Morphological Analysis and Disambiguation of Agglutinative Assamese through Rule-Based Synthesis and Neural Candidate Ranking

Pragyanur Saikia[†], **Utpal Sharma, Arindam Karmakar**
Department of Computer Science and Engineering
Tezpur University, India
csp23110@tezu.ac.in[†], {utpal, arindam}@tezu.ernet.in

Abstract

Assamese is a highly agglutinative language that uses suffix sequences to convey various kinds of information. A single morpheme may exhibit multiple syntactic or semantic functions depending on its position in a word and the preceding morpheme. Such ambiguities and the inadequacy of supervised data make morphological analysis difficult for Assamese. Our approach is based on the observation that a word has a limited set of valid morphological analyses, among which one is contextually the most appropriate. We curate a morphological lexicon, a bigram transition rule set, and several derivational rules, which our rule-based analyzer uses to generate valid analyses of a word by segmenting its surface form. The rule-based analyzer facilitates context-based annotation of over 11.8K tokens across more than 1K sentences. We construct morphological feature vectors from each analysis of a word and train a neural model to discriminate between correct and incorrect sequences of morphological vectors. The trained model is evaluated on the task of assigning the highest score to the correct sequence of morphological vectors among all the candidate sequences. The model achieves a word-level disambiguation accuracy of 80.76% by resolving 1045 out of 1294 ambiguities on the test set without using lexical semantics. The results indicate regularity in sentence-level morphological patterns. The annotated dataset and other resources are available at https://github.com/Pragyanur/morphological-analysis-assamese.

1 Introduction

Like many other Indian languages Assamese is agglutinative, form word-groups, but otherwise exhibit a free word-group ordering (Akshar et al., 1995). In such scenarios position-based parsing algorithms are not sufficient for understanding the sentence structure, which is often required for tasks like shallow semantic parsing, information extraction, and other downstream applications. However, the deficiency in strict parse trees is compensated by the rich suffixation phenomena of such languages, thereby making automatic morphological analyzers essential (Seeker and Kuhn, 2013; Yildiz et al., 2016; Sharma et al., 2008). The suffix sequences in Assamese provide various information about the preceding word-groups, like case, quantity, number, person, tense, emphaticity, etc. and helps in sentence understanding (Roy, 2017; Das et al., 2019). An example for such phenomena is given in Figure 2. Prefixation phenomenon in Assamese is comparatively less diverse and prefixes do not concatenate.

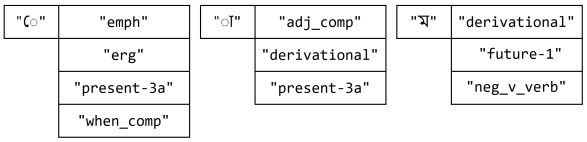


Figure 1: Example of multi-functional surface form of morphemes in the Assamese language.

In Assamese and other agglutinative languages the suffixation on a root morpheme is guided by rules (Das et al., 2019; Yildiz et al., 2016). However, a single surface realization may belong to multiple classes and perform different functions based on the type of root it attaches to (Yildiz et al., 2016). It is common in Assamese for a suffix to serve multiple functions (see Figure 1) (Das et al., 2019; Shen et al., 2016). In investigation of Assamese derivational suffixes, Gayan and Kumar Nath (2025) discovered that some derivational and inflectional suffixes have the same surface realization. Such a suffix might derive the root word into another form, subsequently affecting the succeeding suffixation. Besides multi-functional affixes, ambiguity of root morphemes amplify the problem of producing multiple possible analyses of a single word.

To address these problems, we present a hybrid

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 45155–45167
October 24-29, 2026 ©2026 Association for Computational Linguistics

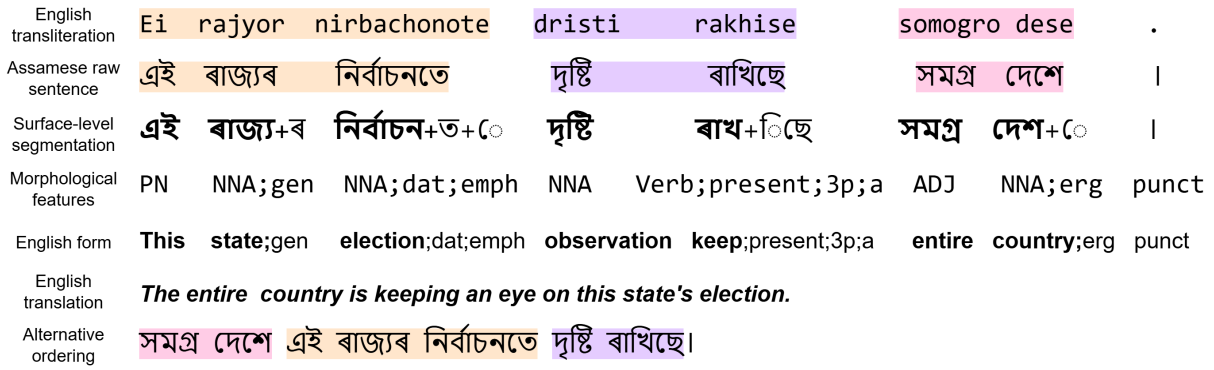


Figure 2: Morphological analysis of a complete Assamese sentence. The highlighted words illustrate word-groups that can exhibit free ordering while maintaining the same meaning.

pipeline comprising a linguistically grounded rule-based analyzer that produces all possible analyses of a word, followed by a disambiguation model that relies on sentence structure rather than semantics. Previous work has shown that context plays a crucial role in the morphological disambiguation of agglutinative languages (Yildiz et al., 2016; Shen et al., 2016). Therefore, we hypothesize that the morphological analyses of the context words in a sentence can be used to identify the correct analysis of an ambiguous word. Our formulation is based on the observation that the part-of-speech (PoS) tags and morphological features of words in a sentence, when correctly assigned, should conform to regularities that can be viewed as a *sentence-level morphological grammar*. To test this hypothesis, we conduct experiments on a manually annotated dataset and obtain promising results. In the following sections, we discuss the related works, linguistic knowledge incorporated into our approach, the hybrid pipeline, the methodology, and the experimental results.

2 Related Works

Morphological disambiguation has been extensively studied for morphologically rich and highly agglutinative languages. Shen et al. (2016) proposed a language-agnostic neural approach for morphological disambiguation that leverages contextual information, and evaluated their method on Turkish, Russian, and Arabic. Similar context-based approaches have also been explored for languages such as German, French, and Kazakh (Yildiz et al., 2016; Kessikbayeva and Cicekli, 2014). Adapting such approaches directly to Assamese is challenging due to lack of large annotated datasets (Shen et al., 2016), substantial linguistic differences and the difficulty of transferring language-specific morphological knowledge across languages.

Context-aware morphological disambiguation has also been explored for several Indian languages. Sudhakar and Singh (2017) conducted experiments on Hindi using multiple neural models that exploit the context and surface form of the target word to disambiguate between its possible morphological analyses. Their approach relies on a large manually annotated dataset from the Hindi-Urdu Dependency Treebank (HUTB) (Sudhakar and Singh, 2017). Dasari et al. (2023) created a sentence-level morphologically annotated dataset for Telugu and evaluated pretrained language models for morpheme-tag classification. Their experiments showed that monolingually pretrained models outperform multilingual models (Dasari et al., 2023). Given the limited availability of large annotated datasets for Assamese and its linguistic differences from these languages, we propose a hybrid approach that combines rule-based morphological synthesis with neural disambiguation to exploit contextual structure without relying on semantic representations.

Previous research on Assamese morphology has primarily focused on identifying morphological rules and linguistic characteristics (Roy, 2017; Das et al., 2019). Sharma et al. (2008) investigated morphology acquisition from corpus data by decomposing words into a root followed by a sequence of suffixes, demonstrating that a single word can admit multiple valid decompositions. However, even when a word has a unique decomposition, determining the correct morphological tag often requires sentence-level contextual understanding.

Relevant datasets and resources for Assamese in-

clude: (1) UniMorph[1] data, which provide morphological features for a subset of Assamese noun and verb forms; (2) an Apertium-based resource[2] containing a limited set of morpheme-concatenation rules; and (3) the 73-sentence Assamese Universal Dependencies Treebank (AUDTB)[3]. The UniMorph data provide standardized morphological feature labels but do not provide alternative analyses or contextual information for morphological disambiguation. The Apertium-based rules provide a useful foundation for morphological analysis, which we extend to incorporate additional morphological phenomena. AUDTB provides manually annotated morphological features without explicit morphological segmentation, resulting in some inconsistencies in the analyses of identical surface forms. In contrast, our annotated dataset provides context-aware morphological analyses grounded in a rule-based morphological analyzer, thereby providing consistent analyses across occurrences of the same forms within the analyzable portion of the corpus.

To the best of our knowledge, this dataset is a novel contribution, and we are not aware of prior work on context-aware, sentence-level morphological disambiguation for Assamese.

3 Assamese Morphology

Affixation is a major phenomenon in Assamese, and natural language processing tasks remain challenging in the absence of morphological analysis (Sharma et al., 2008; Gogoi, 2020). Disambiguation of suffixes is particularly important for determining the PoS tags of Assamese words, as derivational processes are highly prevalent in the language (Roy, 2017; Gogoi, 2020). In surface realizations, affixes frequently undergo orthographic changes due to the phonological properties of adjacent morphemes at their boundaries. These transformations can result in multiple surface forms realizing the same morphological function (see Figure 3). In our approach, such surface variants are mapped to a unified morphological tag category.

Besides surface level variations, we observe Assamese morphology to largely follow the Item-and-Arrangement model, where words are formed through structured sequences of morphemes. The arrangement of these morphemes is governed by

সংস্থাটোৱে	['সংস্থা', 'Abstract Noun'], ['টো', 'number'], ['ৱে', 'erg']
উপায়ুক্তই	['উপায়ুক্ত', 'Common Noun'], ['ই', 'erg']
গৱেষকসকলে	['গৱেষক', 'Common Noun'], ['সকল', 'number'], ['ে', 'erg']

Figure 3: Multiple surface level realization of the ergative suffix in Assamese

morphotactic constraints, many of which have not yet been comprehensively formalized. Some of these rules have been discussed in prior work on the computational morphology and syntax of Assamese (Saharia, 2014). The apertium-based work, mentioned earlier, also introduced some rules of morpheme concatenation. In this work, we emphasize the importance of morpheme subcategorization (like `erg`, `dat`, and `loc` case-tags instead of just `case` tag) and additional morphotactic constraints beyond those previously documented. For example, case suffixes generally do not concatenate and word boundaries are constrained.

Pronouns in Assamese inherently encode case, person, honorificity, and other grammatical information (see Figure 4). Similarly, verbal suffixes express a diverse range of grammatical functions. Certain suffixes encode tense, person, and honorific agreement, which we categorize using tags such as `present_2i` (present tense, second person, informal) and `future_3a` (future tense, third person, any). Verbal suffixes can also express relational or compositional functions. We identify several such functions and categorize them under tags such as `adj_comp`, `when_comp`, `reason_comp`, and `after_comp`. Other verbal features, such as mood and aspect, are also present in Assamese, but we do not model them in our approach.

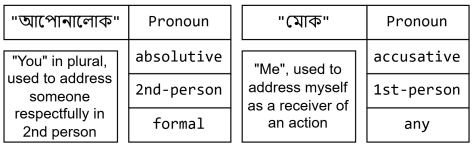


Figure 4: Example of case and other information integrated with pronouns.

The above properties give rise to phenomena related to government and agreement, which are prevalent in Assamese sentence structures. In particular, verb morphology exhibits agreement with nominal or pronominal subjects in terms of person, case, and honorificity. Additionally, factors such as nominal animacy and verb category (e.g., transitive or intransitive) influence the selection of case suffixes (Bhattacharjya, 2012). These linguistic char-

[1] https://github.com/unimorph/asm
[2] https://github.com/apertium/apertium-asm-ben
[3] https://github.com/UniversalDependencies/UD_Assamese-AiW/tree/master

acteristics along with the word-grouping characteristics are advantageous for our proposed approach because these patterns provide strong contextual cues for resolving ambiguity among multiple analyses. Other computationally relevant characteristics are discussed in Appendix B.

4 Rule-based Analyzer

Our rule-based analyzer utilizes *XOBDO*[4], a collaboratively developed lexical resource containing base forms[5] annotated with PoS tags and their dictionary meanings. To improve morphological coverage, we augment this lexicon with multiple surface realizations of 35 suffix categories. These suffix variants were extracted from the Assamese Wikipedia corpus using a suffix-stripping approach combined with manual linguistic analysis (Sharma et al., 2008). The resulting morphological lexicon[6] represents each morpheme as $M : \{T_1, T_2, \ldots\}$, where M denotes the surface morpheme and T_i denotes one of its possible morphological tags.

In addition to suffixes, we incorporate a limited set of prefixes belonging primarily to two categories: (i) negation prefixes, which occur with a restricted set of verbs, and (ii) numerical prefixes, which typically attach to determiners. Due to the multi-functional nature of Assamese morphology, several morphemes belong to multiple grammatical categories, including PoS classes. A detailed description of the morpheme tags and associated linguistic features is provided in Appendix A.

The primary objective of the analyzer is to generate all valid morphological analyses for a given word. To achieve this, we incorporate both derivational and inflectional morphotactic constraints within the analysis process. The major components of the analyzer are described in the following subsections.

4.1 Derivational Rules

We observed the presence of derived words in the *XOBDO* lexicon. Therefore, we apply the suffix-stripping method in order to investigate the presence of derivational suffixes (Sharma et al., 2008). We take a very large rule frequency (50) to exclude spurious matches. We obtained 49 derivational suffixes that each convert at least 50 base words into derived lexical forms. Many of these suffixes match

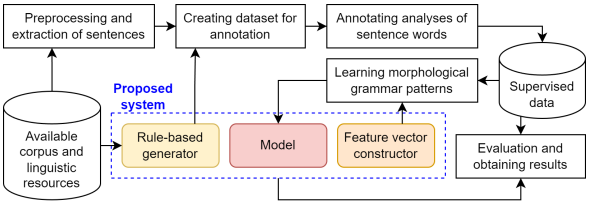


Figure 5: Overview of the proposed methodology with various components of the proposed system. The linguistic knowledge has been incorporated into the rule-based generator and the feature vector constructor.

with those discussed by Roy (2017) and Gayan and Kumar Nath (2025). For each derivational suffix s, we estimate the probability of a PoS transition from tag T_A to tag T_B using lexicon-level transition frequencies:

$$P(T_A \rightarrow T_B \mid s) = \frac{Count(T_A \rightarrow T_B \mid s)}{\sum_i \sum_j Count(T_i \rightarrow T_j \mid s)}$$

where $Count(T_A \rightarrow T_B \mid s)$ denotes the number of observed derivational transformations in which suffix s converts a base word of category T_A into a derived form of category T_B. During inference, the most probable derivational transformation for any tag T induced by suffix s is selected, to limit the number of valid analyses.

$$T_{derived} = \underset{i}{\operatorname{argmax}}\, P(T \rightarrow T_i \mid s)$$

4.2 Bigram Transition Rule-set

To constrain the generation of morphologically valid analyses, we manually constructed a morphotactic transition rule-set describing valid category-level transitions between adjacent morphemes belonging to tags T_i and T_j. The rule-set was developed using linguistic analysis of Assamese suffixation patterns and with the help of previous studies (Saharia, 2014; Das et al., 2019). Special boundary states corresponding to word-start (ws) and word-end (we) were additionally introduced to model valid word initialization and termination conditions. The rule-set is converted into a transition matrix $\mathcal{R} \in \{0,1\}^{n \times n}$, where $n = 62$ is the number of tags associated with the rules. The matrix is used by the analysis generator, with $\mathcal{R}(T_i, T_j) = 1$ indicating that the transition $T_i \rightarrow T_j$ is valid. The rule-set can be extended to capture longer n-gram patterns as more comprehensive linguistic knowledge of Assamese becomes available.

[4] https://www.xobdo.org/
[5] A small number of morphological forms are also included.
[6] The lexicon contains 44,730 entries in total.

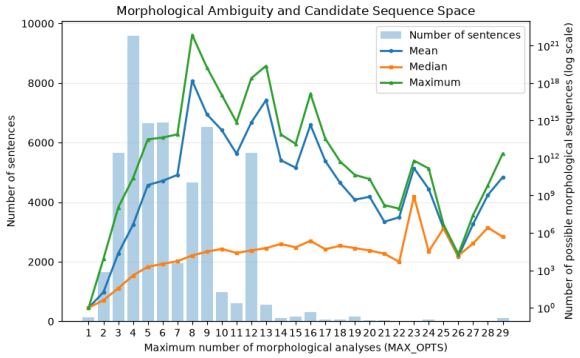


Figure 6: The bar-chart compares the number of sentences vs. maximum analyses of any constituent word. The line graphs compare maximum analyses vs. number of maximum (green), median (yellow) and mean (blue) candidate sequences.

4.3 Analysis Generation

The *rule-based forward analysis* algorithm (Algorithm 3 in Appendix D) produces all valid morphological analyses $A_w = a_1, a_2, \ldots$ of a word w. Each analysis is obtained through a morphological segmentation path over the surface form. An analysis a is represented as a sequence of tuples $(M_1, T_1), (M_2, T_2), \ldots$, where M_j and T_j denote the j-th morpheme and its corresponding tag, respectively. During traversal, only transitions permitted by $\mathcal{R}$ are explored, thereby pruning structurally invalid morpheme sequences and restricting the search space to linguistically valid analyses. The algorithm also accounts for PoS-tag transitions resulting from derivation and for integrated case markers by adding a (ϵ, T_d) tuple to the sequence, where ϵ is the empty string and T_d denotes either a derived PoS tag or an integrated case tag. We assume that a valid Assamese word can undergo at most one derivation. Consequently, analyses containing more than one derivation are discarded from the candidate set.

5 Annotation Process

We obtained the sentences for annotation from the *SANGRAHA* corpus (Khan et al., 2024), a large collection of Assamese text from the internet. Non-Assamese sentences were removed, and Unicode errors were rectified (Saikia et al., 2025). Using the available morphological lexicon and rules, our rule-based analyzer could analyze 81,757 of 153,633 unique word forms (53.21%). Sentences containing at least one non-analyzable word were discarded. We compared the ambiguity profiles of the remaining sentences and sampled 1,091 of them, restricting the sample to sentences whose maximum number of valid morphological analyses per word is 12. We sampled in proportion to the number of sentences at each ambiguity level (see Figure 6). We conduct this data filtration process to evaluate the proposed disambiguation framework under controlled linguistic conditions and to reduce the influence of non-analyzable and ambiguous words.

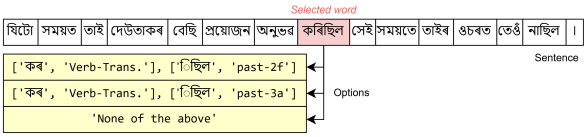


Figure 7: A sample from the annotation set.

On the interface, the possible morphological analyses $A_w = a_1, a_2, \ldots$ for each word w in a sentence are displayed, along with an additional 'None of the above' option (see Figure 7). The task is to select the most appropriate analysis based on the sentence context, as determined by the annotators. If none of the displayed analyses is considered correct, the sample is marked for manual analysis by selecting the 'None of the above' option.

Of the words selected for manual analysis, some were found to result from insufficient coverage of the morphological lexicon, while others provided insights for updating the bigram transition rules. Some words were added to the lexicon, and a few rules were updated, after which the remaining samples were annotated again using the updated resources[7]. The 19 sentences that remained unresolved were dropped.

6 Neural Disambiguation

Multiple studies (Shen et al., 2016; Dasari et al., 2023; Yildiz et al., 2016) have emphasized the role of context in the morphological disambiguation of agglutinative languages. This motivates us to use contextual information to disambiguate among the valid analyses of Assamese words. We formulate the task as analyzing the complete sentence at once and assigning it a *sentence-level grammar score*. Unlike previous methods (Sudhakar and Singh, 2017; Yildiz et al., 2016), our method scores and ranks candidate analysis sequences, thereby jointly evaluating the analyses of all constituent words within the sentence.

[7]All annotations were updated to reflect the modifications, and only a few dozen samples required re-annotation.

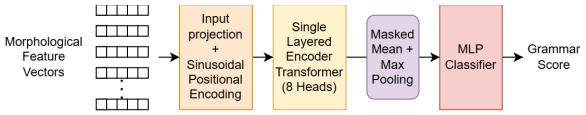


Figure 8: Transformer encoder for discriminating between correct and incorrect sequence of analyses.

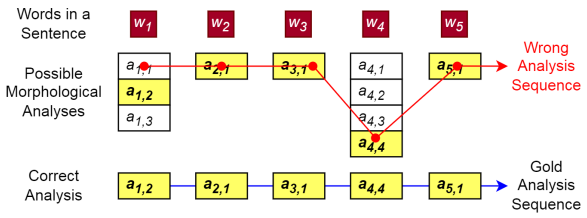


Figure 9: Example of negative sample or wrong morphological analysis sequence, and positive sample or gold analysis sequence.

6.1 Morphological Vector

In contrast to approaches that represent morphology using character or morpheme embeddings (Sudhakar and Singh, 2017; Dasari et al., 2023), we use an explicitly constructed morphological feature vector. This design avoids relying on large amounts of training data to learn distributed representations of morphological units, which is particularly relevant in the low-resource setting of Assamese. We also do not incorporate semantic representations, as existing pretrained models typically use subword tokenization. Integrating such representations with our morphology-based representations would require an additional mechanism for combining information across multiple subword tokens. We leave the integration of semantic representations for future work.

We represent the morphology of a word using a custom 64-dimensional feature vector that captures the PoS tags associated with the word, the presence of derivation, and prefix and suffix information. For each valid morphological analysis, a linguistically informed vector constructor (Algorithm 2, Appendix D) produces a corresponding morphological feature vector.

6.2 Transformer Encoder

We use a custom Transformer encoder block due to its ability to model contextual dependencies across sequence classification tasks (Vaswani et al., 2017). A sequence of morphological feature vectors serves as the input to the encoder, as illustrated in Figure 8. The input is projected onto a 32-dimensional space, after which sinusoidal positional encodings are added. The resulting contextualized representations are aggregated using masked mean and max pooling, and the concatenated pooled representation is passed through a classification layer to produce the *sentence-level grammar score*.

6.3 Training Objective

As the rule-based generator can produce multiple analyses options for a word, a sentence can be represented by multiple sequences of morphological features and corresponding morphological vectors (see Figure 9). However, only one of these candidate sequences is correct. We therefore formulate the training objective as binary sequence classification, where the model assigns a scalar *grammar score* to each input sequence. The model is optimized using Binary Cross Entropy (BCE) loss. At inference time, the model is not used as a conventional binary classifier; instead, the *grammar scores* assigned to all candidate sequences are compared, and the sequence with the highest score is selected as the predicted analysis[8].

7 Experimental Setup

The hybrid pipeline has been depicted in Figure 5 containing the methodological process of annotation and updation of the rule-based component.

7.1 Training Configuration

The dataset contains index of the correct analysis for each word of the sentences. This means that there exist a *gold* sequence of morphological vectors for a sentence, while others are *wrong* (see Figure 9). The tailored dataset statistics is provided in the Table 1. For each sentence we select up to K_n wrong samples by swapping a correct analysis from the gold sequence. We also consider augmenting the dataset with upto K_a random-offset padding for each sample. If $K_n = 2$ and $K_a = 3$, then there are a total of 3 positive (gold) and 18 negative[9] (wrong) samples generated from one sentence. More negative samples are considered to capitalize on the large candidate sequence space.

Sentence-level train-test splitting is performed using unique sentence identifiers to prevent aug-

[8] As the number of candidate sequences can exceed the available memory, we perform the ranking in chunks. The candidate sequences are divided into manageable batches, and the model assigns a *grammar score* to each sequence in a batch. The highest-scoring candidate is retained and, at the end, compared with the gold candidate.

[9] $3 \times 2 = 6$ negative samples, each augmented with 3 random-offset padded samples.

Category	Value
Total tags	62
Total morphological tags	64
Total sentences	1072
Total word tokens	11821
Unique tokens	4111
Ambiguous tokens	6187
Unambiguous tokens	5634
Average candidate analyses	2.0
Maximum candidate analyses	12
Average sentence length	11.03 tokens
Longest sentence length	36 tokens
Shortest sentence length	4 tokens
Training set size	859 sentences
Training set ambiguous words	4893
Test set size	213 sentences
Test set ambiguous words	1294
Median Candidate Sequences	128
Maximum Candidate Sequences	9.6746e+10

Table 1: Summary statistics of the annotated corpus used for morphological analysis experiments. Contribution from rule-based analyzer is in **bold**.

mented variants of the same sentence from appearing in both partitions. Experiments are conducted using different configurations of negative sampling $K_n \in \{1, 7, 14, 21\}$, and augmentation parameter $K_a \in \{1, 7, 14, 21\}$, to analyze their effect on disambiguation performance. The details of the hyperparameter are available in Table 3 of Appendix C.

7.2 Evaluation Metrics

The model is evaluated using both sequence-level and token-level metrics. Sequence-level performance is measured using ranking accuracy, which evaluates whether the gold morphological sequence receives a higher score than all corresponding negative candidate sequences for a sentence.

Let S_g denote the gold morphological sequence for a sentence, and let $\{S'_1, S'_2, \ldots, S'_n\}$ denote the corresponding negative candidate sequences. If $f(\cdot)$ represents the scoring function learned by the model, then a prediction is considered correct when $f(S_g) > f(S'_i), \quad \forall i \in \{1, 2, \ldots, n\}$. The sequence-level ranking accuracy is computed as:

$$\text{Rank Acc.} = \frac{1}{N} \sum_{k=1}^{N} \mathbb{I}\left[f(S_g^{(k)}) > \max_i f(S_i'^{(k)}) \right]$$

where N denotes the total number of evaluation sentences and $\mathbb{I}(\cdot)$ is the indicator function.

Additionally, word-level disambiguation accuracy is computed by selecting, for each token, the candidate analysis associated with the highest-scoring sentence sequence for the i-th sentence.

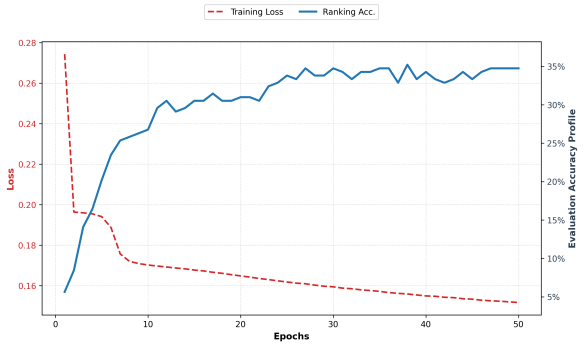


Figure 10: Comparison of model loss and ranking accuracy during model training with $K_n = 21$ and $K_a = 21$.

Let $a_{i,j}$ denote the gold analysis[10] of the j-th token in the i-th sentence, and $\hat{a}_{i,j}$ denote the predicted analysis of that same token within the top-scoring sequence. The token-level accuracy across a corpus of N sentences is defined as:

$$\text{Word Acc.} = \frac{1}{\sum_{i=1}^{N} L_i} \sum_{i=1}^{N} \sum_{j=1}^{L_i} \mathbb{I}(\hat{a}_{i,j} = a_{i,j})$$

We evaluate this metric across two distinct granularities depending on the selection of L_i:

- **Ambiguous Word Accuracy:** L_i denotes the number of ambiguous tokens within the i-th sentence. This captures the effectiveness of the ranking framework in resolving morphological ambiguity at the token level.
- **Overall Word Accuracy:** L_i denotes the total number of all non-padded tokens within the i-th sentence. This metric incorporates the deterministic performance of the rule-based components alongside the neural ranking framework, capturing the end-to-end pipeline accuracy.

The ranking accuracy at each epoch of the training process is only an approximation, as only a limited number of samples are ranked based on the values of K_n and K_a.

8 Results and Discussion

8.1 Quantitative Results

As the model improves on the discriminative sequence classification objective, its ability to rank

[10]Different morphological analyses may generate the same morphological vector. We consider all such analyses as equivalent and perform the computation using a single representative vector.

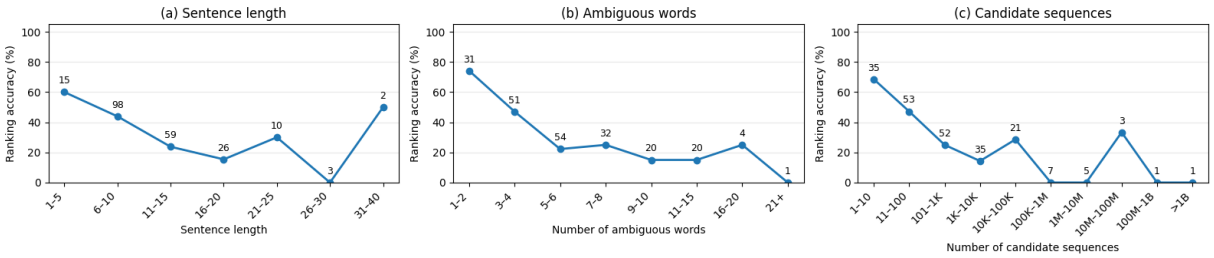


Figure 11: Ranking accuracy of the best performing model with respect to (A) sentence length, (B) number of ambiguous words, and (C) number of candidate sequences. The numbers above each point denotes the number of sentences in the partition.

K_a	K_n	Rank Acc. (%)	Corr. (x/213)	Ambig. Acc. (%)	Resolved (x/1294)	Overall Acc. (%)
1	1	8.45	18	47.84	619	72.08
1	7	16.90	36	62.83	813	80.09
1	14	13.62	29	53.32	690	75.01
1	21	16.90	36	57.11	739	77.03
7	1	22.54	48	74.19	960	86.16
7	7	25.82	55	75.81	981	87.03
7	14	29.11	62	77.13	998	87.73
7	21	33.33	71	78.21	1012	88.31
14	1	25.35	54	74.03	958	86.08
14	7	27.23	58	77.05	997	87.69
14	14	32.39	69	79.13	1024	88.81
14	21	32.86	70	79.21	1025	88.85
21	1	23.00	49	73.03	945	85.54
21	7	29.11	62	76.97	996	87.65
21	14	31.46	67	78.52	1016	88.48
21	**21**	**34.74**	**74**	**80.76**	**1045**	**89.67**

Table 2: Effect of negative sampling and random-offset augmentation on system performance.

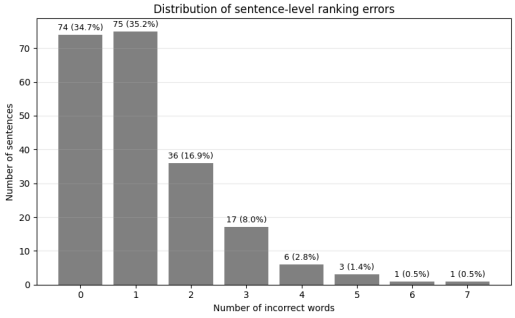


Figure 12: Number of sentences with different degrees of misclassification for the best performing model.

correct morphological sequences above incorrect alternatives also improves. The model achieves the best disambiguation accuracy of 80.76% by disambiguation 1045 out of 1294 ambiguous words (see Figure 10). Consequently, improvements in ranking accuracy lead to corresponding improvements in token-level morphological disambiguation accuracy. The results indicate that sentence-level contextual modeling substantially resolves morphological ambiguity, particularly in cases involving multi-functional morphemes and structurally similar candidate analyses.

The effect of augmentation and negative sampling on the model's performance on the test set has been illustrated in Table 2. Performance seems to improve with increasing values of K_n and K_a. We observed that the model had slight deviations in accuracy on different runs for a particular value of K_n and K_a. So, we only report the best of 3 runs for each configuration. These variations are likely caused by augmented samples created through randomized offsets.

8.2 Error Analysis

The model achieves good accuracy on ambiguity resolution, but fails to rank the correct sequence 65.26% of the time. We investigated the ranking failures and observed that the ranking accuracy is better when the sentences are short, and there is fewer ambiguity. These observations are reported in the Figure 11. In 53.95% of the incorrect rankings there is a single misclassification (see Figure 12). Example of some misclassifications are provided in Figure 13.

Although the task is morphological analysis, ambiguity of root-words introduce an additional challenge of jointly identifying the correct PoS tag. We observe several instances in which the model predicts the correct morphological information except for the PoS tag.

8.3 Discussion

This work does not aim to establish a linguistic theory of Assamese morphology. Rather, it develops a framework for automatic morphological analysis based on the assumption that Assamese exhibits systematic contextual regularities governing valid morphological sequences. The experimental results provide preliminary evidence that such regularities can be captured from morphological and structural



Figure 13: Example of errors in disambiguation. Error in classification of PoS tag is observed in several cases.

representations. Three observations support this interpretation:

- The neural sequence model does not receive lexical-semantic information and instead operates on morphological and structural representations. Nevertheless, it learns to distinguish valid from invalid morphological sequences, as reflected in the improvement in ranking accuracy.
- Performance improves with an increasing number of augmented samples. This suggests that exposure to varied word orders helps the model generalize across positional variations in context, which is compatible with the relatively flexible word-order properties of Assamese.
- Performance also improves with an increasing number of negative samples, suggesting that exposure to incorrect morphological sequences helps the model learn a more discriminative ranking function.

If supported by further investigation, these findings could motivate data-efficient approaches to morphological analysis for low-resource agglutinative languages, where obtaining supervised annotations is difficult. They may also motivate investigation into whether some contextual morphological regularities in Assamese can be captured using deterministic linguistic rules.

Our work also provides a methodology for scaling context-based morphological datasets. The modular design permits further experimentation through upgrades to components such as the morphological rules, lexicon, and vector construction. The framework can potentially be adapted to other languages with similar morphological characteristics. For Assamese, the resulting dataset and rules can facilitate context-aware morphological analysis.

Our analysis shows that disambiguation is necessary for context-aware morphological analysis in Assamese, given the presence of multifunctional suffixes and relatively permissive concatenation patterns (Gogoi, 2020). In Assamese, morphological information encodes grammatical properties such as case, number, person, and tense (Bhattacharjya, 2012; Das et al., 2019), making morphological analysis potentially useful for downstream tasks such as dependency parsing, semantic parsing, information extraction, summarization, machine translation, information retrieval, and text generation (Akshar et al., 1995).

Acknowledgements

This work was supported by the University Grants Commission (UGC) through the Junior Research Fellowship (JRF) awarded to the first author.

Limitations

Since the analyzer relies on a predefined morpheme inventory and rules, out-of-vocabulary (OOV) and foreign words are not handled by the rule-based component. The data filtering process also limits the generalizability of the evaluation to less constrained text. The bigram transition rule-set currently captures only the observed and linguistically validated transitions. Some missing transitions were identified during dataset annotation, indicating scope for further expansion. Additionally, to reduce ambiguity, derivational transformations were restricted to the most probable cases.

During annotation, multiple candidate analyses were occasionally found to be linguistically plausible for the same token (see Figure 14), highlighting the need for linguistic expertise in resource creation and annotation. Performance on a small test set may be sensitive to the specific samples included and may not fully reflect the model's generalization to unseen text. A larger and more diverse dataset would provide a more robust evaluation and allow the effectiveness of the proposed approach to be assessed across a wider range of morphological and contextual variations.

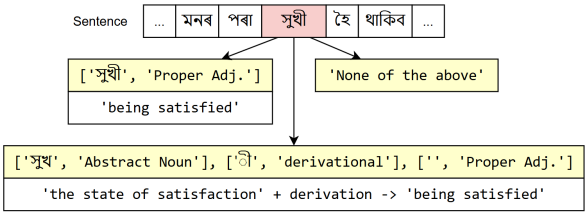


Figure 14: Example of multiple plausible analyses.

The current framework does not explicitly model semantic information, although semantic interpretation can influence morphological disambiguation (Sudhakar and Singh, 2017). PoS tagging has been jointly modeled based on purely the structural context, but it is a highly semantic process. Linguistic attributes, such as animacy, gender, etc. can only be modeled using semantics.

The number of valid morphological sequences can grow rapidly for morphologically complex sentences and OOV words. Although chunk-based ranking was used to handle large candidate sets efficiently, further constraints on candidate generation and heuristic methods may improve scalability.

References

Bharati Akshar, Vineet Chaitanya, et al. 1995. *Natural language processing: a Paninian perspective*. PHI Learning Pvt. Ltd.

Manmee Bhattacharjya. 2012. Case marking in Assamese: Morpho-Syntactic and Semantic Analysis. In Abhijit Debnath, Tariq Khan, Amalesh Gope, and Bindu Madhavi, editors, *Coalescence: A Post Conference Publication cum Students' Resource Book*, pages 195–204. Mudranik Technologies Pvt. Ltd., Bangalore.

Palash Das, Madhumita Barbora, and Utpal Sharma. 2019. Morphological rule formulation for nouns in assamese to develop morphological tools. *Language in India*, 19.

Priyanka Dasari, Abhijith Chelpuri, Nagaraju Vuppala, Mounika Marreddy, Parameshwari Krishnamurthy, and Radhika Mamidi. 2023. Transformer-based context aware morphological analyzer for Telugu. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*, pages 25–32, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Pinky Moni Gayan and Arup Kumar Nath. 2025. Productivity of assamese derivational suffixes. *Language Sciences*, 112:101746.

Jugalakshmi Gogoi. 2020. Word formation in Assamese. Master's thesis, Assam University, Silchar, India. Department of Linguistics.

Gulshat Kessikbayeva and Ilyas Cicekli. 2014. Rule based morphological analyzer of Kazakh language. In *Proceedings of the 2014 Joint Meeting of SIGMORPHON and SIGFSM*, pages 46–54, Baltimore, Maryland. Association for Computational Linguistics.

Mohammed Safi Ur Rahman Khan, Priyam Mehta, Ananth Sankar, Umashankar Kumaravelan, Sumanth Doddapaneni, Suriyaprasaad B, Varun G, Sparsh Jain, Anoop Kunchukuttan, Pratyush Kumar, Raj Dabre, and Mitesh M. Khapra. 2024. IndicLLMSuite: A blueprint for creating pre-training and fine-tuning datasets for Indian languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15831–15879, Bangkok, Thailand. Association for Computational Linguistics.

Bipul Roy. 2017. A suffix based morphological analysis of assamese word formation. *International Journal on Recent and Innovation Trends in Computing and Communication*.

Navanath Saharia. 2014. *Computational Morphology and Syntax for a Resource-Poor Inflectional Language*. Ph.D. thesis, Tezpur University, Napaam, Tezpur, Assam, India. Accessed via Shodhganga: A reservoir of Indian theses.

Pragyanur Saikia, Arindam Karmakar, and Utpal Sharma. 2025. Issues and challenges in training vector representation for assamese words. In *2025 IEEE Guwahati Subsection Conference (GCON)*, pages 1–6.

Wolfgang Seeker and Jonas Kuhn. 2013. Morphological and syntactic case in statistical dependency parsing. *Computational Linguistics*, 39(1):23–55.

Utpal Sharma, Jugal K. Kalita, and Rajib K. Das. 2008. Acquisition of morphology of an indic language from text corpus. *ACM Transactions on Asian Language Information Processing*, 7(3).

Qinlan Shen, Daniel Clothiaux, Emily Tagtow, Patrick Littell, and Chris Dyer. 2016. The role of context in neural morphological disambiguation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 181–191, Osaka, Japan. The COLING 2016 Organizing Committee.

Akhilesh Sudhakar and Anil Kumar Singh. 2017. Neural morphological disambiguation using surface and contextual morphological awareness. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, pages 485–494, Kolkata, India. NLP Association of India.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Eray Yildiz, Caglar Tirkaz, H. Sahin, Mustafa Eren, and Omer Sonmez. 2016. A morphology-aware network for morphological disambiguation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

A Tags and Features: Details

A.1 List of Morpheme Tags

Following is the comprehensive list of all the morpheme tags used in the system. Each entry in the morphological lexicon can belong to one or more of these categories. The name of these tags are kept consistent with the original *XOBDO* lexicon. Special symbols, word-boundary tags and numeric-strings do not belong to this dictionary. The elaborate meaning of these tags can be found in the online repository.

PoS tags: ‘Abstract Noun’, ‘Common Noun’, ‘Material Noun’, ‘Verbal Noun’, ‘Noun’, ‘Proper Noun’, ‘Pronoun’, ‘Adverb’, ‘Verbable’, ‘neg_v_verb’, ‘Verb-Trans.’, ‘Verb-Intran.’, ‘Adjective’, ‘Adjective Adj.’, ‘Proper Adj.’, ‘Verbal Adj.’, ‘Conjuction’, ‘Interjection’, ‘Adposition’, ‘Others’

Suffix tags: ‘derivational’, ‘abs’, ‘erg’, ‘gen’, ‘acc’, ‘dat’, ‘loc’, ‘inst’, ‘number’, ‘modal’, ‘adj_comp’, ‘after_comp’, ‘reason_comp’, ‘when_comp’, ‘ardha’, ‘infix’, ‘dash’, ‘manner’, ‘mod_attr’, ‘quantity’, ‘compare’, ‘emph’, ‘t_loc’

Verb-tense tags: ‘present-2i’, ‘present-3a’, ‘present-2n’, ‘present-2f’, ‘present-1’, ‘past-3a’, ‘past-2f’, ‘past-2a’, ‘past-1’, ‘future-2f’, ‘future-2n’, ‘future-3a’, ‘future-1’

Preffix tags: ‘without’, ‘negative’, ‘numerical’, ‘other’

Word-boundary: ‘ws’, ‘we’

A.2 Feature Vector Details

The features contain super-set PoS tags (e.g. Noun is a super-set for {Abstract Noun, Common Noun, Proper Noun}.) For any PoS tag, the weight of its corresponding super-set is also increased. The symbols are further categorized into – ‘end_sym’, ‘pause_sym’ and ‘other_sym’. String of digits are separated from words during preprocessing, and marked using ‘numeric-string’. Honorifics are marked using – ‘a’ for ‘any’, ‘i’ for ‘informal’, ‘n’ for neutral, and ‘f’ for formal. These can be extracted from the verb-tense tags and integrated pronouns. ‘p1’, ‘p2’, and ‘p3’ are used to mark person.

Morphological vector features: {‘Abstract Noun’, ‘Adjective’, ‘Adjective Adj.’, ‘Adposition’, ‘Adverb’, ‘Common Noun’, ‘Conjuction’, ‘Conjunction’, ‘Interjection’, ‘Material Noun’, ‘Noun’, ‘Others’, ‘Pronoun’, ‘Proper Adj.’, ‘Proper Noun’, ‘Verb’, ‘Verb-Intran.’, ‘Verb-Trans.’, ‘Verbable’, ‘Verbal Adj.’, ‘Verbal Noun’, ‘a’, ‘abs’, ‘acc’, ‘adj_comp’, ‘after_comp’, ‘ardha’, ‘compare’, ‘dash’, ‘dat’, ‘derivational’, ‘emph’, ‘end_sym’, ‘f’, ‘future’, ‘gen’, ‘i’, ‘infix’, ‘inst’, ‘loc’, ‘manner’, ‘mod_attr’, ‘modal’, ‘n’, ‘neg_v_verb’, ‘negative’, ‘erg’, ‘number’, ‘numeric-string’, ‘numerical’, ‘other_sym’, ‘p1’, ‘p2’, ‘p3’, ‘past’, ‘pause_sym’, ‘pl’, ‘present’, ‘quantity’, ‘reason_comp’, ‘sg’, ‘t_loc’, ‘when_comp’, ‘without’}

B Incorporation of Additional Morphological Characteristics

We incorporated the major characteristics of the language and avoided the exceptions.

B.1 Morphophonological Alternation

Morphophonological alternation produces multiple surface realization of morphemes during affixation. To incorporate some of these characteristics we have categorized all the surface forms of the affixes into the same category of their role. However, we only address a few forms of roots that undergo morphophonological alternation. These are – 1) the affixation phenomenon in verbs and 2) number-words (first 9 natural numbers) as numerical prefixes. With verbs, prefixation by 'negation' affix and some kind of suffixation cause alterations. Due to this we created two categories:

1. **'neg_v_verb':** These are verbs whose starting *vowel-character* is removed to facilitate affixation using the 'negation' prefix.

2. **'Verbable':** These are verbs whose ending *vowel-symbol/character* is removed. This facilitates the construction of 'causative stems' (Gogoi, 2020) that attracts verb-like suffixation.

Similarly, to accommodate another major phenomenon, the vowel ending from the number-words are subtracted to create a class of numeric prefixes.

B.2 Integrated Tags

The suffixes under the tag 'number', also known as 'definitives' in Assamese (Sharma et al., 2008; Das et al., 2019), can either be singular or plural. We have used a mapping for all such suffixes which is used by the *feature vector constructor* (algorithm 2). Similarly, we maintain a dictionary for the tags associated with all pronouns present in *XOBDO* lexicon. We also maintain a list for the verb-suffixes that convey tense, person and honorific information. E.g. 'present-1', 'past-2f', etc. To extract the integrated features we use regular expressions.

C Experiment Setup: Details

All the resources were preprocessed using the methods mentioned by Saikia et al. (2025). Details of hyperparameters are provided in Table 3. Hardware and neural model details is provided in Table 4.

Hyperparameter	Value(s)
Context window size	64
Feature dimension	64
Transformer layers	1
Attention heads	8
Feed-forward dimension	64
Classifier hidden dimension	32
Dropout	0.2
Training epochs	50
Batch size	1024
Learning Rate	1×10^{-4}
Random state	0

Table 3: Values for neural model hyperparameter.

Component	Value
Framework	PyTorch
Programming Language	Python 3.13
CUDA Used	Yes
Precision	float32
Optimizer	AdamW
Loss Function	Binary Cross Entropy Loss
Inference Strategy	Highest Scoring Candidate
Total Parameters	14,785
Approximate Size	0.0564 MB

Table 4: Other details regarding experimental setup.

As the candidate space grows exponentially, we use annotated-data statistics to estimate tag-transition probabilities and rank candidate sequences by the summed log probabilities of their analyses. We retain the top 250,000 candidates for evaluation, with the gold sequence always preserved above the cutoff.

D Algorithms

Algorithm 1 OPTIONSOFSENTENCETOKENS

Input: Sentence s, special character set $\mathcal{C}$
Output: Sequence of token-level morphological option sets $\mathcal{O}$

Initialize $\mathcal{O} \leftarrow [\,]$
foreach *token* w *in* s **do**
  $opts \leftarrow$ RULEBASEDFORWARDANALYSIS(w)
  if $opts \neq \emptyset$ **then**
    Append $opts$ to $\mathcal{O}$
  else if $w \in \mathcal{C}$ **then**
    Append $[[[w,$ `'symbol'`$]]]$ to $\mathcal{O}$
  else if w *matches numeric pattern* **then**
    Append $[[[w,$ `'numeric-string'`$]]]$ to $\mathcal{O}$
  else
    Append $[[[w,$ `'Others'`$]]]$ to $\mathcal{O}$
return $\mathcal{O}$

Algorithm 2 FEATUREVECTORCONSTRUCTOR

Input: Sequence of (morpheme, tag) tuples $(m, t) \in \mathcal{S}$, detail map of number tag $\mathcal{N}_{detail}$, integrated tags in pronouns, integrated tags in verb suffixes $\mathcal{V}_{suffixes}$, super set mapping of PoS $\mathcal{P}_{map}$, initial feature state $\mathcal{F} \leftarrow \mathcal{F}_{default}$, and tracking counter $pos_num \leftarrow 0$, $pos_decay \leftarrow 0.25$.

Output: Populated morphological feature vector $vector$

foreach $(m, t) \in \mathcal{S}$ `(right-to-left)` **do**
- **if** $t =$ `'number'` **then**
  - $\mathcal{F}[\mathcal{N}_{detail}[m]] \leftarrow \mathcal{F}[\mathcal{N}_{detail}[m]] + 1$
  - $\mathcal{F}[t] \leftarrow 1$
- **else if** $t =$ `'Pronoun'` **then**
  - $\mathcal{F}[t] \leftarrow \mathcal{F}[t] + 1$
  - $pos_num \leftarrow pos_num + 1$
  - $\mathcal{T} \leftarrow$ person, honorific tags of m
  - **foreach** *valid* $f \in \mathcal{T}$ **do**
    - $\mathcal{F}[f] \leftarrow \mathcal{F}[f] + 1$
- **else if** $t =$ `'symbol'` **then**
  - $\mathcal{F}[\mathcal{S}_{map}[m]] \leftarrow \mathcal{F}[\mathcal{S}_{map}[m]] + 1$
- **else if** $t =$ `'derivational'` **then**
  - $\mathcal{F}[t] \leftarrow 1$
- **else if** $t \in \mathcal{P}_{map}$ **then**
  - $wt \leftarrow 1 - pos_decay \times pos_num$
  - **if** $\mathcal{F}[t] = 0$ **then**
    - $\mathcal{F}[t] \leftarrow wt$
    - $pos_num \leftarrow pos_num + 1$
  - **if** $\mathcal{F}[\mathcal{P}_{map}[t]] = 0$ **then**
    - $\mathcal{F}[\mathcal{P}_{map}[t]] \leftarrow wt$
    - $pos_num \leftarrow pos_num + 1$
- **else if** $t \in \mathcal{V}_{suffixes}$ **then**
  - $\mathcal{T} \leftarrow$ person, honorific, tense tags of m
  - **foreach** *valid* $f \in \mathcal{T}$ **do**
    - $\mathcal{F}[f] \leftarrow \mathcal{F}[f] + 1$
- **else**
  - $\mathcal{F}[t] \leftarrow \mathcal{F}[t] + 1$

$vector \leftarrow$ values of $\mathcal{F}$ sorted alphabetically by key

return $vector$

Algorithm 3 RULEBASEDFORWARDANALYSIS

Input: Input word w, morphological lexicon $\mathcal{M}$, morphotactic transition matrix $\mathcal{R}$

Output: Set of valid analyses $\mathcal{A}$

Initialize $\mathcal{A} \leftarrow \emptyset$

`Analyze(`r, t, s`)(`
- **if** $r = \epsilon$ **then**
  - **if** $\mathcal{R}(t,$ `'we'`$) > 0$ **then**
    - Find number of derivations nod in s
    - **if** $nod \leq 1$ **then**
      - Add s to $\mathcal{A}$
  - **return**
- $\mathcal{T} = \{t' \mid \mathcal{R}(t, t') > 0 \wedge t' \neq$ `'we'`$\}$
- **for** $i \leftarrow 1$ **to** $|r|$ **do**
  - $m \leftarrow r[1 : i]$
  - $u \leftarrow r[i + 1 : |r|]$
  - **foreach** $t' \in \mathcal{T}$ **do**
    - **if** $m \in \mathcal{M}(t')$ **then**
      - $s' \leftarrow s + [(m, t')]$
      - **if** $t' =$ `'derivational'` **then**
        - Determine derived category d
        - $s' \leftarrow s' + [(\epsilon, d)]$
        - `Analyze(`u, d, s'`)`
      - **else if** $t' =$ `'Pronoun'` **then**
        - Determine integrated case p
        - **if** $p \neq$ `'Pronoun'` **then**
          - $s' \leftarrow s' + [(\epsilon, p)]$
          - `Analyze(`u, p, s'`)`
        - **else**
          - `Analyze(`u, t', s'`)`
      - **else**
        - `Analyze(`u, t', s'`)`

)

`Analyze(`w, `'ws'`, $\emptyset$`)`

return $\mathcal{A}$