Computational Analyses of South African English – a Data-Driven Approach

By

Jacques de Lange

DLNJAC001

Supervised by

C. Maria Keet

A dissertation submitted for the fulfilment of the degree M Phil Information Technology

Department of Computer Science

University of Cape Town

February 2024

Declaration

I, Jacques de Lange, declare that the above thesis is my own unaided work, apart from the normal guidance from my supervisor and that no part has been or is being submitted for another degree at this or another university.

Signature: Signed by candidate Date: February 2024

Abstract

South African English across its multiple sub-varieties remains relatively understudied and an inclusive study of the language across the sub-varieties will enable us to uncover words and types of words unique to South African English that have been adopted or donated between the sub-varieties. This is important given South Africa's multilingual, multi-social society and the influence this has had on South African English. Such a study can also be used to improve large language models used in generative artificial intelligence, spellcheckers, sentiment analysis and speech to text technologies used in commercial applications. Computational techniques such as Part of Speech (POS) tagging, a sub-technique of Natural Language Processing, can be used to assist in understanding sentence structure and consequently, aid our uncovering of donor-adopter relationships between sub-varieties of a language. The accuracy of POS taggers on South African English therefore needs to be understood. This dissertation adopts computational data-driven approaches to studying South African English corpora to determine how accurate POS taggers are on South African English and if accuracy can be improved by creating extensions to a POS tagging model. A single layer neural network POS tagging model using word feature representations and a bidirectional long short-term memory (BLTSM) neural network POS tagging model, both trained on English are used as baseline models to predict POS tags on South African English corpora. Two modifications to the BLTSM model are then made, the first by creating a dual language model by including the Afrikaans language and the second by training the tokenizer and POS tagging neural processors of the dual language model on words unique to South African English. The evaluations show that the accuracy of the modified models is improved compared to the baseline models. The evaluation of baseline models when run on two South African English corpora shows a POS tagging F-Score of 0.69 on average across both corpora and baseline models. The evaluation of the modified models on the same corpora shows a POS tagging F-Score of 0.71 on average across both corpora and modified models. Evaluating the baseline models when run on words unique to South African English shows an average F-Score of 0.62 and evaluating the modified models when run on the same dataset shows an average F-Score of 0.72. The results demonstrate that improvements to POS tagging on South African English can be made by including Afrikaans in the model and by training this model on words unique to South African English. A novel Data-Driven Matching model is developed to investigate donor-adopter relationships in South African English. Results show that there is a commonality of use of words between South African English and Afrikaans, Sesotho and isiZulu. 15.7% of the words

in the South African English corpora studied are observed to be in use in Afrikaans, 4.98% of the words are used in Sesotho and 1.09% of the words are used in isiZulu.

Table of Contents

List of Tables

List of Figures

1. Introduction

The Dictionary Unit For South African English study the origins of South African English (SAE), which is the English that is written and spoken in South Africa, estimating that the majority of the unique words in SAE have been derived from other South African languages [19]. This makes SAE different from other varieties of English such as British, American, Canadian, Australian and New Zealand English as SAE contains words unique to South Africa. An example is the word *indaba* 'a meeting' [19]. SAE is further comprised of several sub-varieties [51] and the majority of SAE linguistic research studies sub-varieties in isolation, for example [29, 45, 51, 75]. An inclusive study of SAE across the various the sub-varieties of SAE is important given South Africa's multilingual, multi-social society and the influence this has had on SAE [7]. Such a study would enable us to uncover words and types of words unique to SAE that have been adopted or donated between the sub-varieties. It can also be used to improve large language models used in generative artificial intelligence (AI), spellcheckers, sentiment analysis and speech to text technologies used in commercial applications. Traditional anecdotal methods used to analyse text however, are time consuming. Techniques within the domain of computational linguistics such as Natural Language Processing (NLP) can facilitate this task by processing vast quantities of textual data and inferring some meaningful results thereof. This process begins by assigning tags to words using Part of Speech (POS) tagging, a sub-technique of NLP [13]. POS tagging assists with understanding sentence structure and consequently can be used to assist with understanding similarities or donor-adopter relationships between sub-varieties of a language [10]. Can computational techniques be used to identify donor-adopter relationships between SAE and other South African languages? How well do computational models deal with word disambiguation and word context in SAE? This dissertation aims to answer the following questions:

1. How accurate are existing POS tagging models on SAE text corpora?
2. Does the accuracy of the POS tagging models improve when including the Afrikaans language and by training a model on words that are unique to SAE?
3. Can a data-driven matching algorithm be used to identify donor-adopter relationships in text data between SAE and other South African languages?

Two experiments are conducted in this dissertation to answer these questions. The first experiment, consists of two parts, a baseline (E-1-baseline) and a modification (E-1-modification). E-1-baseline is setup to determine how accurate existing POS tagging models are on SAE text corpora and E-1-modification is setup to determine if the accuracy of the POS tagging models can be improved. The second experiment, is setup to use a data-driven approach to identify donor-adopter phenomena in SAE and how the phenomena is attributed across other South African languages. Experiment 1 and 2 are used to test the following hypotheses that are postulated:

Hypothesis 1: Modifying a POS tagging model by including the Afrikaans language improves POS tagging accuracy of SAE

Under the assumption that POS tagging models need to be improved, the dissertation postulates the hypothesis that the accuracy of existing POS tagging models run on SAE are improved when including the Afrikaans language in a POS tagging model. This improvement is expected to be observed as Afrikaans has played a prominent role in influencing SAE [35, 50, 73]. Experiment 1 tests this hypothesis.

Hypothesis 2: The accuracy of POS tagging models on SAE can be improved by training a POS tagging model on a dataset of English words unique to South Africa

The Dictionary Unit For South African English have published the Dictionary of South African English (DSAE) which is a dictionary of words that are unique to South Africa and are spoken in SAE [19]. It is expected that off-the-shelf POS tagging models for English will not have been trained on all of the words in the DSAE, as off-the-shelf English POS tagging models are trained on datasets based on British, American, Canadian, Australian and New Zealand English [17, 56]. It is therefore expected that off-the-shelf POS tagging model accuracy on SAE can be improved by training a model on words in the DSAE. Experiment 1 tests this hypothesis.

Hypothesis 3: SAE contains words adopted from other South African languages that can be observed using a data-driven approach

We expect donor-adopter relationships to be observed between SAE and other South African languages due to the influence that South Africa's multilingual society has had on SAE [42,

50, 51]. This dissertation postulates the hypothesis that the donor-adopter relationships can be observed using data-driven approach. Experiment 2 tests this hypothesis.

The dissertation is structured as follows: chapter 2 discusses related works, chapter 3 the methods, chapter 4 presents the results, chapter 5 discusses the results and chapter 6 concludes the dissertation.

2. Related Work

The related work falls within two domains, firstly the linguistic study of South African English (SAE) and secondly, computational techniques of text analysis of which part of speech (POS) tagging is studied.

2.1. Linguistic Studies of South African English (SAE)

English spoken in South Africa has its origins in a mixed colonial variety of British, Afrikaans and Dutch dialects, starting in the late 18th and early 19th centuries, to a wider variety of British, Yiddish and Afrikaans dialects in the late 19th and early 20th centuries [5]. This has resulted in expansions of regional variations of English spoken by the descendants of these immigrants [22] as well as several sub-varieties influenced by South Africa's multilingual, multi social society that are being used in different written and spoken formats, the combination of which form what we understand as South African English (SAE) [42, 50, 51]. Linguistic research on these sub-varieties has been observed to study a particular sub-variety or group of sub-varieties in isolation. Pienaar and de Klerk [51] constructed a corpus of Indian South African English (ISAE) comprising 60000 words transcribed from 30 oral dialog samples, studying the syntactic and lexical features of spoken ISAE. Mesthrie [43] analysed syntax in ISAE used by pre-school children, finding no evidence of innovation in syntactic features of their vernacular.

Black SAE (BSAE) is a variety of SAE used by first language speakers of African languages of South Africa [76]. A number of studies examine the linguistic features of BSAE. van Rooy [68] studied the differences between the "Tswana Learner English Corpus" (TLE Corpus) and the "Louvain Corpus of Native English Speaking Students" (LOCNESS) and found that the TLE Corpus places a greater emphasis on interpersonal, rather than informational communication, concluding that stylistic differences exist between the two corpora. Botha [8] extended this study by analysing the frequency of the words 'people' and 'some' in the TLE Corpus and uncovered evidence of undeletion phenomena in BSAE. Undeletion is defined by Mesthrie [44] as a phenomenon where elements of the language that are assumed to have been removed are restored. Hartmann and Zerbian [29] studied the degree to which rhoticity[1] exists in BSAE, concluding that gender and socio-economic levels influence the adoption of rhoticity and consequently, factors in creating further variations of BSAE. The

[1] Rhoticity is defined by Hartmann and Zerbian [29] as the pronunciation of the letter 'r' in a language.

stability of BSAE on a phonetical level was examined by Wissing [76] who concluded that BSAE is still in a transisionary phase to becoming a stable form of a new English. Similar results were uncovered by Singh and Parkinson [60] who studied grammatical features in BSAE using student assignment data and observed a lack of stability in sentence forms, concluding that BSAE as a sub-variety is still transitioning to a standard form. Mesthrie [44] however, argues that BSAE is regular on a mesolectal level, presenting evidence that undeletions are variable rather than categorical across constructions. van Rooy [67] analysed trends of linguistic patterns in three texts of BSAE: student writing from the TLE Corpus, published writing, informal conversations, and observed consistencies in the use of tense and aspect meaning across the datasets. Corpus linguistic research also supports the arugment of stability of constructions unique to BSAE [69]. Compared to white SAE, a sub-variety spoken by descendants of the original colonial variety of English, evidence of modal verb convergence, i.e., use of the words 'must' and 'should' is not supported [71].

The influence of Afrikaans on SAE has been studied to varying degrees. In studying white SAE, Wasserman and van Rooy [73] uncovered a prominent role Afrikaans has played in influencing modality in white SAE, for example with the use of words like 'must'. Furthermore, Kruger and van Rooy [35] demonstrated the influence Afrikaans has had on white SAE, uncovering evidence on syntactic co-occurence between Afrikaans English and SAE. More recently, Olayinka [50] analysed an extract from the Global Web-Based English Corpus and found that the optional syntactic elements of Afrikaans, sepecifcally, *ag* 'an interjection expressing a feeling of dislike', *ja* 'yes', *mos* 'but yet or indeed', *né* 'isn't that so', *nogal* 'rather or quite', *sommer* 'just' [19, 50] are present in and have influenced SAE. Analysing the same dataset, Unuabonah and Mtembu [64] demonstrated that the words *nje* 'just', *mara* 'but', *kanti* 'however', *vela* 'of course' and *kaloku* 'because' [61] have been adopted into SAE.

2.2. Part of Speech (POS) Tagging

Analysing large text corpora can be performed using computational techniques referred to as Natural Language Processing (NLP) tasks. One of the critical tasks within NLP is a process called part of speech (POS) tagging [59], where each word in the text is annotated with its corresponding part of speech that defines its behaviour in the language [17]. POS tagging models assign tags that form part of a particular POS tag framework. A commonly used set of POS tags is the Universal POS (UPOS) tag framework, which is a comprehensive and widely

used framework derived from in the Universal Dependencies[2] project. It defines a set of seventeen traditional parts of speech recognized to be in use across different languages [17]. A description and complete list of all UPOS tags is provided in Table 3.3. POS tags are used as inputs for other linguistic tasks such as analysing word frequency, named entity recognition [23], sentiment analysis [59], information retrieval and speech recognition [11]. The process of performing POS tagging raw text begins with tokenization, whereby sequences of characters in the text, referred to as tokens, are grouped into sentences, then words in each sentence are identified and thereafter each word is assigned a POS tag [56]. The process of grouping tokens into sentences is called sentence segmentation or sentence splitting and is performed by identifying the sentence boundaries through a match on sentence boundary tokens, typically punctuation marks such as a full-stop or question mark [56]. Identifying each word is then performed by identifying the word boundary, a space [56, 70]. These processes are pre-requisites for POS tagging as assigning a POS tag to a word requires the identification of the word. One challenge associated with identifying words using the space boundary is that errors in the raw text such as words separated by two successive spaces, or no space separating two words will lead to errors in POS tagging [70]. Another challenge using a space as the word boundary is that in some languages such as Chinese and Japanese, spaces do not explicitly separate words [20].

POS tagging can be represented as a generic model consisting of three components: Raw Text input, Tokenization processing and POS Tagging processing. This is depicted in the generic POS tagging model illustrated in Figure 2.1.

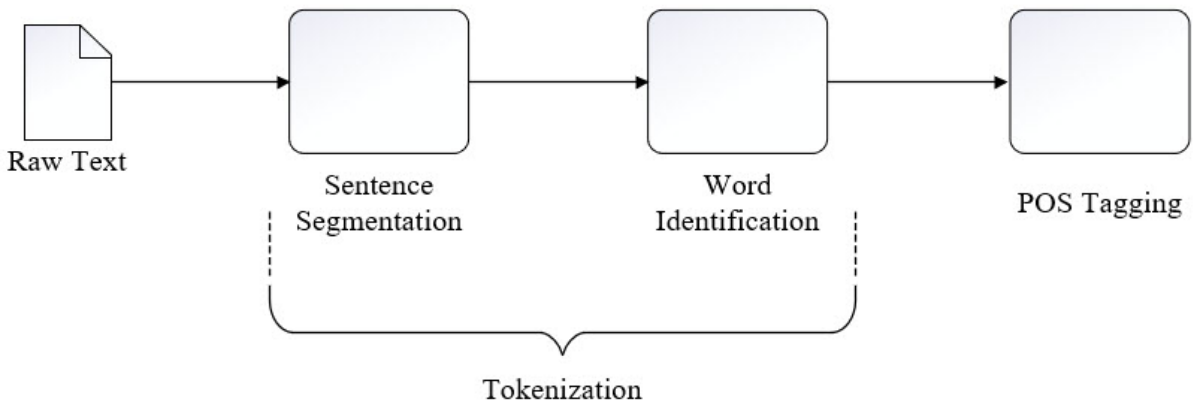

Figure 2.1: A POS Tagging model based on the Stanza architecture created by Qi et al. [56]

[2] The Universal Dependencies (UD) is a community framework for annotating grammar across multiple languages and available from https://universaldependencies.org/

This model can be illustrated with a basic implementation of a popular NLP Python library called the Natural Language Toolkit (NLTK) [6]. Assume for the purpose of this illustration that a user wishes to perform POS tagging on the Raw Text 'The adverse weather conditions have hampered rescue operations. Many roads are closed.'. The Python script is first prepared by importing the NLTK module:

```
import NLTK
```

The first step requires the input of the Raw Text to a variable, in this case `'raw_text'`:

```
raw_text = 'The adverse weather conditions have hampered rescue operations. Many roads are closed.'
```

The second step, Tokenization, is performed with NLTK's `word_tokenize` function, the output of which is stored in a Python Tuple:

```
NLTKtextK = nltk.word_tokenize(raw_text, language= 'English')
```

Tokenization performs sentence segmentation as illustrated in Figure 2.2 and then word segmentation as illustrated in Figure 2.3.

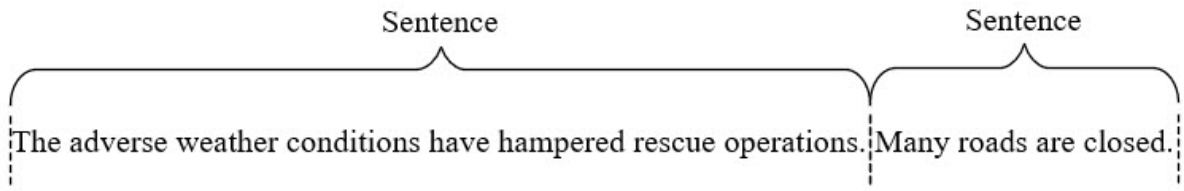

Figure 2.2: Sentence segmentation

Figure 2.3: Python Tuple NLTKtextK illustrating word segmentation

The final step, POS tagging, is performed on the tokenized data using NLTK's `pos_tag` function with results stored in a Python Tuple:

```
NLTKResultsK = nltk.pos_tag(NLTKtextK, tagset='universal')
```

The results can then be viewed in the `NLTKResultsK` tuple:

```
[('The', 'DET'), ('adverse', 'ADJ'), ('weather', 'NOUN'),
('conditions', 'NOUN'), ('have', 'VERB'), ('hampered', 'VERB'),
('rescue', 'NOUN'), ('operations', 'NOUN'), ('.', '.'), ('Many',
'ADJ'), ('roads', 'NOUN'), ('are', 'VERB'), ('closed', 'VERB'),
('.', '.')]
```

This tuple contains each word and its corresponding UPOS tag. An example is the word 'adverse', its corresponding UPOS is 'ADJ', an adjective.

POS taggers predict the POS tag for each word in a corpus as they have been trained to recognize POS tags from a prior training dataset. Different approaches to this problem are used, such as rules-based learning or machine learning techniques that use statistical models, neural networks or combinations of the aforementioned approaches and a variety of implementations are observed in the literature, for example [1, 6, 26, 33, 36, 41, 48, 77, 78]. Studies on this task have an interest in demonstrating the performance and accuracy of computational models.

Rules-based POS Tagging

Transformation-based learning (TBL) is an implementation of a rules-based approach and attempts to learn a set of rules from a labelled corpus. It was first applied to POS tagging by Brill [11] who developed the Transformation-Based Error-Driven Learning (TEL) Model, commonly referred to as the Brill tagger. Figure 2.4 is an illustration of this model.

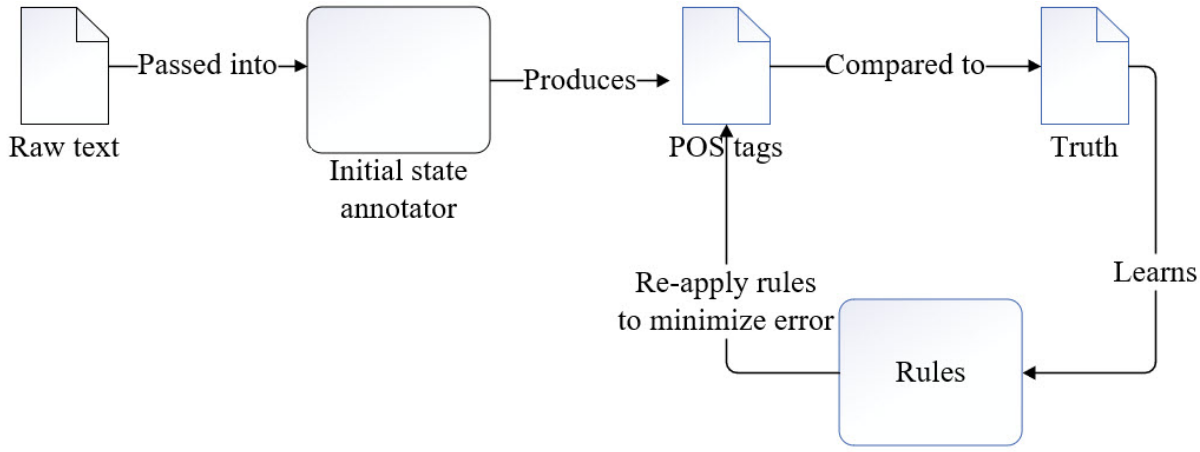

Figure 2.4: Illustration of a Transformation-Based Error-Driven Learning Model [11]

Raw text is passed into an initial state annotator to produce an initial set of POS tags using an existing POS tagging model. The choice of this POS tagging model can vary, for example as simple as assigning nouns to all words to more complex approaches that use stochastic modelling [11]. The initial POS tags are then compared to the Truth, an annotated corpus and then rules are learned to minimise the errors in tagging. The rules are re-applied to the output of the initial state annotator and the learning and error reduction process is iteratively repeated until the lowest errors are achieved. This iterative learning process is computationally expensive, however, which is one of the main drawbacks of TBL based POS tagging. One way of improving this is to optimise the rule learning process. This was demonstrated by Nguyen *et al.* [49] who created a framework using a Single Classification Ripple Down Rules (SCRDR) tree that stores existing rules that have been learnt (as opposed to re-learning them) and only adds new rules through each iteration of error correction. Nguyen *et al.* [49] reported that the training times of their SCRDR approach was 33 times faster than the training time required by the Brill Tagger and this result suggests that the computationally expensive learning process of TBL based POS tagging can be improved.

N-Gram Models

A unigram model is a statistical approach that assigns a POS tag to a word based on the highest frequency of the POS tag of the word in the training data [6]. The training dataset is analysed for which POS tags have the highest frequency for each word and then the model uses this information to assign POS tags to each word on the text that the model is run on [6]. This approach does not take preceding or proceeding words into account when calculating frequency, which means that the model cannot capture the context of the word or sentence in determining the POS tag [33]. This presents a drawback. Another major drawback of unigram models as well as TBL is their ability to deal with unknown words [33]. This drawback presents when the unknown word does not appear in the Truth corpus and contributes to the error rate of the learned rules when they run on the unknown words, impacting the accuracy of POS tag prediction. Whilst the problem of unknown words is present in all POS tagging models, statistical approaches attempt to alleviate this issue. Statistical approaches determine POS tags sequentially, that is, assigning a POS tag to the current word based on the POS tag of the previous word or words [6]. The Bigram, trigram and n-gram models are extensions of the unigram model that are commonly used in statistical approaches to POS tagging. The bigram model uses the POS tag of the preceding word the predict the POS tag of the current word. The trigram extends this model and uses the POS tags of the two preceding words to predict the POS tag of the current word. The n-gram model generalises the unigram, bigram and trigram models and uses the POS tags of the preceding n words, where n is greater than 2, to predict the POS tag of the current word.

Stochastic Models

Stochastic approaches such as those based on Markov processes, use probabilities to predict POS tags. The Hidden Markov Model (HMM) is a stochastic approach that models a sequence of POS tags (states), for a given sequence of words, where the sequence of POS tags cannot be observed directly (i.e., they are hidden) but the states are determined through modelling the probability distribution of each POS tag [14, 62]. The standard HMM can be represented mathematically as follows [14, 62]:

$$w_{1..n} = w_1, w_2, \dots w_n$$
$$t_{1..n} = t_1, t_2, \dots t_n$$

where $w_{1..n}$ is a sequence of words and $t_{1..n}$ is a sequence of POS tags. The assumptions under the HMM are that each tag t_i depends only on its preceding tag t_{i-1} and that each word w_i depends only on its corresponding tag t_i :

$$P(t_i|t_{1..i-1}, w_{1..i-1}) = P(t_i|t_{1..i-1})$$
$$P(w_i|t_{1..i}, w_{1..i-1}) = P(w_i|t_i)$$

The goal is to find a sequence of tags $t_{1,n}$ that are assigned to a sequence of observed words $w_{1,n}$ through maximizing the joint probability of the above conditional probabilities [14, 62]:

$$f(w_{1..n}) = \arg max_{t_{1..n}} \sum_{i=1}^{n} P(t_i|t_{1..i-1})\, P(w_i|t_i)$$

This Maximum Likelihood estimation can be performed with techniques such as the Viterbi algorithm [72] where the conditional probabilities $P(t_i|t_{1..i-1})$ and $P(w_i|t_i)$ have already been estimated or pre-trained with a machine learning model. POS tagging models that are created using variations of the HMM model create variations to the assumptions and, consequently, variations to the conditional probabilities and how they are estimated.

An example is a simple trigram HMM model, such as the one implemented by Merialdo [41]. In a trigram HMM model, the conditional probability $P(t_i|t_{1..i-1})$ is modified to $P(t_i|t_{i-1}, t_{i-2})$ since the assumption is made that the current POS tag can be predicted based on the two prior POS tags. The shortcoming of this model, however, is that there is often insufficient training data to estimate the conditional probabilities with sufficient accuracy, leading to unreliable model predictions [9, 12]. This issue is presented in the study by Merialdo [41] who concluded that more labelled training data was required to achieve better predictive accuracy.

An alternative solution is to use a computational technique to compensate for the data sparsity. Carlberger and Kann [12] started with a simple trigram HMM and investigated several approaches to modifying the model to improve it for POS tagging Swedish. They found that the best improvements were achieved when computing $P(t_i|t_{i-1}, t_{i-2})$ as a linear interpolation over the tri, bi and unigram probabilities. This technique is known as smoothing and was also used by Brants [9] who developed the Trigrams 'n' Tags (TnT) POS tagging model, based on

a second order Markov process. TnT performs linear interpolation over the unigram, bigram and trigram probabilities to produce smoothed trigram probabilities that addresses the sparsity issue. In addition, the TnT model attempts to improve the handling of unknown words by modelling the probabilities of word suffixes, under the assumption that a suffix can be used to predict the word class for an unknown word [9]. Brants [9] tested the model on English and German corpora and reported positive outcomes attributed to the smoothing and unknown word improvements. This model presents an improvement on the simple trigram HMM model. Banko and Moore [4] created a HMM POS tagging model that considers context before and after a word to predict the POS tag of that word. In this model, the probability of a POS tag being tagged correctly is dependent on the previous two tags and the probability of the word is dependent on the prior, current and next tag. The authors concluded that their model shows improvements over a similar trigram HMM but noted that the quality of training data when using an unsupervised approach can have a dramatic effect on the accuracy of the predictions.

In a Bayesian approach, Goldwater and Griffiths [27] implemented a trigram HMM to deal with the sparse data issue by introducing hyperparameters into the conditional probabilities as an a priori assumption. The study conducted an experiment using the Wall Street Journal (WSJ) treebank dataset and demonstrated improved performance over a simple trigram HMM, concluding that the Bayesian approach assists in dealing with the data sparsity issue. The aforementioned statistical models, however, all suffer from the label bias problem as described by Lafferty, McCallum and Pereira [37]. This implication for POS tagging is that the position of the word in the sentence impacts how much of the modelled conditional probabilities are used to predict the POS tag of the current word, with words towards the end of a sentence experiencing more bias. In 2001, Lafferty, McCallum and Pereira [37] introduced Conditional Random Fields (CRF) to solve this problem, building a POS tagger to test their model. CRFs compute the joint probability of the entire list of POS tags across the sequence of words being observed. They trained their model on the Penn Treebank corpus, comparing the POS tagging results to two Markov based models and showed that their model delivers lower errors rates, reporting that CRFs can reduce label bias in POS tagging. In 2003, Toutanova *et al.* [63] introduced a POS tagging model that used a Cyclic Dependency Network to capture bidirectional word context, lexical and unknown word features and regularization of a conditional loglinear model. The authors report model accuracy of 97.24% when run on the Penn Treebank WSJ data, improving on performance of previous POS tagging models run on this dataset.

POS tagging can also be performed at higher lexical categories such as word sentiment labels. Bravo-Marquez, Frank and Pfahringer [10] used supervised machine learning to combine stochastic gradient descent semantic orientation (SGD-SO) with pointwise mutual information SO (PMI-SO) to map words from Twitter data to positive, negative and neutral sentiment labels. The authors demonstrated that this combined approach produced superior results compared to using PIM-SO in isolation. The main contribution this study makes is in demonstrating that a stochastic model is able make inferences from unlabelled words to opinion tags, without the requirement of hard annotations in training data [10].

Neural Networks

Neural networks also feature prominently in natural language tasks, with varying complexity depending on the task being performed. The most basic form of a neural network is a single layer Perceptron. This model performs binary classification using an activation function applied to the weighted sum of a set of input nodes and corresponding weight parameters to produce a binary output [58]. An example of this is illustrated in Figure 2.5.

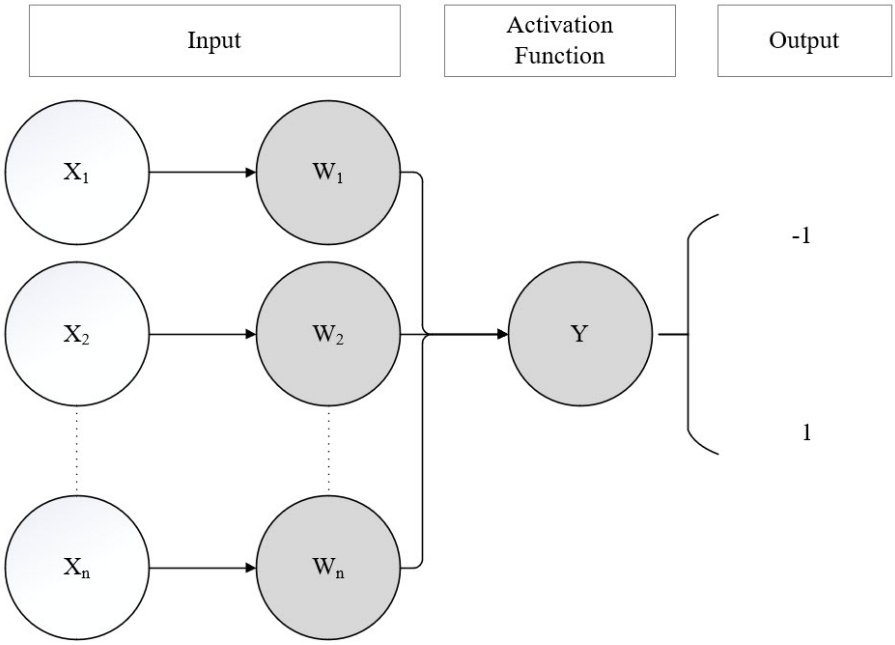

Figure 2.5: Illustration of a Perceptron recreated from Schuld, Sinayskiy and Petruccione [58]

The Averaged Perceptron POS tagger, developed by Honnibal [30], uses a Perceptron to perform linear binary classification. The model predicts the most likely POS tag for each word

based on a set of features of each word, where the features are a combination of the word, its context, its position, previous words and previous POS tags [6, 30]. The inputs $x_{1..n}$ are thus feature representations of the words and the weights $w_{1..n}$ are learned using an iterative approach of guessing the POS tag linked the feature representation, comparing the guess to the true value and updating the weights based on the accuracy of the guess. The weights are then averaged over all iterations to produce the trained weight vector used in the activation function. The strength of this model is its simplicity, performance, and ease of training. The model accuracy, however, is dependent on the extent to which the word features represent the word [30]. More complex neural networks have also been used for POS tagging. Carneiro, França and Lima [13] used a weightless artificial neural network model, the mWANN-Tagger, to train multilingual POS taggers and compared the model to a CRF model. These authors reported that the mWANN-Tagger produced comparable tagging accuracy relative to the CRF model, with superior performance during the training phase. The latter finding is the paper's main contribution. AlKhwiter and Al-Twairesh [1] trained two POS tagging models on a dataset of Arabic tweets using Conditional Random Fields (CRFs) and a Bidirectional Long Short-Term Memory (BLSTM) Neural Network, reporting an accuracy of 96.5% of the BLSTM model when run on a dataset of 3000 Arabic tweets. Kadari *et al.* [32] performed Combinatory Categorical Grammar (CCG) Supertagging using a BLSTM Neural Network. These authors compare this approach to a Recurrent Neural Network (RNN), finding that the BLSTM model improves on a shortcoming of a simple RNN - learning long term dependencies. The authors note that this shortcoming is known but that simple RNNs, which are easier to implement than BLSTM and are still a superior model for POS tagging, compared to unidirectional neural networks. Senthil Kumar and Malarvizhi [59] also studied the BLSTM, combining it with a convolution neural network (CNN) to perform POS tagging in sentiment analysis of Twitter data. One of the findings was that this combined network with fine-tuning was able to perform cross-domain POS tagging. Munoz-Valero *et al.* [47] implemented a RNN using Gated Recurrent Units (GRUs) to perform POS tagging. The authors note GRUs are computationally less intensive than the BLSTM model and can produce comparable results. They compared the results of the GRU based RNN to the NLTK POS tagger run on the Open American National Corpus[3] (OANC) and concluded that both POS tagging models achieve similar accuracy on this corpus.

[3] The OANC is a corpus of American English and available from https://anc.org

A Comparison of POS Tagging Models

Table 2.1 presents a summary of different POS tagging models and their main advantages and shortcomings.

Table 2.1: Comparison of POS Tagging Models

Category of Models	POS Tagging Models	Main Advantages	Main Shortcomings
Rules-based	Brill, SCRDR	It is easier to understand how the POS tags are learned.	The iterative learning process for error reduction is computationally expensive.
N-Gram	Unigram, bigram, trigram, n-gram, TnT	Model implementation is simpler.	Dealing with unknown words and capturing the context of the word use can be challenging. Higher order n-gram models attempt to address this.
Stochastic	HMM, CRF	Stochastic methods can improve on n-gram models when making inferences to unknown words if the data sparsity issues are sufficiently addressed.	Data sparsity in training data can present a challenge to accurately model conditional probablities. Bayesian variations of these models attempt to address this.
Neural Networks	Averaged Perceptron, mWANN, BLSTM, RNN, CNN, GRU	Word features can be modelled, context is better captured, models have proven to be successful for multilingual tasks.	The model training process can become computationally expensive.

Studies on SAE

Computational studies on SAE are limited. In a study performed on a subset of the TLE Corpus, van Rooy and Schäfer [70] implemented three POS tagging models on a sample of 2159 tokens from the TLE Corpus to determine the effect that spelling errors had on incorrect POS tag predictions and found that 19% of the errors in the TOSCA-ICLE POS tagger, 38% of the errors in the CLAWS POS tagger and 14% of the errors in the Brill POS tagger were caused by spelling errors. These observations indicate that computational POS tagging models fail on words in BSAE that are unknown and these models should be able to be improved.

Several off-the-shelf pre-trained POS taggers are available in NLP frameworks such as spaCy [31], Stanford CoreNLP [40], NLTK [6] and Stanza [56]. NLTK is a collection of libraries in Python developed for various NLP tasks. It provides a number of POS tagging modules including the Brill Tagger, TnT, unigram, bigram, trigram, HMM, CRF and the Averaged Perceptron, which is the model they recommend [6]. An illustration of the model has been demonstrated earlier in this chapter. Stanza is a NLP toolkit in Python, trained on multiple languages and is capable of performing six NLP tasks, of which tokenization and POS tagging are used in this dissertation [56]. The architecture of Stanza is depicted in Figure 2.6.

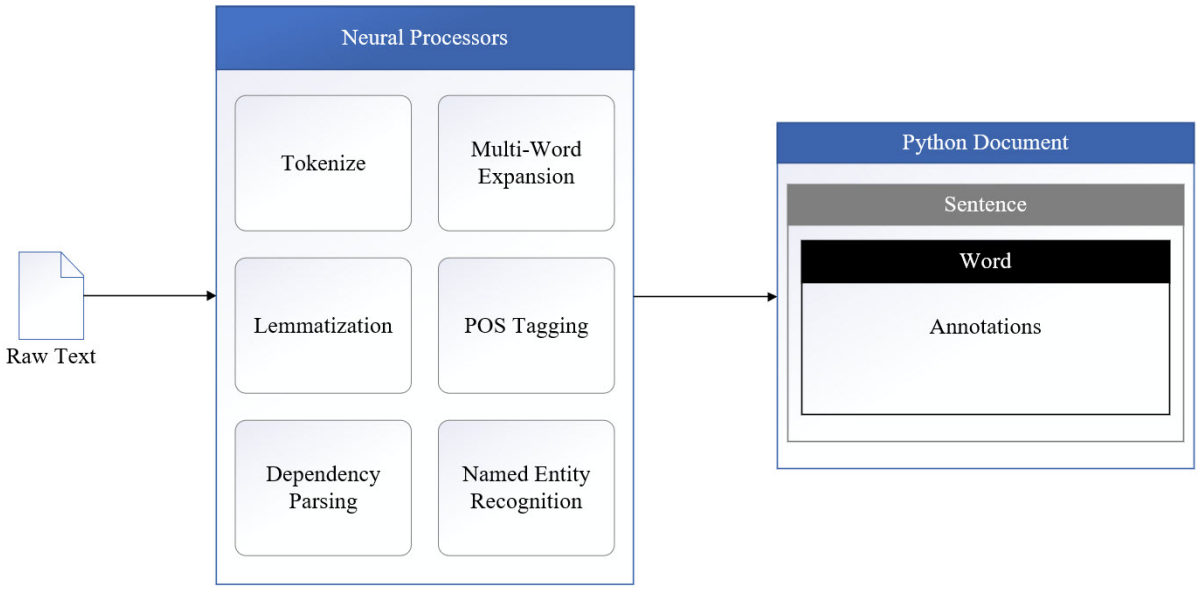

Figure 2.6: Stanza architecture recreated from Qi et al. [56]

Raw text is passed into a neural pipeline which contains the six neural processors. The neural processors run on the raw text and produce outputs, for each word, in each sentence, all of which are stored in a Python Document. The annotations are the outputs of the processors for example the POS tags. The code below illustrates how this can be done in Python:

```
import Stanza
raw_text = 'There are no clouds in the sky. It has not rained all
day.'
stanza_pipeline =
stanza.Pipeline(lang='en',processors='tokenize,pos,mwt,lemma,
depparse,ner')
stanza_output = stanza_pipeline(raw_text)
```

where `raw_text` is a string input of text to be processed, `stanza_pipeline` is the neural pipeline that is setup using the Pipeline module in the Stanza library and `stanza_output` is the Python Document that is created when the text is passed into the pipeline with the `stanza_pipeline(raw_text)` expression. The Python Document can then be accessed to retrieve the required annotations. An extract is shown in Figure 2.7.

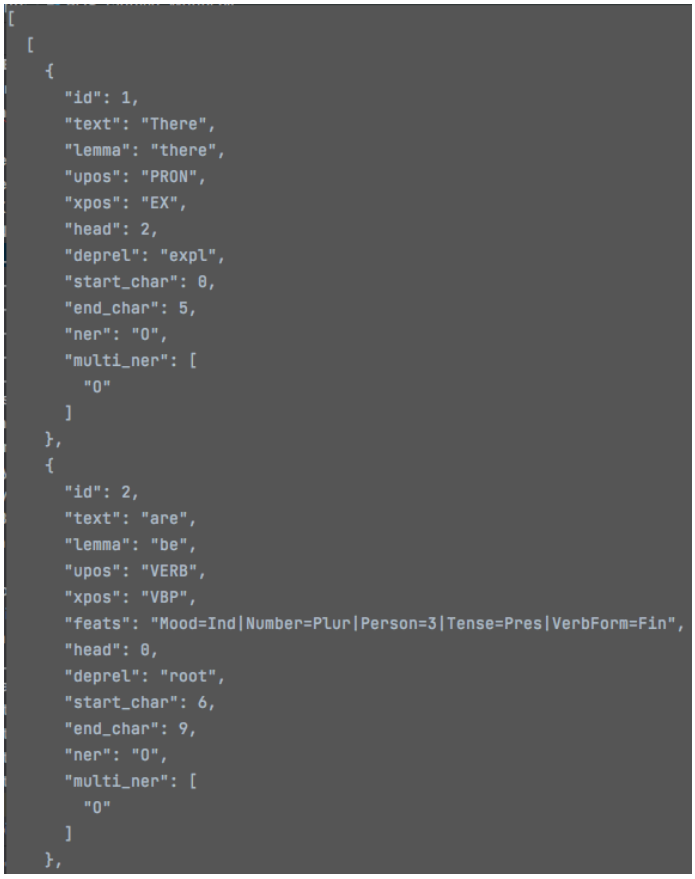

Figure 2.7: Stanza Python Document extract

The first word ‘There’ with id=1, has been tagged with the UPOS tag = ‘PRON’, a pronoun, and the second word ‘are’, id=2, has been tagged with the UPOS = ‘VERB’, a verb. Similarly the other annotations can be extracted. The Stanza POS Tagging processor uses a bidirectional long short-term memory (BLTSM) neural network model and each neural processor within the pipeline can be re-trained on new datasets [56]. The model’s main advantage is that is has been trained on multiple languages, including English and Afrikaans which are relevant for SAE.

This majority of linguistic research on SAE has been observed to study sub-varities of SAE in isolation, focussing on limited datasets, studying specific words or attributes of a sub-variety. Studies of POS tagging on SAE are also limited and based on smaller datasets.

3. Methods

This dissertation aims is to determine how accurate existing off-the-shelf POS tagging models are on SAE text data corpora and if POS tagging accuracy can be improved by including the Afrikaans language and by training the model on words contained in the DSAE. Experiment 1 is conducted to achieve this and the experiment, materials, model and evaluation methods are described in section 3.1. Furthermore, the dissertation aims to determine if adoption-phenomena can identified between a SAE text and an evaluation dataset comprised of words in other South African languages using a data-driven matching algorithm developed for this purpose. Experiment 2 is conducted to achieve this and the materials and model are described in section 3.2.

3.1. Experiment 1: Part of Speech (POS) Tagging South African English (SAE)

Experiment 1 consists of two parts, firstly, establishing a baseline set of results using off-the-shelf POS tagging models and secondly, modifying an off-the-shelf POS tagging model to determine if the accuracy improves. The models are introduced in section 3.1.2. The procedure to conduct Experiment 1 developed by the author is illustrated in Figure 3.1. The procedure begins with the collection of all materials required to run the experiment and are described in Table 3.1. Materials collected from source are in differing raw formats and to enable the POS tagging models to read in the datasets, each needs to be standardised to text data saved into a SQL[4] database. This is performed by the materials preparation process. Thereafter, each word is assigned a POS tag and the POS tags are normalised to the UD POS Tags via the standardisation to Universal Dependencies (UD) POS tags process described in section 3.1.1. Once the data has been prepared, the models are coded in Python[5] during the model preparation process and then the POS tagging models are run on the data and results are evaluated. The details for Experiment 1 are provided in sections 3.1.1., 3.1.2. and 3.1.3.

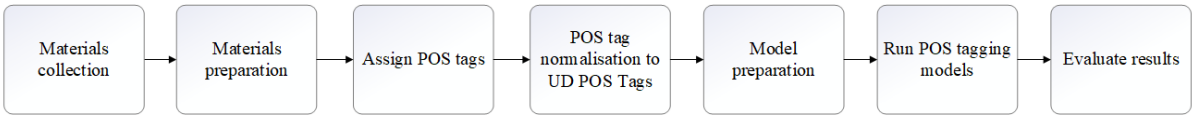

Figure 3.1: Experiment 1 procedure

[4] SQLite version 3.19.3 is used, available from https://www.sqlite.org/releaselog/3_19_3.html
[5] Python version 3.9.2, available from https://www.python.org/downloads/release/python-390/

3.1.1. Materials: SAE Text Corpora, Evaluation Data and Training Data

The materials collected and created are summarised in Table 3.1 and described in the remainder of this section.

Table 3.1: Materials

Data set	Data Type	Description	Size
News Articles Text Corpus	POS tagging is performed on this SAE text corpus	10 opinion articles from Mail & Guardian and 20 headline articles from the Daily voice sourced electronically by the author	17541 words in 813 sentences
Khan SAE Text Corpus [34]	POS tagging is performed on this SAE text corpus	Largest and most comprehensive electronic corpus of SAE available, produced by Khan [34]	1051367 words in 50366 sentences
Dictionary of South African English (DSAE)	Dataset used to evaluate POS tagging accuracy	Words unique to South Africa, spoken in SAE, produced by the Dictionary Unit for South African English [19]	13675 words comprising 4665 words and 9010 word forms
South African Directory Enquiries Name Corpus	Corpus used to evaluate POS tagging accuracy	Placenames transcribed from speech records of English, isiZulu, Sesotho and Afrikaans first language speakers by van Heerden [66]	16000 words
NCHLT Afrikaans Text Corpora	Corpora used to evaluate POS tagging accuracy	Annotated Afrikaans text collected from South African Government websites sourced by Puttkammer *et al.* [55] during the NCHLT Text project	6962 words
WordNet 3.1	Dataset used to evaluate POS tagging accuracy	An online database of verbs, adverbs, nouns and adjectives in the English language produced by Princeton University [52]	155001 words
Manual tags	Dataset used to evaluate POS tagging accuracy	Numbers and punctuation symbol characters tagged manually, they are listed in Table 3.2	10 numbers, 9 punctuation marks, 16 symbols
DSAE Training Corpus	Dataset used to train the POS tagging model on words in the DSAE	*Subject-Verb-Object* sentence forms generated for each word and word form in the DSAE dataset, sentence forms are listed in Table 3.7	68614 words and 13823 sentences

SAE Text Corpora

Two corpus datasets representative of South African English are used in this study. The first is a collection of news articles sourced by the author and the second one is an existing corpus produced by Khan in 2020 for his BSc (Honours) thesis [34].

The first SAE text corpus, the News Articles Text Corpus, is comprised of ten opinion articles from the Mail & Guardian [28] and twenty headline articles from Daily Voice [16], both collected in November 2021 based on the articles being accessible during collection. Article content is extracted electronically using a Python web-scraping script that is created for this purpose. The script is setup to iterate over each URL, downloading the html file using the urllib[6] package and then identifying and extracting content using the beautifulsoup4[7] package. The data is pre-processed to convert the encoding of the incoming data to UTF-8, removing text linked to images and social media and removing carriage returns. The pre-processed data is then staged to a SQL database table from where the POS tagging models source the data. The pre-processed dataset forms the News Articles Text Corpus and comprises 17541 words and 813 sentences.

The second SAE text corpus, the Khan SAE Text Corpus, is a dataset across multiple sub-varieties of SAE, sourced from blogs, government media, South African news media fiction blogs and Twitter data as well and novels [34]. This is the most comprehensive and largest electronic corpus of SAE currently in existence and hence its inclusion in the study. The corpus datafile in XML format is first pre-processed with an Extract-Transform-Load (ETL) workflow. The process entails reading in the file with UTF-8 encoding and looping over the `teiCorpus/TEI/text/body` XPath, extracting the text in the `<p>` node at each iteration. This is done to extract the paragraphs of text, as this dissertation is interested in examining the data content, excluding article titles. 1113 unique paragraphs containing 1051367 words and 50366 sentences are extracted and staged to the database.

To examine the accuracy of POS tagging on words unique to South Africa, i.e., words in the DSAE observed in the two SAE text corpora, subsets of each corpora are examined. The News

[6] urllib version 1.26.12, available from https://docs.python.org/3/library/urllib.html
[7] beautifulsoup4 version 4.9.3, available from https://pypi.org/project/beautifulsoup4/

Articles DSAE subset is a subset of words in the DSAE that are observed in the News Articles Text Corpus and the Khan DSAE subset is a list of words in the DSAE that are observed in the Khan SAE Text Corpus.

Evaluation Datasets

Evaluation of model accuracy requires an evaluation dataset to measure model accuracy against. One approach to produce the evaluation dataset is to manually annotate each word in the text corpora with a POS tag. The POS tags obtained via this process would then need to be converted to the Universal Dependencies (UD) POS tags [17], as the POS tagging models used in this dissertation produce UD POS tags [6, 56]. This manual annotation process is time consuming, however, so a data-driven approach is adopted by this study. Four annotated evaluation datasets that are available electronically are sourced, and a fifth evaluation dataset is created by identifying symbols, punctuation and numbers found in the SAE text corpora via string searches.

The first evaluation dataset, The Dictionary of South African English (DSAE), is produced and maintained by the Dictionary Unit For South African English. The DSAE is an online accessible list of 4665 words[8] with their corresponding POS tags and word forms[9] that have been written and spoken in the English language in South Africa [19]. It represents the unique words that the sub-varieties of SAE have contributed to SAE and to English [57]. There are 9010 word forms in this dictionary and they are included in this evaluation dataset so in total, the DSAE dataset is comprised of 13675 words. These words are extracted electronically from the DSAE website using a Python web-scraping script created for purpose, using the urllib package to download html files and the BeautifulSoup4 package to extract text content. The extracted content is stored in the SQL database and forms the DSAE evaluation dataset.

The second evaluation dataset, The South African Directory Enquiries (SADE) Name Corpus, is a dataset containing placenames transcribed from speech records of English, isiZulu, Sesotho and Afrikaans first language speakers [66]. The corpus is downloaded from the SADiLaR repository [66], each is word assigned the UD POS tag PROPN and the resultant dataset is stored in the SQL database. The corpus consists of 16000 words and forms the SADE evaluation dataset.

[8] The DSAE contained 4665 words as of February 2021, retrieved from https://dsae.co.za/
[9] Word forms are variations of the form of the original word, for example “avondbloem” is a word form of the word “aandblom”, https://dsae.co.za/entry/aandblom/e00005

The third evaluation dataset, a collation of The NCHLT Afrikaans Text Corpora, are a collection of annotated Afrikaans text collected from South African Government websites during the NCHLT Text project [55]. These corpora consists of 6962 words and collectively forming the NCHLT Afrikaans Text evaluation dataset.

The fourth dataset, WordNet, which is not specific to SAE, is an online database of verbs, adverbs, nouns and adjectives in the English language that are used for various natural language processing operations [52]. The WordNet version 3.1 database files are downloaded from the WordNet Princeton University website[10] and stored in the SQL database. The dataset consists of 155001 words[11] are included in the WordNet evaluation dataset.

The fifth evaluation dataset comprises numbers and punctuation symbol characters in the SAE text corpora that are identified using LIKE matches on each word and tagged manually. The list of characters and corresponding UD POS tag are listed in Table 3.2.

Table 3.2: Characters tagged manually

Characters	Universal Dependency POS Tag
0 1 2 3 4 5 6 7 8 9	NUM
! " ' - . ; ? ` : | ‘ ’ ,“ ” • :	PUNCT
% & * + = ||~ # $ / @ £ ¥ … €	SYM

Standardisation to Universal Dependencies (UD) POS Tags

POS tags for a corpus are specific to the treebank that the corpus belongs to [17]. The implication is that to evaluate the predictions of POS tagging models across different corpora, the POS tags for each corpus and evaluation dataset need to belong to the same treebank or be converted to a set of POS tags that are comparable across all datasets. The Universal Dependencies (UD) framework is used for this purpose [17]. UD POS Tags are seventeen universal core parts of speech used for consistent and comparable annotations across multiple languages and are listed in Table 3.3 [17].

[10] https://wordnetcode.princeton.edu/wn3.1.dict.tar.gz
[11] The WordNet dataset consists of 155001 words as of March 2022

Table 3.3: Universal Dependency POS Tags [17]

UD POS Tag	UD POS Name
ADJ	Adjective
ADP	Adposition
ADV	Adverb
AUX	Auxiliary
CCONJ	Coordination Conjunction
DET	Determiner
INTJ	Interjection
NOUN	Noun
NUM	Numeral
PART	Particle
PRON	Pronoun
PROPN	Proper Noun
PUNCT	Punctuation
SCONJ	Subordination Conjunction
SYM	Symbol
VERB	Verb
X	Foreign Word / Other

The first four evaluation datasets downloaded have POS Tags belonging to different treebanks and hence each POS tag is converted via manual mapping to the equivalent UD POS tag. Table 3.4 lists the POS tags in the DSAE dataset, with each equivalent UD POS tag mapping performed by the author.

Table 3.4: DSAE UD POS Tags and UD POS Tag mappings [18]

DSAE POS Tag	UD POS Tag
adjectival phrase	ADJ
adjective	ADJ
adposition	ADP
adverb	ADV
adverbial phrase	ADV
auxiliary	AUX
combining form	ADP
conjunction	CCONJ
determiner	DET
interjection	INTJ
interjectional phrase	INTJ
noun	NOUN
noun phrase	NOUN
noun, passing into adjective	NOUN
participial adjectival phrase	X
participial adjective	ADJ
particle	PART
phrase	X
plural noun	NOUN
plural noun phrase	NOUN
prefix	ADP
preposition	ADP
pronoun	PRON
suffix	ADP
verb	VERB
verb intransitive	VERB
verb transitive	VERB
verb transitive and intransitive	VERB
verb transitive and reflexive	VERB
verbal noun	NOUN
verbal noun phrase	NOUN

Table 3.5 lists the NCHLT Afrikaans Text Corpora POS Tags [55] with mappings to the UD POS tags performed by the author and Table 3.6 lists the WordNet POS tags [52] with the equivalent UD POS tag mappings performed by the author.

Table 3.5: NCHLT Afrikaans Text Corpora POS Tags [55] and UD POS Tag mappings

Afrikaans POS Tag	Afrikaans POS Tag Name	UD POS Tag
A	Adjektiewe	ADJ
B	Adverbia	ADV
K	Konjunkte	CCONJ
L	Lidwoorde	DET
N	Naamwoorde	NOUN
P	Voornaamwoorde	PRON
R	Residu	NOUN
S	Setsels	ADP
T	Telwoorde	ADJ
U	Uniek/ongespesifiseerd	X
V	Verbia	VERB
W	Tussenwerpsels	INTJ
Z	Punktuasie	PUNCT

Table 3.6: WordNet POS Tags [52] and UD POS Tag mappings

WordNet POS Tag	UD POS Tag
n	NOUN
v	VERB
a	ADJ
R	ADV

Evaluation Data in the Data Driven Approach

To form the dataset that the data driven approach uses to retrieve the POS tag of each word to be evaluated, each of the five evaluation datasets are joined together to form one combined dataset containing a list of words with the corresponding UPOS tag of each. This dataset is looked up to in the evaluation process.

Training Corpus

Training a POS tagging model requires the training dataset to consist of a set of natural language sentences that are labelled with each word's POS tag. To train a POS tagging model on words in the DSAE, a corpus consisting of a set of artificially controlled natural language sentences including all of the words and word forms in the DSAE is constructed and labelled with UD POS tags. This constructed corpus is considered superior to the baseline alternative,

which is list of words in the DSAE as there is no alternative training dataset specifically tailored to SAE that the model can be trained on.

The first step is to create sentence forms for each POS tag in the DSAE dataset using the form *Subject-Verb-Object* as a basis. The complete list of sentence forms are listed in Table 3.7.

Table 3.7: DSAE POS Tags [18] and author generated sentence forms

DSAE POS Tag	Sentence
noun	#word# read a news article.
prefix	#word# read a news article.
plural noun	#word# read a news article.
verb transitive	We must #word# them.
noun phrase	We must watch the #word#.
verb transitive and intransitive	We must #word# the country.
participial adjective	We are #word# people.
adverbial phrase	We are #word# people.
interjection	#word# we are people.
adjective	We are #word# people.
phrase	We say #word# to people.
adverb	We work #word# for money.
plural noun phrase	We have #word#.
verbal noun	We are #word# people.
verb	He #word# a news article.
combining form	We went to #word#city.
interjectional phrase	We say #word# to people.
verb intransitive	We will #word# ourselves.
preposition	We went #word# the river.
participial adjectival phrase	We are #word# people.
verbal noun phrase	We are #word# the ground.
adjectival phrase	We went to #word# building.
suffix	We see the road #word#.
conjunction	We are studying English #word# only after Afrikaans.
verb transitive and reflexive	We #word# food.
pronoun	#word# come here.
noun, passing into adjective	We go to #word#.
particle	We go #word# the building.
auxiliary	We #word# go to the river.
determiner	We will go to #word# river.
adposition	We are #word# the people.

The second step is to construct a sentence for each word in the DSAE dataset from the sentence form matching the POS tag of the word. To illustrate this with an example, the sentence form for a noun is '#word# read a news article.'. A sentence for the noun *Aandblom* 'a plant of the Iridaceae family' [19] is then constructed as 'Aandblom read a news article.'. This process is repeated for the words in the DSAE dataset which produced a corpus with

68614 words and 13823 sentences, which is stored in a SQL database, then exported to a raw text file that is used by the training model.

The final step is to produce the training file in CoNLL-U format as this is the required format for training neural processors in the Stanza model [56]. The Universal Dependencies project specifies the CoNLL-U data format as a UTF-8 encoded flat file containing three types of lines for each sentence [17]:

- A comment line for the sentence starting with # text =
- Tab-separated fields for the attributes of each word, provided in Table 3.8
- An empty line marking the end of a sentence

Table 3.8: CoNLL-U fields sourced from Universal Dependencies[12]

ID	Index of the word in the sentence, starting at 1
FORM	The word in the sentence
LEMMA	The lemma or stem of the word
UPOS	the UD POS tag
XPOS	The POS tag specific to the treebank of the corpus
FEATS	Morphological features of the word, can be made _ if not available
HEAD	Head of the word
DEPREL	UD relation the HEAD
DEPS	Set to _
MISC	Set to _

To illustrate the process for the sentence 'Aandblom read a news article.' :

text = Aandblom read a news article
1 Aandblom Aandblom NOUN NN Number=Sing 2 nsubj _ _
2 read read VERB VB Mood=Ind|Number=Sing|Person=3|Tense=Past|VerbForm=Fin 0 root _ _
3 a a DET LS Definite=Ind|PronType=Art 5 det _ _
4 news news PROPN VB Number=Sing 5 compound _ _
5 article article NOUN NN Number=Sing 2 obj _ _
6 . . PUNCT . _ 2 punct _ _

To produce the tab-separated fields for this sentence, an ID is assigned to each word in the sentence, the first word assigned the value of 1 and the last word being the full stop, assigned a value of 6. The UPOS, which is the UD POS tag is obtained from the UD POS tag mapping on the DSAE evaluation dataset and the XPOS tags are equivalent manual mappings to these

[12] https://universaldependencies.org/format.html

UPOS tags. The remaining fields are generated by passing the sentence through a Stanza version 1.0 neural pipeline [56] and extracting the required fields. Where a field is not available, it is assigned the character of _. This process is repeated for all 13823 sentences in the training corpus to produce the complete CoNLL-U training file which contains 58183 records. This training corpus is named the DSAE Training Corpus.

3.1.2. Experiment 1 Models: Baseline (E-1-baseline) and Modification (E-2-modification)

Experiment 1 consists of two parts, the baseline (E-1-baseline) and the modification (E-2-modification). E-1-baseline is performed to establish the accuracy of two off-the-shelf POS tagging models, NLTK version 3.7 [6] and Stanza version 1.0 [56] on the SAE text corpora. This establishes the baseline results. Thereafter, E-2-modification is performed by setting up two POS tagging models, the first, a modification of the Stanza model to include the Afrikaans language to create a dual language model, and, the second, training this dual language model on the DSAE training corpus. All models are implemented in Python version 3.9.2 using PyCharm[13] version 2021.2.4 as the integrated development environment. The pipeline for experiment 1 is illustrated in Figure 3.2 and detailed descriptions of the models are provided thereafter.

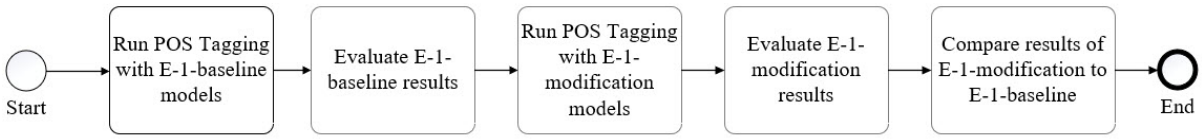

Figure 3.2: POS tagging experiment pipeline

E-1-baseline models

The two models used for the baseline experiment are NLTK's Averaged Perceptron POS Tagger version 3.7 pretrained on English and the Stanza POS Tagging model version 1.0 pretrained on English. These models are named NLTK EN and Stanza EN respectively. NLTK's Averaged Perceptron is selected as it has been trained using word feature representations that incorporate word context and it is expected that this will deliver superior accuracy when dealing with word disambiguation in SAE compared to simpler n-gram HMM

[13] https://www.jetbrains.com/pycharm/

models that only rely on POS tags of previous words for predictions [30]. The Averaged Perceptron POS tagger is accessed through the `pos_tag` function in Python:

```
import nltk
tokenized_data = nltk.word_tokenize(input_corpus_string)
tagged_data = nltk.pos_tag(tokenized_data)
```

where `input_corpus_string` is a string created from an import of the text corpus being analysed, tokenized_data is the tokenized string and tagged_data is the tagged output of the POS tagging model, producing each word in the input corpus and its corresponding POS tag. As tokenization is a requirement for POS tagging, the NLTK EN model setup for the baseline experiment includes tokenization. The model pipeline, developed by the author, is presented in Figure 3.3.

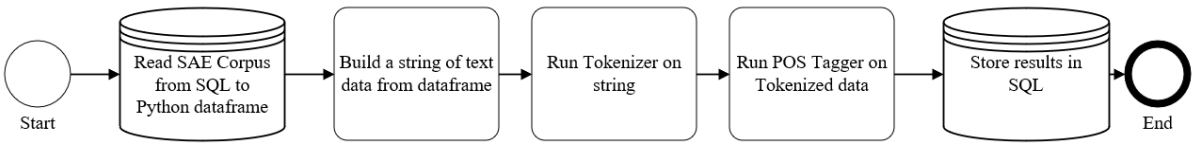

Figure 3.3: NLTK EN model pipeline developed by the author

Tokenization is set to run on the English language model and is performed with the `word_tokenize` function:

```
NLTKtextK =
nltk.word_tokenize(wordListStrK,language="english")
```

where `wordListStrK` is the string of text built from the data frame and `NLTKtextK` is the variable storing the tokenized data.

The POS tagger is then set to run using to `pos_tag` function, which runs the Averaged Perceptron POS tagger to produce Universal Dependency POS tags, specified by the `tagset` parameter:

```
NLTKResultsK = nltk.pos_tag(NLTKtextK,tagset='universal')
```

where `NLTKResultsK` is a variable storing the POS tagged data, which is then written to a SQL database. This model is referred to as NLTK EN.

The second baseline model, Stanza EN, is selected for its language-agnostic feature as it is expected that South African English will contain some multi-language features. The baseline experiment pipeline developed by the author is setup and illustrated in Figure 3.4.

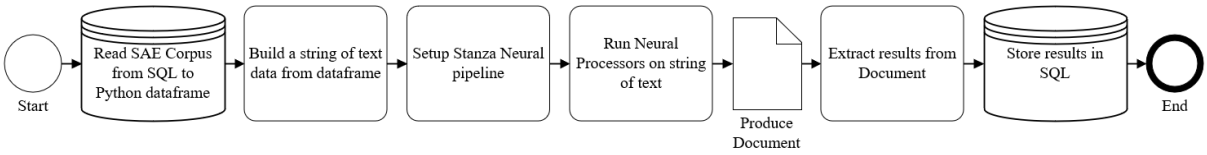

Figure 3.4: Stanza EN model pipeline developed by the author

The Stanza Neural pipeline is setup in Python as follows:

```
nlpposK = stanza.Pipeline(lang='en',
processors='tokenize,pos')
```

where `nlpposK` is the pipeline function created. The pipeline is setup to use the English language model by setting the lang parameter to ‘en’, the English language that the Stanford NLP Group trained the model on. The tokenize and pos neural processors used in this experiment are specified by the processors parameter. The tokenizer is included as it is a dependency of the POS processor. Once the neural pipeline function is setup, the string of text is passed into the function, which runs the processors and writes the results to an output document:

```
contentdocK = nlpposK(wordListStrK)
```

The output document `contentdocK` contains the sentences in the corpus, words in the sentences and POS tags of each word, stored as Python objects Qi *et al.* [56]. The POS tags are extracted by iterating over each word, in each sentence and stored in data frame (dfSAE), which is then written to a SQL database:

```
cursor = sqlite3.connect(db_file_name).cursor()
connection = sqlite3.connect(db_file_name)
sentence_id = 0
for sentence in contentdocK.sentences:
```

```
      sentence_id = sentence_id + 1
      for word in sentence.words:
            SAE = 'text': [word.text], 'upos': [word.upos], 'xpos':
[word.xpos]
            dfSAE = pd.DataFrame(SAE)
                  #append record to Database
                  dfSAE.to_sql('Stanza_Results_Table',   con=connection,
if_exists='append')
```

where `cursor` and `connection` are the SQL database connection configuration parameters.

E-1-modification models

This dissertation modifies an existing POS tagging model to produce two POS tagging models that contribute to POS tagging SAE. The first model, named Stanza EN_AF, is a dual language model that is setup by modifying the Stanza English language POS tagging model[14] to include the Afrikaans language. The second model, named Stanza SAE, is setup by training the Stanza EN_AF model on the DSAE Training Corpus and is thus a dual language POS tagging model that has also been trained on words that are unique to South African English.

Afrikaans words such as *bakkie* and *braai* are observed to be in use in SAE and it is expected that by including this language in the experiment model, improvements to the results can be made. Since Afrikaans is one of the languages that the Stanza model has been trained on by the Stanford NLP Group [56], the experiment pipeline for the Stanza EN_AF model is setup by modifying the Stanza English language neural pipeline to include Afrikaans in addition to English, i.e., a dual language model. This is achieved by including the `langs` parameter and specifying both languages:

```
stanza.Pipeline(langs='en,af', processors='tokenize,pos')
```

where `af` is the Afrikaans language.

To setup the Stanza SAE model, the training process is performed in the following sequence, based on the training guidelines from the Stanford NLP Group [61].

[14] Version 1.0 of the Stanza POS tagging model developed by Qi *et al.* [56] is used.

1. *Environment setup:* source code for stanza-train and stanza is cloned from the Stanza git repository and the environment variables referencing the required folders are set. The required Python libraries are also installed: numpy, tqdm, protobuf, request and torch.

2. *Data preparation:* the training data set produced in Section 3.1.1. is used for the training, test and development sets because the processors are being trained on all of the unique words. The files are saved in the prescribed filenames in the `../udbase folder/UD_English_TEST` folder:

```
en_test-ud-train.conllu
en_test-ud-train.txt
en_test-ud-dev.conllu
en_test-ud-dev.txt
en_test-ud-test.conllu
en_test-ud-test.txt
```

3. *Word vector file:* In addition to the training data, a pre-trained word vector file is required by the POS tag trainer. Since the processor is being trained for Stanza's English language model, the word vector file for English is downloaded and used as recommendation by Stanford NLP Group [61]. The word vector file is included in the download of the English language model: `stanza.download("en")`.

4. *Model preparation:* the tokenizer and POS tagger training datasets are converted to a format required by the training models. This is done via scripts provided by the Stanford NLP Group [61].

```
python3 -m
stanza.utils.datasets.prepare_tokenizer_treebank
UD_English-TEST
python3 -m stanza.utils.datasets.prepare_pos_treebank
UD_English-TEST
```

5. *Model training:* the training of the neural processors is then initiated using scripts provided by Stanford NLP Group [61].

```
python3 -m stanza.utils.training.run_tokenizer
UD_English-TEST --step 400
python3 -m stanza.utils.training.run_pos UD_English-TEST
--max_steps 400
```

Pre-experiment runs of the training performed on a MacBook Pro Intel Core i5 2.7 GHz with 8GB RAM with different step configurations set revealed that the best models can be achieved by 400 steps for the tokenizer and POS tagger. The trainers in the experiment are therefore set to stop at 400 steps to optimise the time taken to train the processors.

The neural pipeline for the Stanza SAE model is then setup to include English and Afrikaans languages and run tokenization and POS tagging on the new trained models. This is done by specifying additional parameters `tokenizer_model_path` and `pos_model_path` which are paths to the pre-trained files respectively.

```
stanza.Pipeline(langs='en,af', processors='tokenize,pos',
tokenizer_model_path ='../en_test_tokenizer.pt' ,
pos_model_path='../en_test_tagger.pt')
```

where `../` refers to the path that the pretrained processors are stored. Stanza SAE is thus a dual language model trained on unique South African words as well. The novelty of this design is that the model is setup to capture words that the off-the-shelf models have not been trained on previously.

3.1.3. Model Evaluation: F-Score, Precision, Recall, Accuracy and Specificity

The F-Score (F) along with Precision (P), Recall (R), Accuracy (A) and Specificity (S) are metrics commonly used to benchmark the performance of classification models [2, 3, 36]. The F-Score is used to measure how accurate the model is in identifying POS tags correctly and is calculated from model Precision (P) and Recall (R), defined as follows:

$$P = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (1)$$

$$R = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (2)$$

$$F = \frac{2\ x\ P\ x\ R}{P+R} \quad (3)$$

Accuracy (A) is defined as:

$$A = \frac{True\ Positives+True\ Negatives}{True\ Positives+False\ Positives+False\ Negatives+True\ Negatives} \quad (4)$$

Specificity is defined as:

$$S = \frac{True\ Negatives}{False\ Positives+True\ Negatives} \quad (5)$$

where:
True Positives = the number of words where the POS tags predicted by the model match the POS tags in the Evaluation Data excluding foreign word tags.
False Positives = the number of words where the model predicted the Foreign Word (FW) POS tag but the Evaluation Data had another POS tag.
False Negatives = the number of words where the POS tags predicted by the model did not match the POS tags in the Evaluation Data, excluding False Positives.
True Negatives = the number of words where predicted POS tag the Foreign Word (FW) POS tag and Evaluation Data had the Foreign Word POS tag as well.

These metrics are calculated for the UPOS predictions on all of the models where the evaluation data is used as the gold standard. To evaluate the outcome of the training process for the tokenizer and POS tagger in the Stanza SAE model, F-Scores are calculated using Stanza's evaluation script, which is based on the 2019 Universal Dependencies shared task [56]:

Tokenizer:

```
python -m stanza.utils.training.run_tokenizer UD_English-TEST --score_dev
```

POS Tagger:

```
python -m stanza.utils.training.run_pos UD_English-TEST --score_dev
```

These scripts produce F-Scores for token, sentence, word and UPOS tag metrics that indicate how well the predicted classifications from the trained processors compare to the actual classifications in the evaluation data [56].

3.2. Experiment 2: Donor-Adopter Relationships

The procedure developed by the author for experiment 2 is illustrated in Figure 3.5. The procedure entails the collection and preparation of materials, the setup of the model and thereafter the running of the model and evaluation of results.

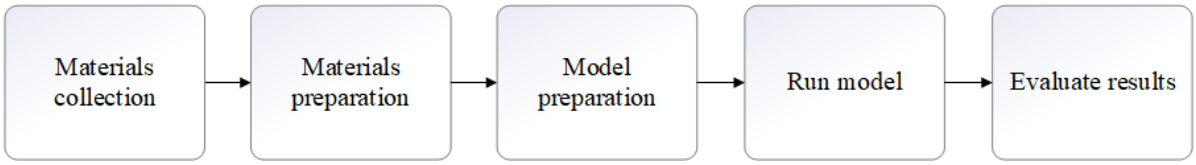

Figure 3.5: Experiment 2 procedure

3.2.1. Materials

The SAE Text Corpora, the News Articles Text Corpus and the Khan Text Corpus, used in experiment 1 are reused in experiment 2 and are analysed for donor-adopter phenomena. The evaluation dataset is created from lists of words that are used in other South African languages, enabling us to identify which words in the SAE Text Corpora are used in other South African languages as well. The full evaluation dataset comprises 214781 words across five subsets, summarised in Table 3.9.

Table 3.9: Adoption phenomena evaluation data

Dataset	Language	Number of Words
Mjaria isiZulu Spellchecker[15] **[46]**	isiZulu	88328
NCHLT Afrikaans Text Corpora[16] **[55]**	Afrikaans	66104
NCHLT Afrikaans Phrase Chunk Annotated Corpus[17] **[25]**	Afrikaans	2353
NCHLT Afrikaans Named Entity Annotated Corpus[18] **[24]**	Afrikaans	21646
NCHLT Sesotho Named Entity Annotated[19] **[24]**	Sesotho	15470
NCHLT Sesotho Phrase Chunk Annotated Corpus[20] **[25]**	Sesotho	1949
NCHLT Sesotho Annotated Text Corpora[21] **[54]**	Sesotho	5227
South African Directory Enquiries (SADE) Name Corpus **[66]**	Afrikaans, Sesotho, isiZulu	2981
Dictionary of South African English (DSAE) **[18]**	Words unique to South African English	10723

The isiZulu word list was sourced from the Mjaria isiZulu Spellchecker project [46]. To date, this is the most comprehensive isiZulu word list electronically available. The dataset contains 103329 words and is download in text format, then pre-processed to remove duplicates and 88328 unique records are imported into the SQL database. Three Afrikaans datasets containing words, named entities and phrases are download from the SADiLaR repository: the NCHLT Afrikaans Text Corpora [55], the NCHLT Afrikaans Phrase Chunk Annotated Corpus [25] and the NCHLT Afrikaans Named Entity Annotated Corpus [24]. 90103 words were extracted from the three corpora and imported into the SQL database. Three Sesotho datasets containing words and named entities are download from the SADiLaR repository: the NCHLT Sesotho Named

[15] https://github.com/fridamjaria/Isizulu_Spellchecker/blob/master/src/isizulu_spellchecker/resources/wordlist.txt
[16] https://repo.sadilar.org/handle/20.500.12185/293
[17] https://repo.sadilar.org/handle/20.500.12185/300
[18] https://repo.sadilar.org/handle/20.500.12185/299
[19] https://repo.sadilar.org/handle/20.500.12185/334
[20] https://repo.sadilar.org/handle/20.500.12185/335
[21] https://repo.sadilar.org/handle/20.500.12185/332

Entity Annotated Corpus [24], the NCHLT Sesotho Phrase Chunk Annotated Corpus [25] and the NCHLT Sesotho Annotated Text Corpora [54]. 22646 words were extracted from the three corpora and imported into the SQL database. The South African Directory Enquiries (SADE) corpus sourced for experiment 1 is reused in experiment 2. The languages belonging to each word, sourced from the SADE Name corpus [66], are included in the evaluation dataset for experiment 2 and 2981 words from the corpus are extracted and stored in the SQL database for experiment 2. 10723 words are extracted from the DSAE dataset [18] used in experiment 1 and stored in a SQL database for use in the evaluation dataset.

3.2.2. Experiment 2 Model: Data-Driven Matching Model

A model developed by the author for this dissertation, named the Data-Driven Matching (DDM) model, is a pipeline that combines tokenization, with word matching to an evaluation dataset to identify donor-adopter phenomena with other South African languages. Figure 3.6 illustrates this model.

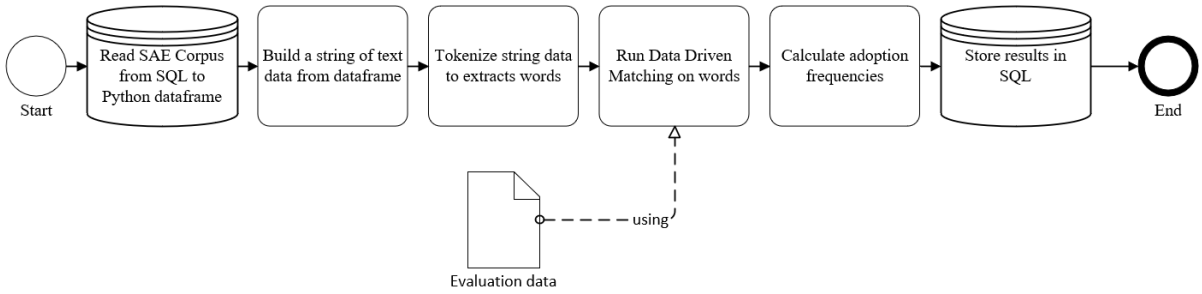

Figure 3.6: DDM model pipeline

The pipeline begins when SAE corpus data is retrieved from the SQL database into a Python data frame. A string of text is created from the data in the data frame and passed into the Stanza EN tokenizer processor, which creates a list of all of the words in the corpus. Once a list of all of the words are extracted and stored in a SQL database table, the Data-Driven Matching process is performed. The matching is performed using a Left Outer Join with the evaluation data on lowercase word. Where no matches are found it indicates that no donor-adopter relationships are. Where a match is found, the language that the word is adopted from or donated to is joined to the word and the resulting outflow is a list of words each with the name of a donor-adopter language, or null for no donor-adopter language. Attribution percentages to

the different South African languages are calculated as the number of words matched to each language divided by the total number of words in this dataset.

4. Results

This chapter presents the results of the dissertation in the order as described in chapter 3. They are discussed in chapter 5.

4.1. Experiment 1: Part of Speech (POS) Tagging South African English (SAE)

Results for experiment 1 are presented.

4.1.1. E-1-baseline Results

Table 4.1 summarises the results of the model predictions of the NLTK EN and Stanza EN models run on each corpus. Higher values indicate better results and F-Scores for the best performing model across each corpus are highlight in bold. The baseline experiment F-Scores demonstrate that both models perform better when run on larger datasets with Stanza EN outperforming NLTK EN. The average F-Score across the larger datasets, the Khan SAE Text Corpus and the News Articles Corpus, is 0.69 compared to the average across the subset datasets of 0.62. This result is expected as the smaller datasets in this experiment contain a higher proportion of words unique to South African English that are tagged incorrectly or not recognised by the models. Both models are observed to incorrectly tag some words and exclamations that are unique to SAE. Some examples are interjections such as *Heer* 'a title', *Jislaaik* 'delight' [19] and adjectives such as *Hell* 'hell', *lekker* 'nice' [19], all of which are tagged as nouns by both models. In addition, a number of words are tagged as 'Foreign Words' (not recognised by the models). Some examples are *abelungo* 'white people', *bloem* 'flower', *alle* 'all', *een* 'one', *imali* 'money', *neh* 'isn't that so', *op* 'on', *ou* 'guy or old', *skiet* 'shoot', *voor* 'in front' [19].

Table 4.1: Baseline experiment results

Model	**Corpus**	**Precision** *(Equation 1)*	**Recall** *(Equation 2)*	**Accuracy** *(Equation 4)*	**Specificity** *(Equation 5)*	**F-Score** *(Equation 3)*
NLTK EN	Khan SAE Text Corpus	0.9988	0.5293	0.5290	0.0379	0.6920
NLTK EN	News Articles Text Corpus	0.9996	0.5187	0.5186	0.2000	0.6830
Stanza EN	Khan SAE Text Corpus	0.9993	0.5450	0.5449	0.3620	**0.7053**
Stanza EN	News Articles Text Corpus	0.9997	0.5238	0.5237	0.2500	**0.6874**
NLTK EN	Khan DSAE subset	0.9994	0.4497	0.4496	0.0000	0.6203
NLTK EN	News Articles DSAE subset	0.9997	0.4719	0.4718	0.0000	**0.64110**
Stanza EN	Khan DSAE subset	0.9992	0.4527	0.4526	0.0075	**0.6231**
Stanza EN	News Articles DSAE subset	1.0000	0.4288	0.4288	N/A	0.6002

4.1.2. E-1-modification Results

Table 4.2 summarises the results of the Stanza EN_AF model predictions for the experiment run on each corpus and the best results are highlighted in bold. The inclusion of Afrikaans in the Stanza model only delivers a marginal improvement in average F-Scores compared to the baseline across larger datasets studied. It is observed that on the larger datasets (Khan SAE Corpus and News Articles Text Corpus), the average F-Score of Stanza EN_AF is 0.6967 compared to the baseline average of 0.6919 across NLTK EN and Stanza EN. The average F-Score of NLTK EN is 0.6875 and Stanza EN is 0.6964 so it is observed that on larger datasets Stanza EN and Stanza EN_AF outperform NLTK EN. Across the smaller datasets, the Khan DSAE subset and the News Articles DSAE subset, it is observed that NLTK EN performs best with an average F-Score of 0.6307 compared to the average F-Scores of Stanza EN and Stanza EN_AF being 0.6117 and 0.6113 respectively. The inclusion of Afrikaans therefore, does not improve the POS tagging accuracy on the smaller datasets studied. The average Precision, Recall and Accuracy of Stanza EN_AF across all datasets is comparable to the average scores of NLTK EN and Stanza EN. Specificity is observed to be the lower in the Stanza EN_AF model compared to the baseline. The average specificity of Stanza EN_AF across all datasets is 0.0684 compared to the average of 0.0595 of NLTK EN and 0.2065 of Stanza EN.

Table 4.2: Stanza EN_AF experiment results

Model	**Corpus**	**Precision** *(Equation 1)*	**Recall** *(Equation 2)*	**Accuracy** *(Equation 4)*	**Specificity** *(Equation 5)*	**F-Score** *(Equation 3)*
Stanza EN_AF	Khan SAE Text Corpus	0.9981	0.5454	0.5449	0.1221	**0.7053**
Stanza EN_AF	News Articles Text Corpus	0.9994	0.5247	0.5245	0.1429	**0.6881**
Stanza EN_AF	Khan DSAE subset	0.9980	0.4530	0.4526	0.0086	0.6231
Stanza EN_AF	News Articles DSAE subset	0.9993	0.4281	0.4280	0.0000	0.5995

F-Scores calculated from the training process of the Stanza SAE model are summarised in Table 4.3. F-Scores from the training process are expected to be high as the Stanza model has been designed to adapt to a variety of languages and datasets [56]. The F-Score of the POS Tagger indicates that the model has learned the UPOS tags of the words in the training dataset with an accuracy of 94.65%. The model has learned to correctly tokenize the words and sentences in the training dataset with near perfect accuracy.

Table 4.3: Model training results

Processor	**Metric**	**F-Score**
POS Tagger	UPOS	0.9465
Tokenizer	Tokens	1
Tokenizer	Sentences	0.9964
Tokenizer	Words	1

Table 4.4 summarises the results of the Stanza SAE model predictions for each corpus and Figure 4.1 illustrates the F-Scores across all models for each dataset. F-Scores demonstrate that the Stanza SAE model delivers the best results. In the largest dataset, the Khan SAE Text Corpus, a marginal improvement is observed but when the model is run on smaller datasets that contain higher proportions of previously unknown words or incorrectly tagged words due to the sample size being smaller, larger improvements are observed. This is expected as the training data contained SAE words. Stanza SAE produced an average F-Score of 0.76605 and performs particularly well on words in the DSAE, with an F-Score of 0.8599 on the Khan DSAE subset and News Articles DSAE subset datasets.

Dutch was observed in the Khan corpus, for example *bloem* ‘flower’ and the Stanza SAE model was able to correctly tag the word *bloem* as a noun whereas the NLTK EN, Stanza EN

and Stanza EN_AF models tagged *bloem* as a Foreign Word. This is an interesting observation as the Stanza SAE model was trained to learn the word *bloem* from the DSAE data and is now able to tag the word correctly. This suggests that Stanza SAE can be further improved by including Dutch as a language. Another example of where the Stanza SAE model improved on tagging SAE is the word *masala* 'blended spice' [19] used in context "35ml (3 tbsp) roast masala", NLTK tagged it as punctuation and Stanza SAE as a noun. In addition, the word *mense* 'people' [19] used in context "Mom uses garden to uplift mense" was tagged correctly as a noun by the Stanza SAE and adjective by NLTK EN. This sentence was however not well formed, missing a full stop, as it was extracted from the heading of an online article. The tokenizers therefore, did not separate this sentence but joined it to two successive sub-heading sentences which were also missing full stops, passing one long sentence for POS tagging and Stanza SAE was able to correctly tag the word. This suggests that there is a difference between NLTK and Stanza in handling longer sentences. NLTK's Averaged Perceptron's feature representations may not adequately capture context of unknown words in longer sentences but the long short-term nature of the BLSTM model used in Stanza can improve on this. This argument is supported by the observation that when the word *mense* is used in a well-formed sentence for example "The station is also urging mense to support their", NLTK tagged it correctly as a noun.

Across all models, the percentage of words tagged as foreign (unknown) has been observed to be low, on average 0.05% of the combined corpus is tagged as foreign. These words are observed to be place, event and business names, web site links, twitter handles, people names, special characters, Afrikaans words and words in other languages. Some examples are '#Mzansi', '…#', '_anything_', 'pic.twitter.com', 'portElizabeth', 'signore', 'SkeemSaam' and 'sabc'. Words that are have incorrect POS tags predicted by the models are observed to be names and places, for example 'ABInBev', 'Afrihost', 'Banyana'. Words combined with symbols such as '+China' and '1950s','-70°C' or '-19' and combinations of punctuation marks such as '!!' have predicted POS tags different to the evaluation dataset POS tags. One explanation is that this could be attributed to the tokenization process, where symbols or punctuation are not being separated from the words or numbers. This can be improved by applying additional pre-processing steps on the data to separate the symbols such as *'-'* from their use in the words '-19'. This finding does highlight a shortcoming of a data-driven approach for assigning POS tags for the training and evaluation datasets. Context of the word is not always taken into account and this leads to errors in evaluating the models' accuracy when dealing with disambiguation. An example is the word *'*doctor', used in a sentence "was

there so the doctor must have made" [16], the Stanza SAE model assigns the POS tag 'NOUN' but the word 'doctor' is assigned the POS tag 'VERB' in the evaluation data. Another example is the word 'kinder', used in English as an adjective and tagged correctly by the baseline models but when Afrikaans and SAE models were run, the word was tagged incorrectly as a noun when used as an adjective in a sentence. The Stanza SAE model is trained on 'kinder' as a noun because the word exists in the DSAE, derived from Afrikaans and Dutch (*kinder* and *kinderen* 'children and children' [19]).

These results demonstrate that by including the Afrikaans language and training a tokenizer and POS tagger on words in the DSAE, improvements to the accuracy of POS tagging models for SAE can be made.

Table 4.4: Stanza SAE model results

Model	**Corpus**	**Precision** *(Equation 1)*	**Recall** *(Equation 2)*	**Accuracy** *(Equation 4)*	**Specificity** *(Equation 5)*	**F-Score** *(Equation 3)*
Stanza SAE	Khan SAE Text Corpus	0.9994	0.5462	0.5460	0.0267	0.7063
Stanza SAE	News Articles Text Corpus	0.9993	0.5894	0.5891	0.0000	0.7415
Stanza SAE	Khan DSAE subset	0.9996	0.6084	0.6083	0.0109	0.7564
Stanza SAE	News Articles DSAE subset	1.0000	0.7543	0.7543	N/A	0.8599

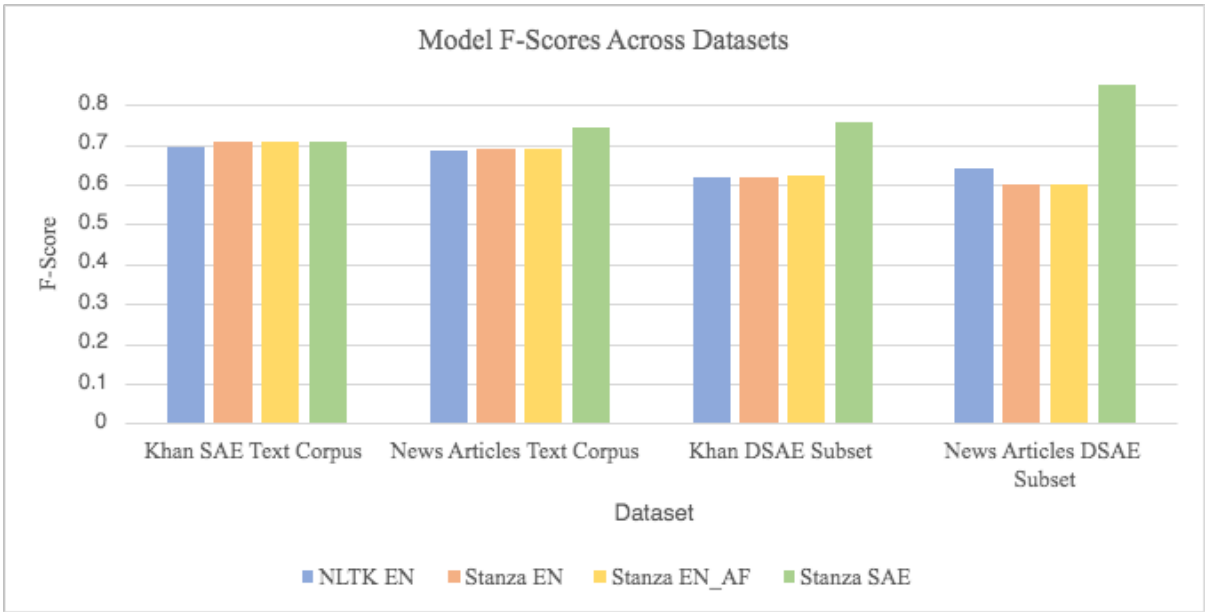

Figure 4.1: POS tagging results

4.2. Experiment 2: Donor-Adopter Relationships

Table 4.5 reports the observations of words that are common between the SAE Text corpora and evaluation data used in this dissertation. It is observed that 25.57% of the combined SAE corpus data studied contains words that are used in other languages, based on the evaluation data used. 25.44% are from or used by other South African languages with the three biggest contributors being Afrikaans, Sesotho and isiZulu, cumulatively making up 21.77%. Words from the DSAE contribute 3.25%. Other South African languages being Sepedi, isiNdebele, isiXhosa, Xitsonga, Tshivenda, Siswati and Khoisan cumulatively make up 0.41% and include words such as *Tsitsikamma* 'a location in the southern region of South Africa' and *Phalaborwa* 'a town in the north-eastern region of South Africa'. Other languages observed contribute 0.13% with words such as *sushi* 'rice wrapped in seaweed', *ninja* 'martial arts', *tapas* 'an appetizer' [53]. The biggest single contributor is Afrikaans and examples of words that are observed to be used in both SAE and Afrikaans are *videos* 'videos', *zululand* 'the area spanning KwaZulu-Natal in South Africa', *wakker* 'awake', *baba* 'father in isiZulu or baby in Afrikaans', *brink* 'boundary', *bakkie* 'a small container or a pick-up vehicle', *braai* 'barbeque' [19]. The word *braai* observed in SAE as "The beef ribs are *thumbs up* and so are the lamb chops, especially on the braai!" [34]. In Afrikaans, an example of the word *braai* used in a sentence is observed as "Fase 1 behels die bou van 'n publieke swembad, inrig van braai fasiliteite vir dagbesoekers en/of kampeerders asook die bou van die ingangshek." [55] 'Phase

1 entails the building of a public swimming pool, installation of barbecue facilities for day visitors and or campers as well as the building of an entrance gate.'. An example of an English word being used in Afrikaans is the word 'boyfriend', observed in SAE as "My boyfriend and I agreed we wouldn't exchange gifts" [34] and in Afrikaans as "dis nou Ant Stienie se loseerder en Lejanatjie se boyfriend" [55] 'this is aunty Stienie's lodger and Lejanatjie's boyfriend'. Sesotho is observed to contribute 4.98% and examples of words are *Sisonke* 'used as a placename'*, tse* 'the'*, tate* 'father' [24, 65]. It is also observed that a number of these words are names for example *Tokelo, Tau, Sisulu*. IsiZulu contributes 1.09%, with words such as *indaba* 'a meeting'*, isibaya* 'a fold'*, induna* 'an officer with oversight over a district'*, wena* 'singular pronoun for a second person' [19] observed. These observations demonstrate the existence of donor-adopter relationships between SAE in the Corpora studied and other South African languages.

Table 4.5: Donor-adopter results from SAE Text corpora and evaluation data

No donor-adopter relationships observed	Afrikaans	Sesotho	DSAE Words	isiZulu	Other South African Languages	Non-South African Languages
74.43%	15.70%	4.98%	3.25%	1.09%	0.41%	0.13%

5. Discussion

This chapter discusses the results of experiments conducted.

5.1. Experiment 1: Part of Speech (POS) Tagging South African English

This dissertation produces a set of baseline results for POS tagging accuracy on SAE text corpora, by measuring the accuracy of the NLTK (NLTK EN) and Stanza (Stanza EN) English language POS tagging models on these corpora. Stanza outperforms NLTK on larger datasets and on longer sentences in terms of POS tagging accuracy. One explanation is that the BLSTM Neural Network, which is used by Stanza, is able to capture long term dependencies better than NLTK EN. This finding about BLSTM's ability to capture long term dependencies is consistent with results reported by Kadari *et al.* [32]. On smaller datasets NLTK EN marginally outperforms Stanza EN. The requirement to model long term dependencies diminishes as the dataset decreases in size.

The first hypothesis postulated in this dissertation, is that modifying a POS tagging model by including the Afrikaans language improves POS tagging accuracy of SAE. Stanza EN_AF marginally outperforms Stanza EN and NLTK EN across larger datasets. On smaller datasets, Stanza EN_AF outperforms Stanza EN but not NLTK EN. The improvement that Stanza EN_AF makes to accuracy, however, is small and the first hypothesis is therefore not proven to be true.

The second hypothesis postulated in this dissertation, is that the accuracy of POS tagging models on SAE can be improved by training a POS tagging model on a dataset of English words unique to South Africa, represented by words in the DSAE. The Stanza SAE model delivers the highest accuracy across all datasets, compared to NLTK EN, Stanza EN and Stanza EN_AF. The highest accuracy is observed on smaller datasets. This is expected as these datasets contain higher proportions of words in the DSAE. The Stanza SAE model thus produces more accurate POS tagging predictions on SAE Text corpora, compared to off-the-shelf POS tagging models. Hypothesis 2 is therefore proven to be true.

A shortcoming of a data-driven approach for generating evaluation data, is that the context of the word in the evaluation dataset can be different to the context of the same word being used in the corpus under analysis. Another challenge with this approach is that some words are ambiguous and the evaluation dataset could only capture one of meanings. To improve on these

shortcomings, the data-driven approach to gathering POS tags needs to be augmented with a linguistic review, performed manually, to assign POS tags based on context and meaning in the sentence.

Whilst this dissertation has demonstrated that POS taggers can be improved for SAE, the accuracy of the baseline (Stanza EN and NLTK EN) and modified models (Stanza EN_AF and Stanza SAE) for SAE are still significantly below what we would expect when compared to other POS studies, such as the study by van Rooy and Schäfer [70]. It is acknowledged that one of the reasons for the poor performance is that each SAE word in the training dataset is exposed to a limited context of use as only one sentence is constructed for each SAE word. Evaluating the accuracy using a data-driven approach, is challenging, as word context and meaning are not always captured. One possible approach to improve the accuracy can be to include more linguistically labelled SAE corpora in the training data that includes word disambiguation, interjections and phrases unique to SAE. Another approach is to use existing training datasets such as the UD English treebanks[22] and substitute words in the datasets with SAE words that have the same POS tags. This will expose the model to more variations of context and ambiguity in the use of SAE words.

5.2. Experiment 2: Donor-Adopter Relationships

The third hypothesis postulated in this dissertation, is that SAE contains words adopted from other South African languages that can be observed using a data-driven approach. We expect donor-adopter relationships to be observed between SAE and other South African languages due to the influence that South Africa's multilingual society has had on SAE [42, 50, 51]. The results from the data-driven matching model confirm this expectation, uncovering relationships between SAE and Afrikaans, Sesotho and isiZulu. The benefit of the data-driven approach is that it is able to uncover these relationships across datasets of varying sizes and will thus be efficient to adopt this approach on large text corpora. One challenge that this approach faces, however, is that it is unable to determine if a word with the same spelling in two or more languages has the same meaning. In the scenario where the word has different meanings, it will identify a donor-adopter relationship which is not true. Further research can be done to improve on this shortcoming.

[22] https://universaldependencies.org/#english-treebanks

6. Conclusion

The first question that this dissertation answers, is to determine how accurate existing POS tagging models are on SAE text corpora. To answer this question, two off-the-shelf POS tagging models, NLTK's Averaged Perceptron (NLTK EN) and Stanza's English language (Stanza EN) POS tagging models are run on two SAE text corpora to establish baseline results. Furthermore, subsets of each corpus consisting of words found in the DSAE are examined to determine POS tagging accuracy of words unique to SAE. These baseline results show the average F-Score for NLTK EN across the SAE text corpora to be 0.6875 and for Stanza EN to be 0.6964. F-Scores calculated across subsets of the corpora show the average F-Score for NLTK EN to be 0.6307 and Stanza EN to be 0.6117. The second question that this dissertation answers is to determine if the accuracy of POS tagging models can be improved by including the Afrikaans language and by training a model on words in the DSAE. To answer this question, the Stanza EN model is modified to include the Afrikaans language, creating the Stanza EN_AF model and then the Stanza EN_AF model is then trained on words in the DSAE, creating the Stanza SAE model. The average F-Score across the SAE text corpora for the Stanza EN_AF model is 0.6967 and for the Stanza SAE model it is 0.7239. The average F-Score across the subsets of the corpora for the Stanza EN_AF model is 0.6113 and for the Stanza SAE model it is 0.8082. Including the Afrikaans language results in only a small improvement in accuracy. Training a model on words in the DSAE does improve POS tagging model accuracy although accuracy varies across datasets and the best improvements are observed with datasets that contain higher proportions of words also found in the DSAE. The novelty of the Stanza SAE model is that it has been trained on words found in the DSAE and this contributes to improving existing POS tagging models that are trained on British, American, Canadian, Australian and New Zealand English. The third question that this dissertation answers, is to determine if a data-driven approach can be used to identify donor-adopter relationships in text data between SAE and other South African languages. The Data-Driven Matching model developed for this purpose reveals that of the SAE text corpora studied, 15.7% of words are also exist in Afrikaans, 4.98% in Sesotho, 1.09% in isiZulu and 0.41% in other South African languages. 3.25% of the words exist in the DSAE. This demonstrates that a data-driven approach uncovers evidence of donor-adopter relationships between South African English other South African languages, the novelty being that this approach can be used on large datasets and corpora.

Future work can look at improving the accuracy of POS tagging SAE. One approach is to include more variations of sentence forms in the training datasets beyond the *Subject-Verb-Object* structure for words in the DSAE. Another approach is to create a larger training corpus for SAE that is tagged manually, using a linguistic approach, so that the context of word use can be better captured. Extracting data from the internet, for example a news article on a web page, can result in sentences that are not well formed, such as missing full stops. This creates a challenge with sentence identification during the tokenization process and further research can be done on pre-processing this data to correctly identify such sentences before passing the text data into the POS tagging model.

7. References

[1] AlKhwiter, W. and Al-Twairesh, N. Part-of-speech tagging for Arabic tweets using CRF and Bi-LSTM. *Computer Speech & Language*, 65: 101138, 2020. Retrieved February 17, 2021. doi: 10.1016/j.csl.2020.101138

[2] Asghar, M.Z., Khan, A., Ahmad, S., Khan, I.A. and Kundi, F.M. A Unified Framework for Creating Domain Dependent Polarity Lexicons from User Generated Reviews. *PLOS One*, 10 (10): e0140204-e0140204, 2015. Retrieved November 8, 2020. doi: 10.1371/journal.pone.0140204

[3] Asghar, M.Z., Khan, A., Ahmad, S., Qasim, M. and Khan, I.A. Lexicon-enhanced sentiment analysis framework using rule-based classification scheme. *PLOS One*, 12 (2): e0171649-e0171649, 2017. Retrieved December 8, 2022. doi: 10.1371/journal.pone.0171649

[4] Banko, M. and Moore, R.C. Part of Speech Tagging in Context. In *Proceedings of the 20th International Conference on Computational Linguistics.* (Geneva, Switzerland 2004), Association for Computational Linguistics, 556–561. doi: 10.3115/1220355.1220435

[5] Bekker, I. The Formation of South African English: a re-evaluation of the role of Johannesburg in the history of South African English. *English Today*, 29 (1): 3-9, 2013. Retrieved November 9, 2021. doi: 10.1017/S0266078412000454

[6] Bird, S., Klein, E. and Loper, E. *Natural Language with Python*. O'Reilly Media, Inc., Sebastopol, California, United States, 2009.

[7] Botha, W., van Rooy, B. and Coetzee-Van Rooy, S. Researching South African Englishes. *World Englishes*, 40 (1): 2-11, 2021. Retrieved Sepemtber 25, 2023. doi: 10.1111/weng.12468

[8] Botha, Y. Corpus evidence of anti-deletion in Black South African English noun phrases: Testing the extent to which Black South Africans restore elements of the

underlying structure of English noun phrases. *English Today*, 29 (1): 16-21, 2013. Retrieved Novermber 9, 2021. doi: 10.1017/S0266078412000478

[9] Brants, T. TnT - A Statistical Part-of-Speech Tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000* (Seattle, Washington, USA, 2000), Association for Computational Linguistics, doi: 10.48550/arxiv.cs/0003055

[10] Bravo-Marquez, F., Frank, E. and Pfahringer, B. Building a Twitter opinion lexicon from automatically-annotated tweets. *Knowledge-Based Systems*, 108: 65-78, 2016. Retrieved November 9, 2021. doi: 10.1016/j.knosys.2016.05.018

[11] Brill, E. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics*, 21 (4): 543-565, 1995.

[12] Carlberger, J. and Kann, V. Implementing an Efficient Part-of-Speech Tagger. *Software, Practice & Experience*, 29 (9): 815-832, 1999. Retrieved September 9, 2023. doi: 10.1002/(SICI)1097-024X(19990725)29:9<815::AID-SPE256>3.0.CO;2-F

[13] Carneiro, H.C.C., França, F.M.G. and Lima, P.M.V. Multilingual part-of-speech tagging with weightless neural networks. *Neural Networks*, 66: 11-21, 2015. Retrieved November 9, 2021. doi: 10.1016/j.neunet.2015.02.012

[14] Charniak, E., Hendrickson, C., Jacobson, N. and Perkowitz, M. Equations for Part-of-Speech Tagging. In *Proceedings of the National Conference on Artificial Intelligence* (Orlando, Florida, USA, 1999), Association for the Advancement of Artificial Intelligence, 784-789.

[15] Chen, W., Zhang, M., Zhang, Y. and Duan, X. Exploiting meta features for dependency parsing and part-of-speech tagging. *Artificial Intelligence*, 230: 173-191, 2016. Retrieved November 9, 2021. doi: 10.1016/j.artint.2015.09.002

[16] The Daily Voice. DailyVoice, (2021). Retrieved November 13, 2021 from Independent Media (Pty) Ltd: https://www.dailyvoice.co.za

[17] de Marneffe, M.-C., Manning, C.D., Nivre, J. and Zeman, D. Universal Dependencies. *Computational Linguistics*, 47 (2): 255-308, 2021. Retrieved October 28, 2023. doi: 10.1162/coli_a_00402

[18] Dictionary Unit for South African English. 2022. Dictionary of South African English. Retrieved March 8, 2022 from https://dsae.co.za/filter

[19] Dictionary of South African English. Dictionary Unit for South African English, (N.D.). Retrieved February 21, 2021 from Dictionary Unit for South African English: https://dsae.co.za/

[20] Ding, C., Utiyama, M. and Sumita, E. NOVA: A Feasible and Flexible Annotation System for Joint Tokenization and Part-of-Speech Tagging. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18 (2): 1-18, 2019. Retrieved November 9, 2021. doi: 10.1145/3276773

[21] du Plessis, A. and Van Niekerk, T. Adapting a Historical Dictionary for the Modern Online User: The Case of the Dictionary of South African English on Historical Principles's Presentation and Navigation Features. In *Twenty-first Annual International Conference of the African Association for Lexicography.* (Tsaneen, South Africa, 2016), Lexikos, 82-102.

[22] du Plessis, D., Bekker, I. and Hickey, R. Regionality in South African English. In. Cambridge University Press, 2019, 74-100.

[23] Ehsan, T., Khalid, J., Ambreen, S., Mustafa, A. and Hussain, S. Improving Phrase Chunking by using Contextualized Word Embeddings for a Morphologically Rich Language. *Arabian Journal for Science and Engineering*, 47 (8): 9781-9799, 2022. Retrieved September 27, 2023. doi: 10.1007/s13369-021-06343-7

[24] Eiselen, R. Government Domain Named Entity Recognition for South African Languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation* (Portorož, Slovenia, 2016), European Language Resources Association (ELRA), 3344-3348.

[25] Eiselen, R. South African Language Resources: Phrase Chunkers. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation* (Portorož, Slovenia, 2016), European Language Resources Association (ELRA), 689–693.

[26] Forsati, R. and Shamsfard, M. Novel harmony search-based algorithms for part-of-speech tagging. *Knowledge and Information Systems*, 42 (3): 709-736, 2015. Retrieved November 9, 2021. doi: 10.1007/s10115-013-0719-6

[27] Goldwater, S. and Griffiths, T.L. A Fully Bayesian Approach to Unsupervised Part-of-Speech Tagging. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics* (Prague, Czech Republic, 2007), Association for Computational Linguistics, 744-751.

[28] The Mail & Guardian. Guardian, M., (2021). Retrieved November 13, 2021 from M&G Online PTY Ltd: https://mg.co.za/section/opinion/

[29] Hartmann, D. and Zerbian, S. Rhoticity in Black South African English - a Sociolinguistic Study. *Southern African Linguistics and Applied Language Studies*, 27 (2): 135-148, 2010. Retrieved November 9, 2021. doi: 10.2989/SALALS.2009.27.2.2.865

[30] A Good Part-of-Speech Tagger in about 200 Lines of Python. Honnibal, M., (2013). Retrieved September 29, 2023 from ExplosionAI GmbH: https://explosion.ai/blog/part-of-speech-pos-tagger-in-python

[31] spaCy. Honnibal, M. and Montani, I., (2016). Retrieved September 30, 2023 from ExplosionAI GmbH: https://spacy.io

[32] Kadari, R., Zhang, Y.U., Zhang, W. and Liu, T. CCG supertagging with bidirectional long short-term memory networks. *Natural Language Engineering*, 24 (1): 77-90, 2018. Retrieved November 9, 2021. doi: 10.1017/S1351324917000250

[33] Kanakaraddi, S.G. and Nandyal, S.S. Survey on Parts of Speech Tagger Techniques. In *Proceedings of the 2018 International Conference on Current Trends towards*

Converging Technologies (ICCTCT) (Coimbatore, Tamil Nadu, India, 2018), IEEE, 1-6. doi: 10.1109/ICCTCT.2018.8550884

[34] Khan, U.M. 2020. *Building a South African English Corpus*. BSc (Honours) Computer Science thesis. Department of Computer Science, University of Cape Town, Cape Town, South Africa. https://projects.cs.uct.ac.za/honsproj/cgi-bin/view/2020/badenhorst_khan.zip/corpus.html

[35] Kruger, H. and van Rooy, B. Syntactic and pragmatic transfer effects in reported-speech constructions in three contact varieties of English influenced by Afrikaans. *Language Sciences*, 56: 118-131, 2016. Retrieved September 21, 2023. doi: 10.1016/j.langsci.2016.04.003

[36] KÜBler, S. and Mohamed, E. Part of speech tagging for Arabic. *Natural Language Engineering*, 18 (4): 521-548, 2012. Retrieved November 9, 2021. doi: 10.1017/S1351324911000325

[37] Lafferty, J.D., McCallum, A. and Pereira, F.C.N. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning* (Williamstown, MA, USA, 2001), Morgan Kaufmann Publishers Inc., 282–289.

[38] Liang, Y., Yang, M., Zhu, J. and Yiu, S.M. Out-domain Chinese new word detection with statistics-based character embedding. *Natural Language Engineering*, 25 (2): 239-255, 2019. Retrieved November 9, 2021. doi: 10.1017/S1351324918000463

[39] Lim, K. and Park, J. Part-of-Speech Tagging Using Multiview Learning. *IEEE Access*, 8: 195184-195196, 2020. Retrieved November 9, 2021. doi: 10.1109/ACCESS.2020.3033979

[40] Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J. and McClosky, D. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System*

Demonstrations (Baltimore, Maryland, USA, 2014), Association for Computational Linguistics, 55-60. doi: 10.3115/v1/P14-5010

[41] Merialdo, B. Tagging English Text with a Probabilistic Model. *Computational Linguistics*, 20 (2): 155-171, 1994.

[42] Mesthrie, R. Standardisation and variation in South African English. *Stellenbosch Papers in Linguistics Plus*, 26: 181-201, 1994. Retrieved September 25, 2023. doi: 10.5842/26-0-128

[43] Mesthrie, R. Children in language shift: The syntax of fifth-generation, pre-school Indian South African English speakers. *Southern African Linguistics and Applied Language Studies*, 21 (3): 119-126, 2003. Retrieved September 25, 2023. doi: 10.2989/16073610309486335

[44] Mesthrie, R. Undeletions in Black South African English. *Stellenbosch Papers in Linguistics Plus*, 34 (1): 75-99, 2006. Retrieved September 26, 2023. doi: 10.5842/34-0-30

[45] Mesthrie, R. Where does a new English dictionary stop? On the making of the Dictionary of South African Indian English: An account of the authorial decisions in compiling the Dictionary of South African English. *English Today*, 29 (1): 36-43, 2013. Retrieved December 6, 2022. doi: 10.1017/S026607841200048X

[46] Mjaria, F. and Keet, C.M. A Statistical Approach to Error Correction for isiZulu Spellcheckers. In *IST-Africa 2018 Conference* (Gabarone, Botswana, 2018), IEEE, 1-9.

[47] Munoz-Valero, D., Rodriguez-Benitez, L., Jimenez-Linares, L. and Moreno-Garcia, J. Using Recurrent Neural Networks for Part-of-Speech Tagging and Subject and Predicate Classification in a Sentence. *International Journal of Computational Intelligence Systems*, 13 (1): 706-716, 2020. Retrieved November 9, 2021. doi: 10.2991/ijcis.d.200527.005

[48] Naseem, T., Snyder, B., Eisenstein, J. and Barzilay, R. Multilingual Part-of-Speech Tagging: Two Unsupervised Approaches. *The Journal of Artificial Intelligence Research*, 36: 341-385, 2009. Retrieved November 9, 2021. doi: 10.1613/jair.2843

[49] Nguyen, D.Q., Nguyen, D.Q., Pham, D.D. and Pham, S.B. A robust transformation-based learning approach using ripple down rules for part-of-speech tagging. *AI Communications*, 29 (3): 409-422, 2016. Retrieved November 9, 2021. doi: 10.3233/AIC-150698

[50] Olayinka, U.F. Afrikaans discourse-pragmatic features in South African English. *Lingua*, 272: 103309, 2022. Retrieved September 21, 2023. doi: 10.1016/j.lingua.2022.103309

[51] Pienaar, L. and de Klerk, V. Towards a Corpus of South African English: Corralling the Sub-varieties. In *Thirteenth International Conference of the African Association for Lexicography.* (Stellenbosch, South Africa, 2009), Lexikos, 353-371. doi: 10.4314/lex.v19i1.49135

[52] Princeton University. 2016. WordNet 3.1 database files. Retrieved 12 March 2022 from https://wordnetcode.princeton.edu/wn3.1.dict.tar.gz

[53] WordNet Search - 3.1. Princeton University, (2023). Retrieved October 30, 2023 from Princeton University: http://wordnetweb.princeton.edu/perl/webwn

[54] Puttkammer, M., Schlemmer, M. and Bekker, R. Developing Text Resources for Ten South African Languages. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation* (Reykjavik, Iceland, 2014), European Language Resources Association (ELRA), 3698-3703.

[55] Puttkammer, M., Schlemmer, M., Pienaar, W. and Bekker, R. Developing Text Resources for Ten South African Languages. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation* (Reykjavik, Iceland, 2014), European Language Resources Association (ELRA), 3698-3703.

[56] Qi, P., Zhang, Y., Zhang, Y., Bolton, J. and Manning, C.D. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (Online, 2020), Association for Computational Linguistics, 101–108. doi: 10.48550/arxiv.2003.07082

[57] Rufus, H.G. Situating A Dictionary of South African English on Historical Principles within a More Comprehensive Lexicographic Process. *Lexikos*, 9: 269-282, 2021. Retrieved September 25, 2023. doi: 10.5788/9-1-926

[58] Schuld, M., Sinayskiy, I. and Petruccione, F. Simulating a perceptron on a quantum computer. *Physics Letters A*, 379 (7): 660-663, 2015. Retrieved September 30, 2023. doi: 10.1016/j.physleta.2014.11.061

[59] Senthil Kumar, N.K. and Malarvizhi, N. Bi-directional LSTM–CNN Combined method for Sentiment Analysis in Part of Speech Tagging (PoS). *International Journal of Speech Technology*, 23 (2): 373-380, 2020. Retrieved November 9, 2021. doi: 10.1007/s10772-020-09716-9

[60] Singh, V. and Parkinson, J. Stability of features of Black South African English. *Per Linguam*, 23 (2): 54-67, 2007.

[61] Model Training and Evaluation. Stanford NLP Group, (2022). Retrieved December 11, 2022 from Stanford NLP Group: https://stanfordnlp.github.io/stanza/training.html

[62] Tian, Y. and Lo, D. A Comparative Study on the Effectiveness of Part-of-Speech Tagging Techniques on Bug Reports. In *SANER 2015 22nd IEEE International Conference on Software Analysis, Evolution, and Reengineering* (Montreal, Canada, 2015), IEEE, 570-574. doi: 10.1109/SANER.2015.7081879

[63] Toutanova, K., Klein, D., Manning, C.D. and Singer, Y. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on*

Human Language Technology - Volume 1. (Edmonton, Canada, 2003), Association for Computational Linguistics, 173–180. doi: 10.3115/1073445.1073478

[64] Unuabonah, F. and Mtembu, N. Multilingual pragmatic markers in South African English. *Southern African Linguistics and Applied Language Studies*, 41 (3): 264-279, 2023. Retrieved September 25, 2023. doi: 10.2989/16073614.2022.2123366

[65] SADiLaR
ICELDA. 2023. Generic Multilingual Academic Wordlists with Definitions. Retrieved October 30, 2023 from https://repo.sadilar.org/handle/20.500.12185/666?show=full

[66] SADiLaR. 2015. The South African Directory Enquiries (SADE) Name Corpus. Retrieved December 10, 2022 from https://repo.sadilar.org/handle/20.500.12185/378

[67] van Rooy, B. An alternative interpretation of tense and aspect in Black South African English. *World Englishes*, 27 (3-4): 335-358, 2008. Retrieved September 26, 2023. doi: 10.1111/j.1467-971X.2008.00572.x

[68] van Rooy, B. A Multidimensional Analysis of Student Writing in Black South African English. *English World-Wide: A Journal of Varieties of English*, 29 (3): 268-305, 2008. Retrieved September 21, 2023. doi: 10.1075/eww.29.3.03van

[69] van Rooy, B. Corpus linguistic work on Black South African English: An overview of the corpus revolution and new directions in Black English syntax. *English Today*, 29 (1): 10-15, 2013. Retrieved September 26, 2023. doi: 10.1017/S0266078412000466

[70] van Rooy, B. and Schäfer, L. The effect of learner errors on POS tag errors during automatic POS tagging. *Southern African Linguistics and Applied Language Studies*, 20 (4): 325-335, 2002. Retrieved September 21, 2023. doi: 10.2989/16073610209486319

[71] van Rooy, B. and Wasserman, R. Do the Modals of Black and White South African English Converge? *Journal of English Linguistics*, 42 (1): 51-67, 2014. Retrieved September 7, 2023. doi: 10.1177/0075424213511463

[72] Viterbi, A. Error bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Transactions on Information Theory*, 13 (2): 260-269, 1967. Retrieved September 27, 2023. doi: 10.1109/TIT.1967.1054010

[73] Wasserman, R. and van Rooy, B. The Development of Modals of Obligation and Necessity in White South African English through Contact with Afrikaans. *Journal of English Linguistics*, 42 (1): 31-50, 2014. Retrieved September 21, 2023. doi: 10.1177/0075424213514588

[74] Westbury, C. and Hollis, G. Conceptualizing syntactic categories as semantic categories: Unifying part-of-speech identification and semantics using co-occurrence vector averaging. *Behavior Research Methods*, 51 (3): 1371-1398, 2019. Retrieved November 9, 2021. doi: 10.3758/s13428-018-1118-4

[75] Wiebesiek, L., Rudwick, S. and Zeller, J. South African Indian English: A qualitative study of attitudes. *World Englishes*, 30 (2): 251-268, 2011. Retrieved November 9, 2021. doi: 10.1111/j.1467-971x.2011.01709.x

[76] Wissing, D. Black South African English: A new English? Observations from a phonetic viewpoint. *World Englishes*, 21 (1): 129-144, 2002. Retrieved September 26, 2023. doi: 10.1111/1467-971X.00236

[77] Xue, X. and Zhang, J. Building Codes Part-of-Speech Tagging Performance Improvement by Error-Driven Transformational Rules. *Journal of Computing in Civil Engineering*, 34 (5): 4020035, 2020. Retrieved November 9, 2021. doi: 10.1061/(ASCE)CP.1943-5487.0000917

[78] Zhou, D., Zhang, Z., Zhang, M.-L. and He, Y. Weakly Supervised POS Tagging without Disambiguation. *ACM transactions on Asian and Low-Resource Language Information Processing*, 17 (4): 1-19, 2018. Retrieved November 9, 2021. doi: 10.1145/3214707