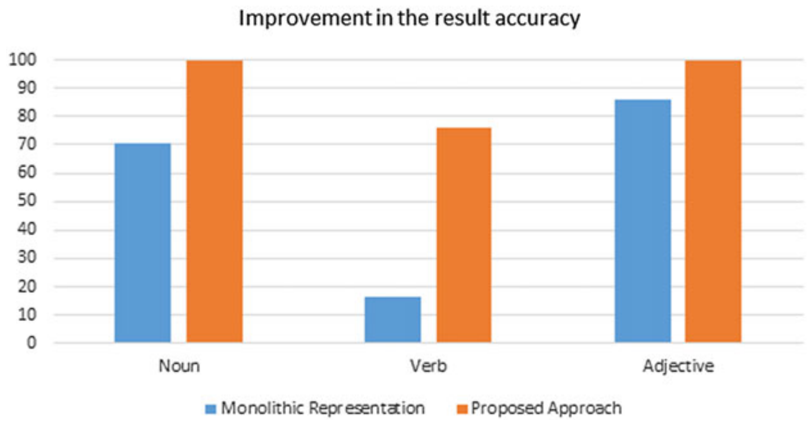


Figure 3. Comparison of result accuracy between the monolithic representation and individual feature representation.

- Error analysis: We can diagnose the performance of the model by analyzing the training and validation errors, as outlined in the study by Jung (2022b). Table 11 presents a comparison of training and validation loss between the baseline model and the current system.

From the error analysis table, we make the following observations:

- The loss in the current approach has shown a substantial reduction compared to the baseline. It suggests that the current approach performs better than the baseline system.

https://doi.org/10.1017/nlp.2024.14 Published online by Cambridge University Press

Table 12. Comparison of the results of proposed approach with other deep neural network architectures—RNN and LSTM

		Results in accuracy (%)		
POS	Feature	RNN	LSTM	Bi-LSTM (proposed)
Noun	Gender	66.47	66.47	100
	Number	92.96	92.55	100
	Case	96.42	96.35	99.85
Verb	Person	49.72	48.38	57.22
	Gender	62.51	62.9	63.93
	Tense	100	100	100
	Aspect	87.29	88	88.79
	Number	80.43	80.98	83.9
Adjective	Gender	91.03	92.38	99.54
	Number	85.65	84.01	100
	Type	84.01	85.2	100

Table 13. Comparison of results obtained from neural model and unsupervised model

POS category	Unsupervised approach	Bi-LSTM monolithic representation	Bi-LSTM individual feature representation
		Accuracy	
Noun	68.27	70.64	99.95
Verb	12.95	16.18	78.76
Adjective	25.72	85.85	99.84

- The difference between the training and validation loss remains less than 0.1. This minimal difference indicates that there is no overfitting.

The proposed model uses the Bi-LSTM model. It is important to compare the results of the proposed model with the other popular deep learning architectures. We have compared the results of the proposed model with vanilla RNN and simple LSTM models, and the results are shown in Table 12. It is clear from the comparison that the proposed model performs better than the other deep learning architectures. In Table 13, we compare the results of the proposed Bi-LSTM-based neural model (both monolithic and individual feature representation) with the unsupervised model. The results of morpheme boundary detection with the unsupervised model are taken from Baxi and Bhatt (2021). The unsupervised model was implemented using the morfessor model (Creutz 2005) and can be used through the Indic-NLP library (Kunchukuttan 2020). From the comparison, it is clearly inferred from the results that the neural model using proposed individual feature representation technique outperforms the unsupervised model by a large margin.

We observe that for the verb POS category, the system gives correct output for the examples where multiple suffixes attach to the root verb as shown in Table 14. We did the manual analysis of the incorrectly predicted words to understand the possible linguistic angle behind the incorrect output. We make note of the following points:

https://doi.org/10.1017/nlp.2024.14 Published online by Cambridge University Press

Table 14. System output prediction for the words containing multiple suffix attachments

Word	System output
કરતી હતી (Karti hati) (Was doing)	Root: કર, female, 3rd person, past progressive
કરી લીધું છે (Kari lidhu che) (Done)	Root: કર, present perfect
કરી લીધું હશે (Kari lidhu hashe) (Would have done)	Root: કર, future perfect

Table 15. Examples of incorrect root identification

	Root morpheme detected	
Inflected word	by the system	Actual root word
દેખાશે (*Dēkhāśē*) (will be seen) → દેખા (*dēkhā*) + શે (*śē*)	દેખા (*dēkhā*)	દેખા (*dēkhā*) + વું (*vu ṁ*) → દેખાવું (*Dēkhāvu ṁ*)
છોકરા (*Chōkarā*) (Boys) → છોકર (*chōkara*) + ◌ા (*ā*)	છોકર (*chōkara*)	છોકર (*chōkara*) + ◌ુ (*U*) + ◌ં (*ṁ*) → છોકરું (*Chōkaru ṁ*)
ધંધાનું (*Dhandhānu ṁ*) (Of business)→ ધંધ (*dhandha*) + ◌ાનું (*ānu ṁ*)	ધંધ (*dhandha*)	ધંધ (*dhandha*) + ◌ો (Ō) → ધંધો (*dhandhō*)
ઈશારા (*Īśārā*) (Hints)→ ઈશાર (*iśāra*) + ◌ા (*ā*)	ઈશાર (*iśāra*)	ઈશાર (*iśāra*) + ◌ો (Ō) → ઈશારો (*Īśārō*)

- For the morpheme segmentation task, the rules to form valid root word are different for each POS category and they also depend on the grammatical features. The output obtained by the system may be a valid stem but may not be correct root word. In future, the neural method can be used with rule-based method in combination to address this issue. Table 15 highlights such examples.
- Ambiguities in the word formation rules and attachment of multiple suffixes lead to incorrect morpheme splitting output.
- The feature prediction model captures the pattern of inflections. In some cases, the grammatical features do not have any correlation with the attached suffixes. Due to these exceptions, some errors are present in the grammatical feature prediction output. For example, consider two adjectives: સારી (Sārī) and અજ્ઞાની (Ajñānī). Both have ◌ી (Ī) suffix which indicates female gender, but the adjective અજ્ઞાની (Ajñānī) does not demonstrate any gender; it can be used in the same form for all genders.

5. Conclusion and future scope of research

We conclude that our neural morph analyzer for the Gujarati language performs very well across the POS categories and addresses the limitations of the baseline system. The dataset created by us can be used by the researchers to carry out further experiments related to the computational morphology tools for Gujarati. The linguistic analysis of the results helped in understanding the morphological complexities of the Gujarati language. We plan to increase the size of the training data in the future to enhance the results of the current system. We also plan to accept the input at the sentence level rather than the current word-based input. This will help to study the sentence-level dependencies for the morphological analysis. We also plan to predict the POS category and morphological features of the word side by side and in order to improve the accuracy of the existing system and to study the effect of POS tagging on the prediction of morphological features. The morph analyzer model can be used to enhance the accuracy of the downstream NLP tasks for the Gujarati language in the future.

References

Acikgoz E.C., Chubakov T., Kural M., Şahin G. and Yuret D. (2022). Transformers on multilingual clause-level morphology. In *Proceedings of the 2nd Workshop on Multi-lingual Representation Learning (MRL)*, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics, pp. 100–105.

Ak K. and Yildiz O.T. (2012). Unsupervised morphological analysis using tries. In *Computer and Information Sciences II*, pp. 69–75.

Anand Kumar M., Dhanalakshmi V., Soman K. and Rajendran S. (2010). A sequence labeling approach to morphological analyzer for Tamil language. *International Journal on Computer Science and Engineering* **02**(06), 1944–1951.

Bapat M., Gune H. and Bhattacharyya P. (2010). A paradigm-based finite state morphological analyzer for Marathi. In *Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing*, Beijing, China. Coling 2010 Organizing Committee, pp. 26–34.

Batsuren K., Goldman O., Khalifa S., Habash N., Kieraś W., Bella G., Leonard B., Nicolai G., Gorman K., Ate Y.G., Ryskina M., Mielke S., Budianskaya E., El-Khaissi C., Pimentel T., Gasser M., Lane W.A., Raj M., Coler M., Samame J.R.M., Camaiteri D.S., Rojas E.Z., Francis D.L., Oncevay A., Bautista J.L., Villegas G.C.S., Hennigen L.T., Ek A., Guriel D., Dirix P., Bernardy J.-P., Scherbakov A., Bayyr-ool A., Anastasopoulos A., Zariquiey R., Sheifer K., Ganieva S., Cruz H., Karahóğa R., Markantonatou S., Pavlidis G., Plugaryov M., Klyachko E., Salehi A., Angulo C., Baxi J., Krizhanovsky A., Krizhanovskaya N., Salesky E., Vania C., Ivanova S., White J., Maudslay R.H., Valvoda J., Zmigrod R., Czarnowska P., Nikkarinen I., Salchak A., bhatt b., Straughn C., Liu Z., Washington J.N., Pinter Y., Ataman D., Wolinski M., Suhardijanto T., Yablonskaya A. and Stoehr N. (2022). Unimorph 4.0: universal morphology. In *Proceedings of the Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association, pp. 840–855.

Baxi J. and Bhatt B. (2021). Morpheme boundary detection & grammatical feature prediction for Gujarati: dataset & model. In *Proceedings of the 18th International Conference on Natural Language Processing*, NIT, Silchar.

Baxi J. and bhatt B. (2022). Gujmorph - a dataset for creating Gujarati morphological analyzer. In *Proceedings of the Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association, pp. 7088–7095.

Baxi J., Patel P. and Bhatt B. (2015). Morphological analyzer for Gujarati using paradigm based approach with knowledge based and statistical methods. In *Proceedings of the 12th International Conference on Natural Language Processing*, Trivandrum, India. NLP Association of India, pp. 178–182.

Beesley K. (2003). Finite-state morphological analysis and generation for aymara: project report. In *Proceedings of the Workshop of Finite-State Methods in Natural Language Processing, 10th Conference of the European Chapter of the Association for Computational Linguistics*, vol. **5**, pp. 2–5.

Beesley K. and Karttunen L. (2003). *Finite-State Morphology*, Bibliovault OAI Repository, the University of Chicago Press.

Beesley K.R. (1998). Arabic morphology using only finite-state operations. In *Proceedings of the Workshop on Computational Approaches to Semitic languages. Association for Computational Linguistics*, p. 50.

Beesley K.R. and Karttunen L. (1992). Two-Level Rule Compiler. Technical Report. Xerox Palo Alto Research Center. Palo Alto, California.

Bharati A., Chaitanya V., Sangal R. and Gillon B. (2002). *Natural Language Processing: A Paninian Perspective*. Prentice Hall of India.

Byrd R.J., Klavans J.L., Aronoff M. and Anshen F. (1986). Computer methods for morphological analysis. In *Proceedings of the 24th Annual Meeting on Association for Computational Linguistics, ACL'86*, USA. Association for Computational Linguistics, vol. **86**, pp. 120–127.

Chakrabarty A., Chaturvedi A. and Garain U. (2016). A neural lemmatizer for Bengali. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, Portorož, Slovenia. European Language Resources Association (ELRA), pp. 2558–2561.

Chakrabarty A., Pandit O.A. and Garain U. (2017). Context sensitive lemmatization using two successive bidirectional gated recurrent networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada. Association for Computational Linguistics, pp. 1481–1491.

Chauhan U. and Shah A. (2021). Improving semantic coherence of Gujarati text topic model using inflectional forms reduction and single-letter words removal. *ACM Transactions on Asian and Low-Resource Language Information Processing* **20**(1), 1–18.

Creutz M. and Lagus K. (2005). Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0. Computer and Information Science, Helsinki University of Technology.

https://doi.org/10.1017/nlp.2024.14 Published online by Cambridge University Press

Goldsmith J. (2005). Unsupervised learning of the morphology of a natural language. *Computational Linguistics* **27**(2), 153–198.

Gupta A., **Krishna A.**, **Goyal P. and Hellwig O.** (2020). Evaluating neural morphological taggers for Sanskrit. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Association for Computational Linguistics, pp. 198–203, Online.

Hammarström H. and Borin L. (2011). Unsupervised learning of morphology. *Computational Linguistics* **37**(2), 309–350.

Heigold G., **Neumann G. and van Genabith J.** (2016). Neural Morphological Tagging from Characters for Morphologically Rich Languages. arXiv e-prints, arXiv: 1606.06640.

Heigold G., **Neumann G. and van Genabith J.** (2017). An extensive empirical evaluation of character-based morphological tagging for 14 languages. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Long Papers, Valencia, Spain. Association for Computational Linguistics, vol. **1**, pp. 505–513.

Jung A. (2022a). Components of ml. In *Machine Learning: The Basics*. Springer, pp. 19–56.

Jung A. (2022b). Model validation and selection. In *Machine Learning: The Basics*. Springer, pp. 113–134.

Kirov C., **Cotterell R.**, **Sylak-Glassman J.**, **Walther G.**, **Vylomova E.**, **Xia P.**, **Faruqui M.**, **Mielke S.J.**, **McCarthy A.**, **Kübler S.**, **Yarowsky D.**, **Eisner J. and Hulden M.** (2018). UniMorph 2.0: universal morphology. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Kondratyuk D. (2019). Cross-lingual lemmatization and morphology tagging with two-stage multilingual BERT fine-tuning. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Florence, Italy. Association for Computational Linguistics, pp. 12–18.

Koskenniemi K. (1984). A general computational model for word-form recognition and production. In *Proceedings of the 4th Nordic Conference of Computational Linguistics (NODALIDA 1983)*, Uppsala, Sweden, Sweden. Centrum för datorlingvistik, Uppsala University, pp. 145–154.

Kunchukuttan A. (2020). The IndicNLP Library. Available at https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf

Lample G., **Ballesteros M.**, **Subramanian S.**, **Kawakami K. and Dyer C.** (2016). Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California. Association for Computational Linguistics, pp. 260–270.

Liu L. (2021). Computational morphology with neural network approaches.

Magueresse A., **Carles V. and Heetderks E.** (2020). Low-resource Languages: A Review of Past Work and Future Challenges. arXiv e-prints, arXiv: 2006.07264.

Malaviya C., **Gormley M.R. and Neubig G.** (2018). Neural factor graph models for cross-lingual morphological tagging. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia. Association for Computational Linguistics, pp. 2653–2663.

Malladi D.K. and Mannem P. (2013). Context based statistical morphological analyzer and its effect on Hindi dependency parsing. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, Seattle, Washington, USA. Association for Computational Linguistics, pp. 119–128.

McCarthy A.D., **Kirov C.**, **Grella M.**, **Nidhi A.**, **Xia P.**, **Gorman K.**, **Vylomova E.**, **Mielke S.J.**, **Nicolai G.**, **Silfverberg M.**, **Arkhangelskiy T.**, **Krizhanovsky N.**, **Krizhanovsky A.**, **Klyachko E.**, **Sorokin A.**, **Mansfield J.**, **Ernštreits V.**, **Pinter Y.**, **Jacobs C.L.**, **Cotterell R.**, **Hulden M. and Yarowsky D.** (2020). UniMorph 3.0: universal morphology. In *Proceedings of the 12th Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association, pp. 3922–3931.

Megerdoomian K. (2004). Finite-state morphological analysis of Persian. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages. Association for Computational Linguistics*, p. 35.

Minnen G., **Carroll J. and Pearce D.** (2001). Applied morphological processing of English. *Natural Language Engineering* **7**(3), 207–223.

Narasimhan K., **Barzilay R. and Jaakkola T.** (2015). An unsupervised method for uncovering morphological chains. *Transactions of the Association for Computational Linguistics* **3**, 157–167.

Nivre J., **de Marneffe M.-C.**, **Ginter F.**, **Goldberg Y.**, **Hajič J.**, **Manning C.D.**, **McDonald R.**, **Petrov S.**, **Pyysalo S.**, **Silveira N.**, **Tsarfaty R. and Zeman D.** (2016). Universal dependencies v1: a multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, Portorož, Slovenia. European Language Resources Association (ELRA), pp. 1659–1666.

Nivre J., **de Marneffe M.-C.**, **Ginter F.**, **Hajič J.**, **Manning C.D.**, **Pyysalo S.**, **Schuster S.**, **Tyers F. and Zeman D.** (2020). Universal dependencies v2: an evergrowing multilingual treebank collection. arXiv preprint arXiv: 2004.10643.

Patel C. and Gali K. (2008). Part-of-speech tagging for Gujarati using conditional random fields. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages.*

Patel P., **Popat K. and Bhattacharyya P.** (2010). Hybrid stemmer for Gujarati. In *Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing*, Beijing, China. Coling 2010 Organizing Committee, pp. 51–55.

https://doi.org/10.1017/nlp.2024.14 Published online by Cambridge University Press

Premjith B., **Soman K.P. and Kumar M.A.** (2018). A deep learning approach for Malayalam morphological analysis at character level. *Procedia Computer Science* **132**, 47–54.

Raval D., **Pathak V.**, **Patel M. and Bhatt B.** (2020). End-to-end automatic speech recognition for Gujarati. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, Patna, India. Indian Institute of Technology Patna, NLP Association of India (NLPAI), pp. 409–419.

Shambhavi B.R., **P R.K.**, **Srividya K.**, **Jyothi B.J.**, **Kundargi S. and Shastri G.** (2011). Kannada morphological analyser and generator using trie. *IJCSNS International Journal of Computer Science and Network Security* **11**(1), 112–116.

Singh P., **Rutten G. and Lefever E.** (2021). A pilot study for BERT language modelling and morphological analysis for ancient and medieval Greek. In *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, Punta Cana. Dominican Republic (online). Association for Computational Linguistics, pp. 128–137.

Srirampur S., **Chandibhamar R. and Mamidi R.** (2015). Statistical morph analyzer (SMA++) for Indian languages. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pp. 103–109.

Suba K., **Jiandani D. and Bhattacharyya P.** (2011). Hybrid inflectional stemmer and rule-based derivational stemmer for Gujarati. In *Proceedings of the 2nd Workshop on South and Southeast Asian Natural Language Processing (WSSANLP)*, pp. 1–8.

Tang X. (2006). English morphological analysis with machine-learned rules. In *Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation*, Wuhan, China. Huazhong Normal University, Tsinghua University Press, pp. 35–41.

Tkachenko A. and Sirts K. (2018). Modeling composite labels for neural morphological tagging. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, Brussels, Belgium. Association for Computational Linguistics, pp. 368–379.

Yu X., **Falenska A. and Vu N.T.** (2017). A general-purpose tagger with convolutional neural networks. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, Copenhagen, Denmark. Association for Computational Linguistics, pp. 124–129.

Cite this article: Baxi J and Bhatt B. A bidirectional LSTM-based morphological analyzer for Gujarati. *Natural Language Processing* https://doi.org/10.1017/nlp.2024.14

https://doi.org/10.1017/nlp.2024.14 Published online by Cambridge University Press