On the usage of semantics, syntax, and morphology for noun classification in isiZulu

Imaan Sayed, Zola Mahlaza, Alexander van der Leek, Jonathan Mopp, C. Maria Keet
Department of Computer Science
University of Cape Town
South Africa
{zmahlaza,mkeet}@cs.uct.ac.za
{SYDIMA002,VLKALE003,MPPJON002}@myuct.ac.za

Abstract

There is limited work aimed at solving the core task of noun classification for Nguni languages. The task focuses on identifying the semantic categorisation of each noun and plays a crucial role in the ability to form semantically and morphologically valid sentences. The work by Byamugisha (2022) was the first to tackle the problem for a related, but non-Nguni, language. While there have been efforts to replicate it for a Nguni language, there has been no effort focused on comparing the technique used in the original work vs. contemporary neural methods or a number of traditional machine learning classification techniques that do not rely on human-guided knowledge to the same extent. We reproduce Byamugisha (2022)'s work with different configurations to account for differences in access to datasets and resources, compare the approach with a pretrained transformer-based model, and traditional machine learning models that rely on less human-guided knowledge. The newly created data-driven models outperform the knowledge-infused models, with the best performing models achieving an F1 score of 0.97.

1 Introduction

Solid performance when using modern Natural Language Processing (NLP) approaches, especially ones that are popular with languages like English, is dependent on the availability of large text corpora. Unlike English, all Niger-Congo B[1] (NCB) languages do not have large training datasets; hence, contemporary techniques have not been used for tasks such as noun classification. Since the languages are characterized by agglutinative morphology, have an intricate noun class system, and possess little datasets and tools that can be repurposed for various tasks (Moors et al., 2018), most problems have been tackled with knowledge-infused approaches. The discrepancy of resource availability also means that there are limited efforts to contrast contemporary data-driven and knowledge-infused techniques to determine whether there is any difference in performance.

In this paper, we address this lack of comparison for the task of noun class disambiguation for NCB languages. Using isiZulu, the focus of this paper and the largest language in South Africa by L1 speakers, as a case study the task consists of predicting the noun class (e.g., NC2) when one is given a noun (e.g., *abantu* 'people'). We limit our investigation to this task since it is a crucial but unsolved problem for all NCB languages, isiZulu especially. Due to NCB languages' low-resourced state, the only work that tackles the task for a NCB language was done by Byamugisha (2022) focusing on Runyankore and related languages from Guthrie's Zone J (Maho, 1999).

Byamugisha (2022)'s work deals with the lack of a large dataset of noun and class pairs by introducing a number of modules, each solving some crucial function. Some modules use unlabelled or automatically labelled datasets that, when combined are able to predict the noun class of the noun. Byamugisha's work is a promising start, since it obtained accuracies in the range 80%-87% for Runyankore. The work does not resolve the question of technique comparison; hence, the utility of relying on a multi-modular and knowledge-infused approach that combines morphology, syntax, and morphology vs. machine learning techniques and a neural model, especially a large language model (LLM) adapted via the pretrain-

[1] Some authors use the term Bantu languages

Proceedings of the Third Workshop on Resources and Representations for Under-Resourced Languages and Domains (RESOURCEFUL 2025), pages 96–105

finetune paradigm for classification, is still unclear. All of the aforementioned models have the potential to classify nouns using morphology, syntax, and morphology but differ in the following way:

System complexity and resource requirements: Knowledge-infused approaches tend to increase the number of sub-modules, each with a clear and dedicated responsibility, hence the complexity of the system increases. While the dedicated functionality of the subcomponents makes the entire system more auditable, such techniques tend to rely on stopgap resources (e.g., models that are trained using automatically labelled datasets (e.g., (Mahlaza et al., 2025)) due to a lack of a context-free grammar that can be used to generate a gold standard dataset unlike Byamugisha (2022)) and they are sometimes inferior with respect to advanced pattern recognition vs. modern blackbox models (e.g., LLMs).

Reliance on morphosyntax: Knowledge-infused approaches have not been used to investigate noun classification while relying only on morphosyntax, in the context of NCB languages, due to the difficulty associated with the lack of clarity regarding effective representations, especially ones that separate semantics from syntax, morphology and other features (see (Huang et al., 2021) for similar challenges with English sentences).

Our approach is two-fold. First, we reproduce Byamugisha's knowledge-infused noun classifier for isiZulu, while noting the differences in resource requirements and their availability. The primary goal is to determine how best to build a syntactic-semantic model for isiZulu since there is no Context Free Grammar (CFG) that can be used to generate labelled and unlabelled datasets. This requires that we identify how changes in training corpora characteristics for the data-driven components affect accuracy. In that regard, we consider various options for labels in the labelled data (concord, noun class, or both), training corpus size, annotation quality (manually annotated by an expert or automatically labelled), and data-level (sentential, phrasal, or word-based).

Second, we create various supervised machine learning classifiers (k-Nearest Neighbours (kNN) algorithm, decision trees, Support Vector Machines (SVM)), and deep learning based models (a fully connected feed-forward neural network and a fine-tuned version of the Serengeti language model (Adebara et al., 2023)). We compare all the models using a larger dataset (cf. Byamugisha (2022)) made up of nouns and their classes to ascertain whether one can obtain similar or superior performance by relying on a traditional ML model that makes use of morphosyntax only. We also investigate whether similar, or superior, performance can be achieved via a neural model that relies on morphological, syntax, and semantic knowledge trained from scratch or adapted from a pre-trained multilingual LLM (in this case, Serengeti (Adebara et al., 2023)).

Our results showed that the neural-based and traditional ML models perform the best. The best multi-modular model that relies on human-guided knowledge achieves an F1 score of 0.71 while the best neural model and traditional ML models have scores of 0.97.

The rest of the paper is structured such that Sections 2-3 introduce noun classification and the existing models, Section 4 details the created dataset, Sections 5-7 introduce our models, Section 8 presents the results, Section 9 discusses, and Section 10 concludes.

2 NCB noun classification

NCB languages are found in more than 54 countries, with an estimated 240 million speakers, and they have a lot of diversity (Gowlett, 2014). Nonetheless, they all have a noun class system that categorises each noun to one of 23 classes, as informally summarized in Table 1 for isiZulu. To demonstrate the impact of the noun classes on the formation of sentences, consider the following example English sentence and its translation:

English: The dog is unhealthy
The$_{article}$ dog$_{subj.\ noun}$ is$_{singl.identifier}$ un$_{negation}$-healthy$_{adjective}$

IsiZulu: *Inja ayiphilile*
I$_{NC9}$-nja$_{stem}$ a$_{neg.preprefix}$-yi$_{NC9\ SC}$-philile$_{adjective\ root}$

The formation of the word *ayiphilile* 'is unhealthy' relies on identifying the noun class (here: NC9) of the subject *inja* 'dog'. However, there are no models for automatically classifying nouns into their respective classes for isiZulu.

Table 1: List of NCB, including isiZulu, noun classes and the semantics that govern inclusion for each class (Source: (Byamugisha, 2022))

Noun class	Example semantic categorization
1, 2	People and kinship
3, 4	Plants, nature, and some parts of the body
5, 6	Fruits, liquids, some parts of the body and paired things
7, 8	Inanimate objects
9, 10	Tools and animals
11	Long thin stringy objects, languages,and inanimate objects
12, 13	Diminutives
14	Abstract concepts
15	Infinitives and parts of the body
16, 17, 18	Locative classes
19	Diminutives
20, 21, 22	Augmentatives
23	Locative

3 Existing models for noun classification

The only work that has tackled the task at hand, for NCB languages at least, was conducted by Byamugisha and it took inspiration from the existing linguistic theory on the NCB noun class system by modelling the possible avenues for classifying a noun namely the morphological prefix, semantic categorization and syntactical context (Byamugisha, 2022). They pursue the task via a multimodular knowledge-infused model, whose function will now be described.

The simplest avenue relies on the prefix. Byamugisha's model uses the morphological prefix information to classify a noun if it is unique. For instance, the noun *abantu* 'people' has as prefix *aba-* and stem *-ntu* and the prefix *aba-* is unique to NC2, the noun will be correctly classified. The noun *umuntu* 'person' has as prefix *umu-*, but it is ambiguous, because the prefix associated with both NC1 and NC3 is either *um-* or *umu-* depending on the number of syllables of the stem. This simple model will only output a prediction if a unique prefix is found otherwise it is considered ambiguous and continues to the next step.

When the prefix is insufficient, it draws on the semantic generalizations to determine the noun class. This is done by training a new model to determine similar words, using FastText[2] with a corpus of 1 million sentences, to determine a noun's semantic neighbours. For instance, for the Runyankore noun *omuntu* 'person' from NC1, the model determines that the nearest neighbours are *omugyesi* 'reaper' (NC1), *omutaahi* 'companion' (NC1), *omukoreesa* 'overseer' (NC1), *omushomesa* 'teacher' (NC1), and *omukuru* 'elder' (NC1). The semantic information derived from the nearest neighbours allows discerning between ambiguous classes, since information associated with, e.g., *omuntu* 'person' (NC1) can be used to distinguish it from the noun *omukono* 'arm', based on the noun class frequencies associated with its neighbours. The noun *omukono* shares the same prefix *omu-*, but one retrieves different neighbours, such as *omunwa* 'mouth' (NC3), *omutwe* 'head' (NC3), *eriino* 'tooth' (NC5), and *enkokora* 'elbow' (NC9), i.e., body parts, vs. *omuntu* 'person' and certain roles they play. Fundamentally, the differentiation between the two is done by analysing the noun classes associated with the neighbouring words and ascertaining that NC1 is the most common class among the neighbours for *omuntu* 'person', hence, the input noun is inferred to belong to the same class.

The determination of the most common class among the neighbours requires filtering out some elements. Specifically, when given neighbouring nouns, without any labels, a corpus made up of 1 million sentences is used to train a FastText classifier, where the corpus' is annotated with parts-of-speech, the noun class, and the concord (where possible). The resulting model is used to annotate the input neighbouring words and if these predictions are found to be inconsistent then they are dropped from consideration. The concord annotation is then used for the syntax-based filtering step because it is unique among the classes (Gowlett, 2014; Maho, 1999).

Alternative work involving processing NCB nouns exists, but it does not tackle the problem of noun classification; specifically, the efforts on building morphological analysers (Bosch et al., 2008), morphological generators (Bosch and Pretorius, 2003), part-of-speech taggers (De Pauw et al., 2012), and noun pluralization tools (Byamugisha et al., 2018, 2017) show attempts to

[2] https://radimrehurek.com/gensim/models/fasttext.html

deal with the ambiguity of nouns. Other researchers have created a massively multilingual transformer-based encoder-only language model, named Serengeti, whose training data includes isiZulu (Adebara et al., 2023). However, none of these models have been investigated, despite their potential capability, to classify nouns or to compare them with Byamugisha (2022)'s approach.

4 New dataset for the experiments

The aim of the experiments is to ascertain and compare the performance of multiple methods, detailed in Sections 5-7. In this section, we describe the dataset that is used to compare the techniques.

We created a new isiZulu dataset by extracting nouns and their classes from the Oxford Zulu-English dictionary (de Schryver, Gilles-Maurice, 2015) via optical character recognition and manual cleaning. We created two versions of the dataset where one version is labelled with a single noun class, either singular or plural depending on the modality of the noun, and the second is labelled with the singular and plural classes. For instance, the word *umuntu* 'person' is labelled with the singular noun class 1 in one dataset and labelled with the singular and plural combined classes 1/2 in another. The dataset version that combines noun classes is only used to train some of the traditional machine learning models and the details are provided in Section 6.

The number of nouns per class in the dataset is listed in Table 2. We used an 80-20 train-test split.

Table 2: Distribution of nouns per class in the dataset used for training and testing models.

Class	**% of nouns**	**|nouns|**
1	4.80	110
1a	6.37	144
2	4.13	94
2a	2.11	48
3	7.02	160
4	3.82	87
5	12.99	296
6	10.14	231
7	10.05	229
8	7.33	167
9	13.08	299
10	6.80	155
11	4.30	98
14	2.63	60
15	4.43	101
Total		2279

5 Knowledge-infused models

We created multiple variations of Byamugisha (2022)'s multi-modular classifier to support isiZulu. This is done by creating multiple versions of each module in the architecture, labelled A-G in Figure 1. We now turn to describe the design decisions and resources used.

Component A This module identifies the noun class via the noun's prefix. We use Table 7 to determine if a noun has a unique prefix hence it is possible to uniquely determine its noun class. When the prefix is unique then we resolve the noun class while ensuring that we prioritise values that have the longest length. For instance, when a noun begins with the prefix *aba-* then it can be uniquely identified as belonging to NC2, however, a noun such as *umthandazo* 'prayer' can be classified to NC1 or NC3. When a noun's class is ambiguous then the noun is passed to the following modules.

Component B and C These modules take first responsibility in the pipeline to determine the class when a noun's prefix is not unique. They first embed words in a vector space as a means of identifying similar words. Words are embedded using two possible models; both versions are FastText skipgram models, motivated by our interpretation of the work done for Runyankore. One version is a pre-trained isiZulu model created using 1 million sentences sourced from Dlamini et al. (2021). It was trained with 300 dimensions, and subwords are formed using n-grams in the range of 3-6. The alternative model is trained on 180 000 unlabelled web-crawled sentences, whose sources are listed in Table 3. For each word representation, we identify K similar words using the traditional kNN algorithm, where K was selected from the range 10 to 200.

Component D This module takes each of the predicted neighbouring nouns, produced by modules B and C, and labels them with a noun class and/or a class-specific concord using a classifier. This annotation classifier is trained from scratch. Since we do not have access to a context-free grammar to generate training data à la Byamugisha (2022) for the classifier, we investigated

the use of different datasets to determine the impact of certain characteristics (e.g., annotation quality); all features are listed in Table 4. Seven classifier versions were developed, each with FastText's supervised training capability and its hyperparameter autotuning feature (Joulin et al., 2017). Training data was split in the ratio 80/20 for training and validation respectively. As an internal evaluation approach, the performance of each classifier is tested on the Keet dataset listed in Table 3.

Component E and F These modules are responsible for automatically filtering nearest-neighbouring nouns using either a part-of-speech classifier or regular expressions. Since module D did not annotate the words with a part-of-speech, these modules rely on a newly trained POS classifier for the annotations. The classifier was trained on web-crawled data with simplified POS tags and sourced from (du Toit and Puttkammer, 2021). The new classifier is able to identify verbs with 96% accuracy when tested against the combined gold standard datasets listed in Table 4. When the current modules use the trained classifier, they remove all neighbouring words that are identified as verbs. These modules also rely on an alternative filter that removes verbs by matching their subject concord using regular expressions based on the work by Keet and Khumalo (2017), along with additional rules from the Oxford isiZulu Bilingual Dictionary (de Schryver, Gilles-Maurice, 2015).

The second phase of filtering removes words that do not contain a morpheme associated with their predicted noun class. There are two alternative models considered to achieve this. The first version (i.e., subword-level) removes a neighbour if the morpheme associated with predicted label is not contained in the word, by matching all possible versions of it (including phonological conditioned variations) (Keet and Khumalo, 2017). The second model (i.e., word-level) filters neighbours based on their subwords. It fetches the character n-gram range for the word model, computes all substrings for the word that matches that length, labels each subword with a noun class and concord using the previously mentioned classifier and returns True if the neighbour's predicted label is in the set of predictions for its subwords.

Component G This module is responsible for identifying the noun class from the set of annotated words produced by the previous steps in the pipeline. It does so by computing the frequencies for each noun class found in the dataset and identifies the class with the highest count in the final list of nearest neighbours. The most common class is then used as the final prediction.

Table 3: List of datasets used to build the annotation classifiers required for the isiZulu knowledge-infused model.

Dataset	Size	Type	Label
Web-crawled data (Leipzig CC - isiZulu 2016 Mixed Corpus) (Leipzig University, 2024)	180 000	Sent.	✗
NCHLT Morph. Corpus (Gaustad and Puttkammer, 2022)	45 000	Word	✓
Ukwabelana (Spiegler et al., 2010)	21 416	Word	✓
Gaustad & McKellar (Gaustad and McKellar, 2024)	50 000	Word	✓
Keet (sourced from author and (Gilbert and Keet, 2018))	795	Word	✓

We compared the various versions of the knowledge-infused model by determining their accuracies on the Keet dataset, listed in Table 3. The evaluation results will be discussed in Section 8. For the final evaluation, we compute the precision, recall, and F1 scores using the test set detailed in Section 4 for the best performing models.

6 Traditional machine learning models

To create novel supervised ML models that rely on morphosyntax, and possibly syntax and semantics, for noun classification we considered four supervised machine learning algorithms and models. In addition, we also experimented with various ways of preprocessing and representing the nouns. We describe the choices made regarding these elements in the following subsections.

Noun forms We investigate the use of compressed and uncompressed versions of each noun

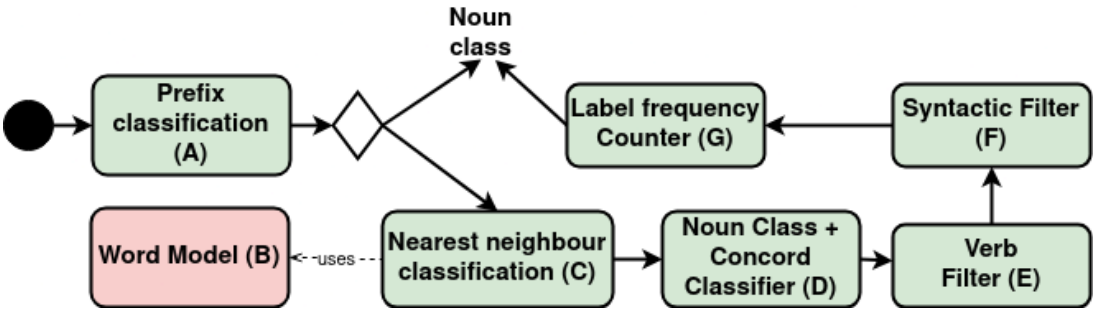

Figure 1: Architecture of the approximated knowledge-infused model for noun classification.

Table 4: Datasets used to train a word embedding model for the replicated classifier. Abbreviations: SN = Sentence, Part = Partial, B = Bronze, G = Gold, W = Word, SC = Subject concord, NC = Noun class, and PC = Possessive concord.

Dataset name	Label(s)	Size	Level
Automatically labelled datasets (Bronze)			
SN-B	SC, NC	103 895	SN
SN-B-PartSN	SC, NC	103 895	Phrase
SN-BW	SC, NC	336 029	Word
N-BW	NC	246 362	Word
Expert labelled datasets (Gold)			
Full-GW	SC, NC, OC, PC, Verb	61 954	Word
SN-GW	SC, NC	50 954	Word
N-GW	NC	36 713	Word

and this is done to address the hypothesis that the compressed form of the morphosyntactic model will outperform the surface-form variant, drawing from the existing literature surrounding the accuracy gains observed when compressing text in the context of topic classification (i.e., (Jiang et al., 2023)). Specifically, nouns are compressed using gzip with all the default parameters found in Jiang et al. (2023) but we use a single time parameter (i.e., 0) instead of relying on the current time.

Noun representations We convert each noun into a vector by relying on term frequencies, obtained via *scikit-learn*'s *TfidfVectorizer* and *TfVectorizer* with the same 'character' and 'lowercase=false' parameters to ensure that we only consider character level n-grams within the nouns and account for capitalization. When creating vector representations, we made use of term frequency (TF) and term frequency inverse document frequency (TF-IDF) to determine the impact of taking into account the rarity of an n-gram in the noun set (Shahmirzadi et al., 2019).

Models We investigated the use of a nearest neighbours classifier, decision tree, and a support vector machine, all created using *scikit-learn*[3]. The main hyper-parameter adjusted and tested for in the case of kNN was the number of neighbours considered, otherwise all defaults for the *scikit-learn*'s *KNNClassifier* class were used. For the decision tree, we adjusted the tree depth, in addition to assigning an integer to the random state parameter to achieve deterministic behaviour. We also combatted overfitting via cost complexity pruning; otherwise, all defaults for the *scikit-learn*'s *DecisionTreeClassifier* class were used. For the SVM, we used a linear kernel since it leads to faster training speed and tends to be less prone to overfitting (Rochim et al., 2021).

We also created an ensemble variation of the kNN, SVM, and DT models. Specifically, we created models that first predict dual noun classes hence they predict the plural and singular noun classes first. For instance, when given the noun *abantu* 'people' each model would predict the noun class pair '1/2'. The final noun class prediction is then determined based on the two predicted classes (i.e., 1 and 2) based on the probability associated with each of the two classes via the *predictProb* function from the *scikit-learn* library.

We computed the precision, recall, and F1 scores for all models using the test set.

7 Deep learning-based models

We created two types of deep learning models. One is a simple neural network that is trained from

[3] https://scikit-learn.org/

scratch while the second is a pre-trained large language model that supports isiZulu and fine-tuned for the task at hand. We describe each of the models in the following subsections.

Simple feed-forward neural network We created a fully connected feed-forward neural (FNN) network that consists of an input layer, two hidden layers, and an output layer. The FNN was trained using isiZulu embeddings sourced from Adelani (2022), therefore, it may capture not only morphosyntax but other linguistic features. We relied on *FastTextKeyedVectors* for loading the embeddings so that out-of-vocabulary words can be inferred. The hidden layers make use of a ReLU activation function and they are followed by dropout layers to prevent over-fitting. The neural network's hyper-parameters are listed in Table 6. This was created to act a simple baseline for the pre-trained model.

Pre-trained LLM We also fine-tuned Serengeti (Adebara et al., 2023), a model based on the XLM-R (Conneau et al., 2020) architecture, by updating all parameters through the *transformers* library[4] and the *trainer* application programming interface[5] using a training batch size of 16, with 100 training warm-up steps, and a weight decay of 0.01. Serengeti was originally pre-trained and tested on a variety of tasks which include named entity recognition, part of speech tagging, and phrase chunking but not noun classification hence there is no additional baseline to compare against, other than the newly created FNN.

We computed the precision, recall, and F1 scores for both models using the test set.

8 Results

The results of the internal evaluation of the reproduced knowledge-infused technique are presented in Figure 2. The best performing model relies on a expert labelled dataset, with syntax and verb information, for component D (*N-GW* in Table 4), which achieves an accuracy of 85%. The best performing model that was created from the largest automatically labelled dataset, at the word-level, has an accuracy that is lower by 16.99%. The prefix-only models that only rely on the prefix to classify nouns perform the worst with an accuracy of 36.6%.

The results from comparing all the developed models are provided in Table 5. The traditional machine learning-based approaches that rely only on morphosyntax, make use of compressed data, and used in an ensemble approach, perform the best. Specifically, the best support vector machine model has an F1 score of 0.9736. The model performs comparably to the best model that relies on morphology, syntax, and semantics with 0.965 and performs slightly better than the best performing morphosyntax-based model that makes use of uncompressed data (3% difference).

9 Discussion

We now revisit the problem of inferring a noun's noun class by the various techniques and determining whether the use of semantics, syntax, and morphology, in a human-guided setting, yields the best results. The results obtained show that the neural-based models that make use of semantics, syntax, and morphology without human-guidance (the FNN and pre-trained LLM) and the traditional machine learning models that rely only on morphology, with less human-guided knowledge, perform better.

For NC detection, is better to rely on data-driven models that use human-guided knowledge in a less labour intensive approach where there are fewer modules so that errors are not propagated and have less negative impact on performance. This is evidenced by the observation that all the models that achieve an F1 score above 0.9, as listed in Table 5, do not rely on significant human guidance that ensures that the task is solved via only the prefix or a semantic approach that mandates the identification of semantically similar words and infers the noun class based on related words. Even if the 'good performance' threshold is lowered to an F1 score of 0.8, we see that none of the knowledge-infused models obtained by replicating the work by (Byamugisha, 2022) can be considered as having good performance. In fact, all 14 models that meet that standard are either neural-based or make use of traditional machine learning models.

The performance difference between the FNN and LLM is small (4%), in particular considering the simplicity of the FNN. This suggests that pre-training offers limited benefit for the current task. When comparing the traditional machine learning

[4] https://huggingface.co/docs/transformers/index

[5] https://huggingface.co/docs/transformers/main_classes/trainer

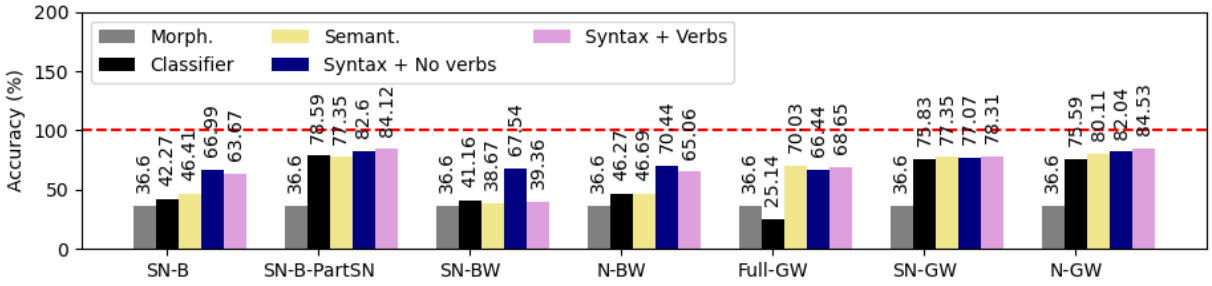

Figure 2: Accuracies in replicated models, across the models that differ based on the datatset used for the NC-Concord classifier module. The abbreviates used correspond to those detailed in Table 4.

Table 5: Precision, recall, and F1 scores of the best performing models. Abbreviations: FT = Fine-tuned, Ens = Ensemble, Prec = Precision, Rec = Recall, TF = Term frequency, and IDF = Inverse document frequency

Model	Prec.	Rec.	F1
Morphology, syntax, and semantics			
SN-B	0.714	0.591	0.604
SN-BP-PartSN	0.768	0.736	0.714
SN-BW	0.795	0.675	0.686
N-BW	0.743	0.641	0.655
Full-GW	0.625	0.565	0.576
SN-GW	0.789	0.771	0.762
N-GW	0.725	0.729	0.713
FNN	0.9213	0.9273	0.9209
Serengeti-FT	0.9642	0.9666	0.9650
Morphosyntax-based (uncompressed)			
kNN-TFIDF	0.7094	0.7149	0.6979
kNN-TF	0.6968	0.7281	0.6928
kNN-Ens.	0.7269	0.7302	0.7060
SVM-TFIDF	0.8222	0.8421	0.8273
SVM-TF	0.8424	0.8509	0.8439
SVM-Ens.	0.9367	0.9429	0.9385
DT-TFIDF	0.7300	0.7478	0.7133
DT-TF	0.7961	0.7917	0.7902
DT-Ens.	0.7916	0.8052	0.7691
Morphosyntax-based (compressed)			
kNN-TFIDF	0.8640	0.8770	0.8585
kNN-TF	0.8349	0.8070	0.7970
kNN-Ens.	0.8693	0.8662	0.8632
SVM-TFIDF	0.8608	0.8487	0.8419
SVM-TF+	0.8947	0.8904	0.8883
SVM-Ens.	**0.9742**	**0.9736**	**0.9736**
DT-TFIDF	0.8824	0.8706	0.8642
DT-TF	0.9038	0.8904	0.8859
DT-Ens.	0.9094	0.8991	0.8918

models, we see that most of the models that use compressed data perform better than their counterparts that use uncompressed data. In fact, the ensemble SVM model also outperforms deep learning models. This suggests Jiang et al. (2023)'s findings on the utility of compression also apply in the context of a Nguni language. Since not all the compression-based models outperform the neural models, this might demonstrate that there is utility in using minimal knowledge in traditional machine learning models. This is because the best performing model is an ensemble that exploits the fact that it is easier to identify the plural and singular noun classes of a noun vs. predicting the singular or plural in isolation. Then it disambiguates between just the plural vs. singular classes instead of 15, unlike the single class prediction problem. As such, this may indicate that there is value in using insights about a language in less labour intensive ways.

10 Conclusions

In reproducing the work by (Byamugisha, 2022) with different configurations and techniques, the results showed that the neural and ML models perform best, with an F1 score of 0.97, while the replicated models achieve a score of 0.71 despite their reliance on human-guided knowledge.

In future work, we plan to consider also other NCB languages and determine whether the number of 'ambiguous' prefixes among the number of prefixes might influence a technique's performance. We also plan to investigate the use of transformer-based and decoder-focused models.

Acknowledgments

This work was financially supported in part by the National Research Foundation (NRF) of South Africa (Grant Number and CPRR23040389063).

References

Ife Adebara, AbdelRahim Elmadany, Muhammad Abdul-Mageed, and Alcides Alcoba Inciarte. 2023. Serengeti: Massively multilingual language models for Africa.

David Ifeoluwa Adelani. 2022. *Natural language processing for African languages*. Phd thesis, Faculty of Mathematics and Computer Science of Saarland University.

Sonja Bosch, Laurette Pretorius, and Axel Fleisch. 2008. Experimental bootstrapping of morphological analysers for Nguni languages. *Nordic Journal of African Studies*, 17(2):23.

Sonja E Bosch and Laurette Pretorius. 2003. Building a computational morphological analyser/generator for Zulu using the Xerox finite-state tools. In *Proceedings of the Workshop on Finite-State Methods in Natural Language Processing, 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 27–34.

Joan Byamugisha. 2022. Noun class disambiguation in Runyankore and related languages. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4350–4359, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Joan Byamugisha, C. Maria Keet, and Brian DeRenzi. 2017. Toward an NLG system for Bantu languages: first steps with Runyankore (demo). In *Proceedings of the 10th International Conference on Natural Language Generation, INLG 2017, Santiago de Compostela, Spain, September 4-7, 2017*, pages 154–155. Association for Computational Linguistics.

Joan Byamugisha, C. Maria Keet, and Langa Khumalo. 2018. Pluralising nouns in isiZulu and related languages. In *Computational Linguistics and Intelligent Text Processing*, pages 271–283, Cham. Springer International Publishing.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Guy De Pauw, Gilles-Maurice de Schryver, and Janneke van de Loo. 2012. Resource-light Bantu part-of-speech tagging. In *Workshop on Language Technology for Normalisation of Less-Resourced Languages (SALTMIL 8-AFLAT 2012)*, pages 85–92. European Language Resources Association.

de Schryver, Gilles-Maurice. 2015. *Oxford Bilingual School Dictionary: isiZulu and English / Isichazamazwi Sesikole Esinezilimi Ezimbili: IsiZulu NesiNgisi, Esishicilelwe abakwa-Oxford. Second Edition*. Oxford University Press Southern Africa.

Sibonelo Dlamini, Edgar Jembere, Anban W. Pillay, and Brett van Niekerk. 2021. isiZulu word embeddings. In *Conference on Information Communications Technology and Society, ICTAS 2021, Virtual Event / Durban, South Africa, March 10-11, 2021*, pages 121–126. IEEE.

Tanja Gaustad and Cindy A. McKellar. 2024. Updated morphologically annotated corpora for 9 South African languages. *Journal of Open Humanities Data*.

Tanja Gaustad and Martin J. Puttkammer. 2022. Linguistically annotated dataset for four official South African languages with a conjunctive orthography: IsiNdebele, isiXhosa, isiZulu, and Siswati. *Data in Brief*, 41:107994.

Nikhil Gilbert and C. Maria Keet. 2018. Automating question generation and marking of language learning exercises for isiZulu. In *Controlled Natural Language - Proceedings of the Sixth International Workshop, CNL 2018, Maynooth, Co. Kildare, Ireland, August 27-28, 2018*, volume 304 of *Frontiers in Artificial Intelligence and Applications*, pages 31–40. IOS Press.

Derek Gowlett. 2014. Zone S. In Derek Nurse and Gérard Philippson, editors, *The Bantu languages*, chapter 30, pages 609–636. Routledge.

James Y. Huang, Kuan-Hao Huang, and Kai-Wei Chang. 2021. Disentangling semantics and syntax in sentence embeddings with pre-trained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1372–1379, Online. Association for Computational Linguistics.

Zhiying Jiang, Matthew Yang, Mikhail Tsirlin, Raphael Tang, Yiqin Dai, and Jimmy Lin. 2023. "low-resource" text classification: A parameter-free classification method with compressors. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6810–6828, Toronto, Canada. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers*, pages 427–431. Association for Computational Linguistics.

C. Maria Keet and Langa Khumalo. 2017. Grammar rules for the isiZulu complex verb. *Southern*

African Linguistics and Applied Language Studies, 35(2):183–200.

Leipzig University. 2024. Leipzig Corpora Collection: Zulu mixed corpus based on material from 2016.

Zola Mahlaza, Imaan Sayed, Alexander van der Leek, and C. Maria Keet. 2025. IsiZulu noun classification based on replicating the ensemble approach for Runyankore. In *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, pages 335–344, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jouni Maho. 1999. *A Comparative Study of Bantu Noun Classes*. Acta Universitatis Gothoburgunsis.

Carmen Moors, Ilana Wilken, Karen Calteaux, and Tebogo Gumede. 2018. Human language technology audit 2018: analysing the development trends in resource availability in all South African languages. In *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists, SAICSIT 2018, Port Elizabeth, South Africa, September 26-28, 2018*, pages 296–304. ACM.

Adian Fatchur Rochim, Khoirunisa Widyaningrum, and Dania Eridani. 2021. Performance comparison of support vector machine kernel functions in classifying COVID-19 sentiment. In *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pages 224–228.

Omid Shahmirzadi, Adam Lugowski, and Kenneth Younge. 2019. Text similarity in vector space models: A comparative study. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 659–666.

Sebastian Spiegler, Andrew van der Spuy, and Peter A. Flach. 2010. Ukwabelana - An open-source morphological Zulu corpus. In *Proceedings of the 23rd International Conference on Computational Linguistics, 23-27 August 2010, Beijing, China*, pages 1020–1028.

Jakobus S. du Toit and Martin J. Puttkammer. 2021. Developing core technologies for resource-scarce Nguni languages. *Information*, 12(12).

Appendix A. Hyper-parameters and linguistic information

In this appendix, we provide the hyperparameters used to train the FNN detailed in Section 7 and the noun classes with unique prefixes of which module A of the knowledge-infused model uses, as detailed in Section 5.

Table 6: Hyper-parameters used to train the neural networks

Hyper-parameter	**Value**
Activation function	ReLU
Optimizer	Adam
Learning rate	0.001
Epochs	11 - 20
Hidden Layer Sizes	256, 128

Table 7: List of classes whose prefixes uniquely identify a class in isiZulu.

Prefix	**Class**	**Prefix**	**Class**
aba	2	isi	7
abe	2	si	7
ba	2	zi	8
be	2	n	9
o	2a	m	9
bo	2a	zin	10
imi	4	zim	10
mi	4	lu	11
ili	5	ulu	11
il	5	bu	14
li	5	uku	15
ama	6	ku	15
am	6	pha	16
ma	6	ph	16