

Morphological segmentation of isiXhosa using unsupervised machine learning

L Mzamo

 **orcid.org/0000-0002-8867-7416**

Thesis accepted in fulfilment of the requirements for the degree
Doctor of Philosophy in Computer and Electronic Engineering at
the North-West University

Promoter: Prof ASJ Helberg

Co-Promoter: Prof SE Bosch

Graduation: December 2021

Student number: 24827304

Koomama bam, abamithandazo ndiphila yiyo.

Kootata bam, abangqibelelo ndimi ngayo.

Kubantwana bam bonke, oomoya namaza endiqhutywa ngawo.

Kulentlantsi ibubom, ibungenathuba endixheshwa bubo.

Kubacebisi balomzamo, abankolelo kum ndikhanyisa ngayo.

ABSTRACT

In this work the use of unsupervised machine learning in the morphological segmentation of Nguni languages, evaluated on isiXhosa, is advanced. The work researches, extends, implements and evaluates unsupervised machine learning techniques in the morphological segmentation of Nguni language, specifically isiXhosa.

IsiXhosa, one of the eleven South African official languages, is a Nguni language with 9.7 million mother-tongue speakers (17% of the South African population). The language is an agglutinating and synthetic language characterised by highly inflected words. Its nouns are characterised by a class system based on their prefixes. Verbs are characterised by a concatenation of numerous prefixes and suffixes - and their concordial agreement with the subject noun or object noun.

Two segmenters were developed using state of the art techniques in morphological segmentation: the IsiXhosa Branching Entropy Segmenter (XBES), using branching entropy; and the IsiXhosa Heuristic Maximum Likelihood Segmenter (XHMLS), using probabilistic language models.

The XBES branching entropy segmenter matched the $77.2 \pm 0.10\%$ accuracy of the benchmark (Morfessor-Baseline) with an accuracy of $77.4 \pm 0.32\%$ and outperforms the benchmark on its F1 Score of $48.9 \pm 0.75\%$ with a score of $58.0 \pm 0.10\%$. These accuracy results were achieved with a z-score normalised variation of branching entropy (N_zVBE) mode trained on 11-grams from a 1.5 million word corpus. The F1 Score was achieved using the unnormalised variation of branching entropy (VBE) technique trained with 9-grams from a 1.5 million word corpus.

The study established that the character level n-gram length that encapsulates the predictability of isiXhosa words to be ten (10), which corresponds to using 9-grams.

The XHMLS maximum likelihood segmenter has an accuracy of $75.2 \pm 0.18\%$, which did not meet the accuracy of the benchmark but outperformed both XBES and Morfessor-Baseline on the F1 Score with a score of $59.4 \pm 0.20\%$. The best performing mode of XHMLS was the independent affixes (XHMLS-IA) language model confirming that morphemic dependencies in isiXhosa are stronger within the affixes and weaker among the affix and that the roots in the language permeate many of the word categories.

Key Terms

Natural Language Processing, Human Language Technology, Unsupervised Machine Learning, Morphological Segmentation, IsiXhosa

ISISHWANKATHELO

Kulomsebenzi siqhubela phambili ukusetyenziswa koomatshini abangagadwanga kumba wocalulo-magama weelwimi zesiNguni, ngokuvavanya esiXhoseni. Umsebenzi uphanda, wandise, uphumeze kwaye uvavanye ubuchule bokufunda ngomatshini obungagadwanga kucalulo-magama lolwimi lwesiNguni, isiXhosa.

IsiXhosa, olunye lweelwimi ezisemthethweni zaseMzantsi Afrika, lulwimi lwesiNguni kwaye lune-9.7 yezigidi (i-17% yabemi boMzantsi Afrika) abasithetha ulwimi lwenkobe. Olulwimi lulwimi oluhlanganisayo noludityanisiweyo olubonakaliswa ngamagama aneempembelelo kakhulu. Izibizo zalo zibonakaliswa yinkqubo yamahlelo asekw kwisimaphambili. Izenzi zibonakaliswa kukunxibelelana kwezimaphambili ezininzi kunye nezimamva, kunye nesivumelwano sesivumelwano kunye nesibizo sentloko okanye isibizo sento.

Izicandi-magama ezibini sele ziphuhlisiwe kusetyenziswa ubuchwepheshe bobuchwepheshe kucando-magama: iIsiXhosa Branching Entropy Segmenter (XBES), esebenzisa i-entropy yesebe, kunye isiXhosa Heuristic Maximum Likelihood Segmenter (XHMLS), esebenzisa imizekelo yobukho yolwimi.

Isicandi-magama se-entropy, iXBES, sihambelana nokuchaneka kwe-77.2 $\pm$ 0.10% komgangatho, iMorfessor-Baseline, nge-77.4 $\pm$ 0.32% kunye nokuwugqwesa umgangatho kumanqaku eF1 angama-48.9 $\pm$ 0.75% ngamanqaku angu58.0 $\pm$ 0.10%. Ezi ziphumo zichanekileyo esiziphumeleleyo nge-z-score aqhelekileyo okwahluka kwe-entropy ye-sebe (iNzVBE) eqeqeshwe nge-11-gram kwi-1.5 yezigidi zamagama. Inqaku le-F1 lafezekiswa kusetyenziswa ukwahluka okungacwangciswa kwendlela ye-entropy yesebe (iVBE) eliqeqeshwe nge-9-gram yamagama eli-1.5 lesigidi.

Olu phando lubonise ukuba ubude be-n-gram obubandakanya konke ukuqikeleleka kwamagama esiXhosa lishumi (10), ahambelana nokusebenzisa i-9-gram kwi-entropy yesebe.

Imodeli yolwimi ekhokelwa yindalo, iXHMLS, ukuchana kwayo kwe-75.2 $\pm$ 0.18% akuhlangabezani nokuchana komgangatho kodwa kwazigqwesa zombini iXBES kunye neMorfessor-Baseline kumanqaku eF1 ngamanqaku angama-59.4 $\pm$ 0.20%. Eyona ndlela iqhuba kakuhle ye-XHMLS yayiyimodeli yolwimi ezimeleyo (iXHMLS-IA) eqinisekisa ukuba ukuxhomekeka kwezaki-magam kwisiXhosa kunamandla ngaphakathi kwezihlomelo kwaye kubuthathaka phakathi kwezihlomelo kwaye iingcambu kulwimi zigcwele kwindidi ngeendidi zamagama.

Amagama aphambili

Ukucalula uLwimi lweNdalo, iTekhnoloji yoLwimi loLuntu, ukuFunda koMatshini Ongagadwanga, uCalulo-Magama, isiXhosa.

TABLE OF CONTENTS

ABSTRACT	II
ISISHWANKATHELO	III
1. INTRODUCTION.....	1
1.1 Research Work	1
1.1.1 Research Aim	2
1.1.2 Research Question	2
1.1.3 Research Objective	2
1.2 Research Methodology.....	2
1.2.1 Literature Review	3
1.2.2 Investigation and Extension of the Identified Techniques	3
1.2.3 Evaluation	4
1.3 Publications.....	4
1.4 Thesis Structure.....	4
1.5 Conclusions	5
2. LITERATURE REVIEW	7
2.1 Segmentation Metrics.....	7
2.1.1 Basics of Prediction Measurement	8
2.1.2 Performance on Out of Vocabulary Words	10
2.1.3 Hypothesis Testing.....	11
2.1.4 Indirect Segmentation Metrics	11
2.1.4.1 Bilingual Evaluation Understudy (BLEU)	11

2.1.4.2	Word Error Rate	12
2.1.4.3	Perplexity	12
2.1.5	Direct Morphological Segmentation Metrics	13
2.1.5.1	Levels of Measuring Segmentation	13
2.1.5.2	Morpheme Boundary Identification measurement	14
2.1.5.3	Morpheme Identification measurements.....	14
2.1.5.4	Evaluation Metric for Morphological Analysis (EMMA).....	14
2.1.6	Conclusions on Segmentation Metrics	15
2.2	Knowledge/Rule-based techniques	15
2.2.1	Finite-State Automata.....	15
2.2.2	Logic Programming	17
2.2.3	Challenges of rule-based systems.....	17
2.3	Supervision in Machine Learning	18
2.4	Supervised Morphological Segmentation techniques.....	19
2.4.1	Rules lists.....	19
2.4.2	Hidden Markov Models.....	20
2.4.3	Maximum Entropy Modelling	21
2.4.4	Markov Random Fields	22
2.4.5	Support Vector Machines	26
2.4.6	Markov Chain Monte-Carlo based segmenters	27
2.4.7	Memory-Based learning	28
2.4.8	Neural Networks.....	29
2.4.9	Conclusions on Supervised Morphological Segmentation techniques	32

2.5	Semi-Supervised	33
2.6	Unsupervised Morphological Segmentation works	34
2.6.1	Branching Entropy.....	34
2.6.2	Maximum Entropy Learning.....	37
2.6.3	Minimum Description Length	38
2.6.4	Paradigmatic Clustering	40
2.6.5	Context based segmentation.....	41
2.6.6	Non-parametric Bayesian techniques	42
2.6.7	Neural Networks.....	45
2.6.8	Contrastive Estimation	46
2.6.9	Language Model based segmentation.....	47
2.6.10	Summary of Unsupervised Morphological Segmentation	50
2.7	Techniques and Benchmark of choice for isiXhosa segmentation	51
3.	INFORMATION THEORETIC FUNDAMENTALS	54
3.1	Entropy	54
3.1.1	Joint Entropy	55
3.1.2	Conditional Entropy and Residual entropy.....	56
3.1.3	Cross Entropy	57
3.1.4	Relative Entropy.....	58
3.1.5	Conditional Relative Entropy	59
3.2	Mutual Information	60
3.2.1	Conditional mutual information	61

3.3	Chain rule of entropy, relative entropy and mutual information.....	62
3.3.1	Chain Rule of Entropy	62
3.3.2	Chain Rule of Relative Entropy	63
3.3.3	Chain Rule of Mutual Information	64
3.4	Predictive metrics for stationary stochastic processes.....	66
3.4.1	Entropy Rate	67
3.4.2	Redundancy	68
3.4.3	Excess Entropy	69
3.4.4	Transient Information	69
3.4.5	Transfer Entropy	70
3.5	Predictive Information Metrics	70
3.6	Conclusions	72
4.	SEGMENTATION WITH BRANCHING ENTROPY.....	73
4.1	Branching Entropy.....	73
4.1.1	Morpheme Boundary Identification using Branching Entropy.....	74
4.1.2	Morpheme Identification using Branching Entropy.....	75
4.1.2.1	Wordhood from Bracing Joint Entropy.....	76
4.1.2.2	Wordhood from individual left- and right- branching entropies	77
4.1.2.3	Normalisation across substring sizes	78
4.2	Branching Entropy Metrics and Predictive Information	79
4.2.1	Branching Entropy and Entropy Rate	79
4.2.2	Variation of Branching Entropy and Predictive Gain	80

4.2.3	Normalisation and Predictive Information	81
4.2.4	Branching Entropy Inference and Predictive Information	83
4.2.4.1	Inference in BE and VBE.....	83
4.2.4.2	Inference in normalised BE and VBE	84
4.2.5	Conclusions on Branching Entropy Metrics and Predictive Information	85
4.3	Bi-directional Entropy Language Modelling.....	85
4.3.1	Un-smoothed bi-directional Branching Entropy language model flow	86
4.3.2	The modified Kneser-Ney smoothed bi-directional Branch Entropy language model.....	88
4.3.3	Summary of Bidirectional Entropy Language Models	89
4.4	Conclusions	90
5.	ISIXHOSA SEGMENTS	91
5.1	Surface Segmentation	91
5.2	Word Categories of isiXhosa.....	92
5.3	IsiXhosa Segmentation Specifics	93
5.3.1	Noun (Isibizo)	95
5.3.1.1	Noun Concords	97
5.3.2	Pronoun (Isimelabizo)	98
5.3.2.1	Absolute Pronoun (Esoqobo)	98
5.3.2.2	Demonstrative Pronouns (Isilathisi)	99
5.3.2.3	Quantitative Pronoun (Esoquko)	100
5.3.2.4	Differentiative Pronoun (Esokwahlula).....	101
5.3.2.5	Superlative Pronoun (Esobalulo).....	101

5.3.3	Qualificative (Isichazi)	102
5.3.3.1	Adjective (Isiphawuli).....	102
5.3.3.2	Relative (Isibaluli)	103
5.3.3.3	Enumerative (Esobalo).....	105
5.3.3.4	Possessive (Isimnini).....	105
5.3.4	Predicate (Isivisa).....	107
5.3.4.1	Verbs	107
5.3.4.1.1	Verb Extensions	107
5.3.4.1.2	Conjugation.....	108
5.3.4.1.3	Forms.....	108
5.3.4.1.4	Moods	109
5.3.4.1.5	Aspect.....	110
5.3.4.1.6	Tense.....	111
5.3.4.1.7	Conclusions on Verbs	112
5.3.4.2	Copulatives	112
5.3.4.2.1	Pronoun Derived Copulatives.....	112
5.3.4.2.2	Noun Derived Copulatives.....	113
5.3.5	Descriptive	114
5.3.6	Adverbs.....	115
5.3.7	Conjunctions	116
5.3.8	Interjections.....	116
5.3.9	isiXhosa NLP Surface Segments in Summary.....	116
5.4	Dependencies among affixes and morphemes.....	117

5.4.1	Intra-affix dependencies	117
5.4.2	Affix-root dependencies.....	118
5.4.3	Circumfixing	118
5.4.4	Summary on possible morpheme dependency.....	118
5.5	Conclusions	119
6.	ISIXHOSA LANGUAGE MODEL BASED SEGMENTATION	120
6.1	IsiXhosa Probabilistic Models.....	121
6.1.1	Morpheme Sequence model (MS).....	122
6.1.2	Independent Affixes (IA).....	122
6.1.3	Affix-Links (AL).....	123
6.1.4	Circumfix Link (CL).....	124
6.2	Training and Inference.....	125
6.3	Conclusions	125
7.	EVALUATION.....	127
7.1	Experimental Setup.....	127
7.1.1	Data Sources	128
7.1.1.1	Training Corpus.....	128
7.1.1.2	Testing Corpus.....	129
7.1.1.3	Conclusion on the corpora	131
7.1.2	Data Splits.....	131
7.1.2.1	Development, Validation and Evaluation Corpora	131
7.1.2.2	N-Fold Validation for unsupervised systems.....	132

7.1.2.3	Evaluation sets.....	133
7.1.3	Segmenters.....	133
7.1.4	Choice of benchmark unsupervised segmenter.....	135
7.1.5	Experiments conducted.....	135
7.1.6	Conclusions on Experimental Design	135
7.2	Results	136
7.2.1	Benchmark performance	136
7.2.2	IsiXhosa Branching Entropy Segmenter (XBES) Results	139
7.2.2.1	Overall performance of XBES	139
7.2.2.2	XBES overall performance trends	140
7.2.2.3	XBES Performance on OoV words.....	142
7.2.2.4	Conclusions on XBES Performance	144
7.2.3	Predictive information in isiXhosa.....	144
7.2.4	IsiXhosa Heuristics Maximum Likelihood Segmenter (XHMLS) performance	147
7.2.4.1	XHMLS Overall performance results	147
7.2.4.2	XHMLS out of vocabulary results	149
7.2.4.3	Conclusions of XHMLS results	151
7.2.5	Conclusions on Experiments Results	152
7.3	Summary.....	152
8.	CONCLUSIONS AND FUTURE WORK.....	153
8.1	Summary of the work.....	153
8.2	Contributions.....	154

8.3	Main Findings	155
8.3.1	Addressing Research Objectives	156
8.3.1.1	Findings on Objectives 1 and 2	156
8.3.1.2	Findings on Objective 3.....	156
8.3.2	Answering the Research Questions	157
8.3.3	Achievement of the Aim	158
8.4	Future work	158
8.5	Conclusions	158
BIBLIOGRAPHY		160
ANNEXURES A: GLOSSES USED		190

LIST OF TABLES

Table 5-1:	Noun Class Prefixes.....	95
Table 5-2:	Noun Basic Prefixes.....	96
Table 5-3:	Subject Concords.....	97
Table 5-4:	Object Concords	98
Table 5-5:	Absolute Pronouns.....	99
Table 5-6:	Quantitative Pronouns.....	100
Table 5-7:	List of Adjective Concords (Louw et al., 1984:77).....	102
Table 5-8:	Relative Concords (Pahl et al., 1989:685; Louw et al., 1984:84)	104
Table 5-9:	List of Possessive Concords (Louw et al., 1984:100)	106
Table 5-10:	Possessive Pronominal Stems (Pahl et al., 1989:690)	106
Table 5-11:	Verb Extension examples.....	107
Table 5-12:	Examples of different isiXhosa tenses.....	111
Table 5-13:	List of Copula (Louw et al., 1984:220)	113
Table 5-14:	Noun Derived Copulative Prefixes (Louw et al., 1984:220).....	114
Table 7-1:	Quantities of undesirable NCHLT corpus entries.....	130
Table 7-2:	Test Corpora words overlap	132
Table 7-3:	Benchmark Overall Boundary Identification Results with p-values	137
Table 7-4:	Benchmark OoV Boundary Identification Results with p-values.....	138
Table 7-5:	XBES Overall Boundary Identification Results	139
Table 7-6:	XBES OoV Boundary Identification Results.....	142
Table 7-7:	Best XHMLS Overall Boundary Identification Results per model.....	147

Table 7-8:	Best XHMLS OoV Boundary Identification Results per model	150
------------	--	-----

LIST OF FIGURES

Figure 2-1:	Graphical Representation of Precision and Recall Measures (Manning & Schütze, 1999:268)	8
Figure 2-2:	Graphical Representation of Out of Vocabulary words	10
Figure 2-3:	An example of a Directed Graphical model	22
Figure 2-4:	An example of an Un-directed Graphical Model	23
Figure 4-1:	Right Branching from a substring	73
Figure 4-2:	Location of a possible boundary in a substring	74
Figure 4-3:	A graphical representation of left and right attachment to a substring	76
Figure 4-4:	Representation of Reverse Key Transformation	86
Figure 4-5:	Representation of the Un-Smoothed BELM process	87
Figure 4-6:	Graphical representation of the mKN Smoothed BELM Process	89
Figure 5-1:	The hierarchy of IsiXhosa word categories we use	93
Figure 5-2:	Glossing Description	94
Figure 6-1:	Probabilistic graphical model of the Morpheme Sequence model	122
Figure 6-2:	Probabilistic graphical model of the Independent Affixes model	123
Figure 6-3:	Probabilistic graphical model of the Affix-Links model	124
Figure 6-4:	Probabilistic graphical model of the Cicumfix-Link model	124
Figure 7-1:	Training Corpus word frequency distribution	129
Figure 7-2:	Testing Corpus word frequency distribution	130
Figure 7-3:	Graphical representation of Source corpora	131
Figure 7-4:	Benchmark overall performance by training set size	137
Figure 7-5:	Benchmark OoV overall performance by training set size	138

Figure 7-6:	Average overall accuracy of the segmenters and the best BE and BJE modes by training set size.....	140
Figure 7-7:	Average overall F1 Score of the Random Segmenter, Morfessor-Baseline, and the best BE and BJE modes by training set size	141
Figure 7-8:	Average OoV Accuracy of the Random Segmenter, Morfessor-Baseline, and the bests BE and BJE modes by training set size.....	143
Figure 7-9:	Average OoV F1 Score of the Random Segmenter, Morfessor-Baseline, and the bests BE and BJE modes by training set size.....	143
Figure 7-10:	Trend in Block Entropy by n-gram length	145
Figure 7-11:	Trend in Entropy Rate trend by n-gram length.....	145
Figure 7-12:	Trend in Predictive Gain by n-gram length	146
Figure 7-13:	Average overall accuracy of the best modes for the segmenters by training set size	148
Figure 7-14:	Average overall F1 Scores of the best modes for the segmenters by training set size	149
Figure 7-15:	Average OoV Accuracy of the best modes for the segmenters by training set size.....	150
Figure 7-16:	Average OoV F1 Scores of the best modes for the segmenters by training set size	151

ACRONYMS

ANN	Artificial Neural Network
ASR	Automatic speech recognition
BAMA	Buckwalter Arabic Morphological Analyser
BE	Branching Entropy
BELM	Branching Entropy Language Model
BJE	Bracing Joint Entropy
BLEU	Bilingual evaluation understudy
BN	Bayesian Network,
CER	Character error rate
CNN	Convolutional Neural Network
CRF	Conditional Random Field
DAG	Directed acyclic graphs
DBN	Dynamic Bayesian Network
DGM	Directed graphical models
DLG	Descriptive Length Gain
DNN	Deep Neural Network
DP	Dirichlet process
EMMA	Evaluation Metric for Morphological Analysis
ETRF	Extra Trees Random Forests
FN	False negative
FP	False positive
FSA	Finite-state machine/automata
HDP	Hierarchical Dirichlet process
HLT	Human language technology
HMM	Hidden Markov Model
HSMM	Hidden Semi-Markov Model
KL	Kullback-Leibler in KL Divergence
LSA	Latent Semantic Analysis
LSE	Letter Successor Entropy
LSTM	Long Short Term Memory
MAP	<i>Maximum a posteriori</i> (probabilistic condition)
MBI	Morpheme boundary identification
MBL	Memory-based learning
MCMC	Markov Chain Monte Carlo
MDL	Minimum Description Length.

MEM	Maximum Entropy Modelling
MEMM	Maximum Entropy Markov Model
mKN	modified Kneser Ney
ML	Maximum Likelihood
MSD	Morphosyntactic descriptions
MI	Mutual Information
MRF	Markov Random Fields
MRI	Morphological re-inflection
MRL	Morphologically rich language
NER	Named entity recognition
NLP	Natural language processing
NPYLM	Nested Pitman-Yor Language model
N_{μ} VBE	Mean normalised Variation of Branching Entropy
N_{μ} VBJE	Mean normalised Variation of Bracing Joint Entropy score
N_z VBE	Z-score normalised Variation of Branching Entropy
N_z VBJE	Z-score normalised Variation of Bracing Joint Entropy
OoV	Out of vocabulary
PCFG	Probabilistic Context Free Grammar
PER	Phoneme error rate
PGM	Probabilistic Graphical Model
PID	Partial Information Decomposition
POS	Part of speech
PYLM	Pitman-Yor Language model
RNN	Recurrent Neural Network
SADiLaR	South African Centre for Digital Language Resources
SVM	Support Vector Machines
TN	True negative
TP	True positive
UD	Universal Dependencies
UDM	Undirected graphical model
VBE	Variation of Branching Entropy
VBJE	Variation of Bracing Joint Entropy score
WER	Word Error rate
WFST	Weighted Finite-state Transducer
XGL	IsiXhosa lexicalised probabilistic Graphical Lemmatiser
XHMLS	IsiXhosa Heuristics Maximum Likelihood Segmenter

NOMENCLATURE

X	an uppercase letter represents a random variable
x	a lowercase letter represents an instance/outcome of a random variable
$x_{1..k} = x^k$ $= x_1, \dots, x_k$	a sequence of length k whose items are indexed with integers 1 to k
$x_{i..j}$	a subsequence of x containing the elements indexed from i to j
$x_{i:i+1}$	a boundary location of x between the elements indexed i and $i + 1$.
$\#x$	the occurrence count of x in a corpus.
$a \oplus b$	concatenation operator, concatenates lists a and b

1. INTRODUCTION

Word segmentation is a requisite natural language processing task for agglutinating and polysynthetic languages (Bender, 2013). The typology of agglutinating and polysynthetic languages glues a number of the smallest meaning bearing units, i.e. morphemes, in the generation of a word (Kosch, 2006; Bender, 2013). Word segmentation is the splitting of an orthographic word into morphemes. This is not a trivial task, especially when using raw data.

South African indigenous languages have different types of orthography, with the Nguni languages being conjunctive and the Sotho languages being disjunctive (Kosch, 2006:3). IsiXhosa is a Nguni language together with isiZulu, Siswati and isiNdebele. Nguni languages account for 45.8% of languages spoken most often in South Africa. IsiXhosa, one of the languages most often spoken in South African households, account for 9.3 million South African speakers (17% of the South African population), second only to isiZulu (Statistics South Africa, 2016).

Within the context of availability and quality of annotated data, isiXhosa can be classified as a resource-scarce language. The quality and levels of annotation of the language data available inform the approaches that can be pursued in the development of language technology. The scarcity of isiXhosa data has therefore led to the continued pursuit of rule-based techniques. Rule-based techniques require linguistic expertise, which is expensive and usually does not cover all the rules of a language, hence the need for better techniques. The alternative, and the contribution that is pursued in this study, is the use of unsupervised machine learning, which builds a model from raw data, thereby bypassing the need for annotated data or rules

Hammarström and Borin (2011) differentiate between morphological segmentation, which splits words into constituent morphemes; and analysis, which also classifies the identified morphemes, a differentiation that originated from Zwicky (1966). A further differentiation is made between the above and surface segmentation where a word form is segmented into a sequence of substrings whose concatenation results in the word as opposed to morphological segmentation which does not. This work contributes the development of an unsupervised isiXhosa surface segmenter that models segmentation from raw data.

1.1 Research Work

This research work covers investigating and implementing unsupervised surface segmenters for isiXhosa and comparing their performance to a benchmark morphological segmenter.

Two modalities of unsupervised surface segmentation, i.e. branching entropy and probabilistic graphical modelling with language heuristics, are investigated.

1.1.1 Research Aim

The aim of this research work is to:

- advance the use of unsupervised machine learning in the surface segmentation of Nguni languages, evaluated on isiXhosa.

1.1.2 Research Question

The research questions that apply to the use of unsupervised machine learning in the segmentation of isiXhosa are, therefore:

- Question 1: Are there techniques that can readily advance the use of unsupervised machine learning in the segmentation of Nguni languages, and specifically isiXhosa?
- Question 2: Can the unsupervised segmentation of isiXhosa words be improved using these techniques relative to a benchmark?
- Question 3: What configuration of these techniques provides the best performance and why?

1.1.3 Research Objective

The research objectives are, therefore, to:

- Objective 1: Identify a benchmark in unsupervised surface segmentation and evaluate it on isiXhosa,
- Objective 2: Research and investigate unsupervised text segmentation techniques that could readily be used for isiXhosa,
- Objective 3: Identify, research, extend, implement and evaluate unsupervised machine learning techniques that can readily advance the surface segmentation of isiXhosa.

1.2 Research Methodology

The research included an experimental study where an intervention was applied to a sample of a population. The results of the interventions to the sample are evaluated and compared to a control intervention and generalised to the population (Welman et al., 2005:78; Rasinger, 2013:41).

In this study, the population is isiXhosa text and the sample is an existing isiXhosa segmented corpus. The interventions being evaluated are machine learning segmenters that were specifically

designed for isiXhosa, while the control interventions are an existing segmentation benchmark. The process of applying the segmenter involves first training the segmenters using a raw corpus and then evaluating them against a test corpus.

The objectives stated above were achieved by following the phases below:

1.2.1 Literature Review

A thorough literature study was done on:

- (1) Human language technology techniques;
- (2) Unsupervised text segmentation measurement techniques, and
- (3) Text Segmentation techniques.

An unsupervised text segmentation benchmark, and techniques for the unsupervised text segmentation of isiXhosa, were identified.

The objective of the literature study was to gain an understanding of the techniques used in text segmentation with a focus on unsupervised text segmentation. This work is presented in chapter two (2) of the document.

This study provides guidance on how to compare the performance of the segmenters, suitable approaches to implementing unsupervised segmenters for isiXhosa and choosing an effective unsupervised control segmenter that could be used for comparison purposes.

1.2.2 Investigation and Extension of the Identified Techniques

Two techniques, branching entropy and heuristic guided probabilistic modelling, covered in chapters three (3) to six (6), are researched, extended and implemented.

For the braching entropy technique, Chapter three (3) states the theory and Chapter (4) specifies the implementation details of the technique.

Chapter five (4) digs into the surface segmentation structure for each word category of isiXhosa and chapter six (6) specifies the modeling implemented for the heuristic guided probabilistic models.

1.2.3 Evaluation

To evaluate the performance of the isiXhosa segmenters against a benchmark segmenter, experiments were set up and the results were captured and compared. Statistical comparisons that had been determined during the literature review were used to evaluate the segmenters' performance.

The experiments were set up to ensure the reliability and validity of the results. Reliability is a measure of the repeatability of an experiment, i.e. with the method used repeatedly, it determines how consistent and stable the results are (Welman et al., 2005:9; Rasinger, 2013:28).

To ensure the validity of the result, the experiments were set up to ensure that the results could be compared without bias, using the 10-fold cross-validation.

1.3 Publications

Parts of the work presented in this thesis have been presented in the following peer-reviewed publications:

Mzamo, L., Helberg, A. and Bosch, S., 2019b. Towards an unsupervised morphological segmenter for isiXhosa. In *Proceedings of 2019 SAUPEC/RobMech/PRASA Conference*. Bloemfontein, South Africa, pp. 166–170.

Mzamo, L., Helberg, A. and Bosch, S., 2019a. Evaluation of combined bi-directional branching entropy language models for morphological segmentation of isiXhosa. In E. Barnard & M. Davel, eds. *Proceedings of the South African Forum for Artificial Intelligence Research*. Cape Town, South Africa, pp. 77–89.

Mzamo, L., Helberg, A. and Bosch, S., 2020 'Evaluation of heuristic guided character level word models for morphological segmentation of isiXhosa', in Gerber, A. J. eds. *Proceedings of the First Southern African Conference for Artificial Intelligence Research*. Johannesburg, South Africa: Center for AI Research, pp. 162–178.

1.4 Thesis Structure

This document comprises eight (8) chapters, including this introduction.

Chapter two (2) covers broad aspects of unsupervised morphological segmentation. The main themes in unsupervised morphological segmentation are explained. Based on the literature reviewed, a benchmark unsupervised segmenter is chosen, techniques that show promise in the unsupervised segmentation of isiXhosa are identified and the evaluation metrics are identified.

The review identified branching entropy and language models as the techniques to use in the surface segmentation of isiXhosa. This chapter identifies Morfessor-Baseline as an appropriate benchmark unsupervised surface segmenter.

Chapter three (3) introduces the reader to information theoretic measures, which underpin the branching entropy technique. It presents predictive information metrics that are be used test the predictability of a time series and language text.

Chapter four (4) details branching entropy techniques. A new branching entropy technique called bracing joint entropy (BJE) is introduced. In addition, a relation between the branching entropy variants and predictive information metrics is investigated. The chapter also looks at how to efficiently model the branching entropy language model.

Chapter five (5) details surface segmentation for each isiXhosa category. This is then summarised as a set of heuristics that are used to guide a probabilistic language model for text segmentation. An analysis of the dependencies among the affixes and morphemes in isiXhosa leads to four statistical language models being identified .

Chapter six (6) details the different generative models that could be used model isiXhosa words. Four (4) models that make different assumptions about the dependencies structure of isiXhosa affixes and morphemes are proposed. These models are guided by the set of heuristics identified in the previous chapter.

Chapter seven (7) details the experimental configuration used, how the experiments were conducted, how the segmenters were evaluated and presents the results. Two corpora were used, one for training the segmenters and the other for evaluation. The branching entropy based segmenter outperformed the benchmark's F1 Score and achieved as much as the benchmark on accuracy. The heuristic guided generative models segmenter outperformed the F1 Scores of both the benchmark and the branching entropy segmenter but could not reach their accuracy levels.

Chapter eight (8) summarises the work conducted, presents the main findings and proposes future work.

1.5 Conclusions

In this chapter, the need for morphological segmentation was set out. It was asserted that morphological segmentation is the base technology in the natural language processing of agglutinating languages. The case for work in isiXhosa, as a Nguni language, is motivated. The Nguni class of languages is dominant in South Africa and accounts for 45.8% of mother tongue

speakers. Commonality within the Nguni languages is strong so that this work on isiXhosa should be portable to other Nguni language with minor changes.

A list of peer-reviewed publications that stem from this research is provided.

The description of the work that will be presented in the rest of this thesis has been set out according to the defined document structure. Two techniques are applied and evaluated against an unsupervised segmentation benchmark.

2. LITERATURE REVIEW

This chapter reviews current techniques for morphological segmentation of words. The objective of this review is to guide the selection of techniques that would be best suited for the morphological segmentation of isiXhosa, whilst giving regard to the availability of language modelling resources. The technique chosen will be trained on raw text, should exhibit ease of implementation, and when used with complex languages it should provide good morphological segmentation performance.

The chapter starts with an overview of the evaluation metrics used in morphological segmentation. An understanding of evaluation metrics enables comparison of the performance of techniques used. That is then followed by a review of the techniques used in morphological segmentation.

Jurafsky and Martin (2000:5) categorise the elements of a natural language processing toolkit under "state machines, formal rule systems, logic, and probability theory", and go on to highlight state space search algorithm and dynamic programming as being the most important elements.

For the purposes of this work, this chapter categorises Human language technology (HLT) techniques under knowledge-based / rule-based, statistical based and hybrid systems. Because the oldest morphological segmentation techniques are rule-based techniques a few of them are cited as a background first. This is followed by an examination of statistical based techniques, particularly machine learning techniques. The review of these statistical techniques starts with a general review of machine learning. This is followed by specifying the different levels of supervision in machine learning. Next is a brief overview of supervised morphological segmentation. The bulk of the chapter then looks at different unsupervised morphological segmentation techniques.

A comparative study of minimally supervised morphological segmentation techniques is available in Ruokolainen et al. (2013).

2.1 Segmentation Metrics

This section looks at the metrics used in evaluating morphological segmentation performance.

A metric should be a single summary statistic, to ensure easy comparison, and be intuitive (Morris et al., 2004). It should also be direct, objective, interpretable and modular. In this way, it is easy to automate, use in a benchmark and apply (McCowan et al., 2005).

Measurement of the performance of a morphological segmenter depends on the objective of the analysis. There are methods of directly measuring the analysis. Other objectives of morphological analysis include analysis as a pre-processing step to another language processing task, e.g. information retrieval (Darwish, 2002), automated machine translation (K.A. Lee et al., 2003; Minkov et al., 2007; Virpioja et al., 2007) and in lemmatisation (Erjavec & Džeroski, 2004; Brits et al., 2005; Roth et al., 2008). In these scenarios, the morphological analysis is measured indirectly as the effect of the analysis on the information retrieval, translation or lemmatisation task.

The focus of this study is surface segmentation, therefore, the emphasis will be on direct segmentation metrics. However, it is important to understand the methods of the indirect metrics particularly as these provide a general framework for understanding prediction measurement, and several studies use these for morphological segmentation as an enhancement to another process. The review of the direct metrics follows.

2.1.1 Basics of Prediction Measurement

A useful way to understand binary metrics is illustrated by Manning and Schütze (1999:268), as reproduced below. In this simplified binary task, a predictor is required to select from the presented items, the items that satisfy a particular criterion.

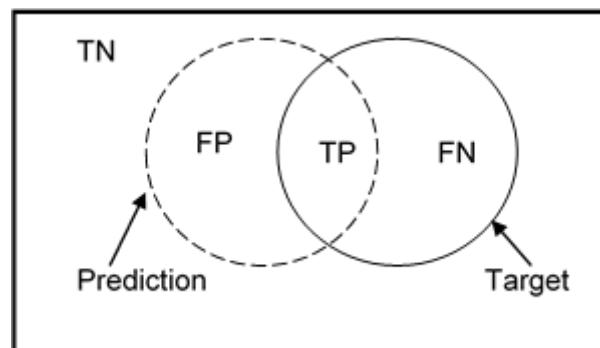


Figure 2-1: Graphical Representation of Precision and Recall Measures (Manning & Schütze, 1999:268)

In Figure 2-1, the rectangle contains all the items that need classification and four subsets of this population are shown. The different partitions of the population are labelled with acronyms defined as follows:

- TN = True Negatives: the number of items the predictor correctly identified not satisfying the criterion

- TP = True Positive: the number of items that the predictor correctly identified satisfying the criterion
- FP = False Positives: the number of items that the predictor incorrectly identified satisfying the criterion
- FN = False Negatives: the number of items the predictor incorrectly identified not satisfying the criterion

The most obvious metric is **accuracy**. Accuracy is the proportion of the items that the classifier classified correctly in relation to all the items, and this is calculated as:

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (2-1)$$

The complement to accuracy is the **error rate** (Y.-S. Lee et al., 2003), which is calculated as the ratio of items classified incorrectly to the total number of items presented for classification.

$$error\ rate = \frac{FP + FN}{TP + FP + TN + FN} \quad (2-2)$$

As much as accuracy and error-rate give an overall performance of the classifier, sometimes there is a need for a better prediction of the classifier when the importance of positive predictions differs from that of negative prediction, or cases where the ratio of the number of positive items to negative items is orders of magnitude in size. These measures are provided by the precision and recall metrics.

Precision is the proportion of items predicted to be positive (items meeting the criterion) that the predictor classified correctly,

$$precision = \frac{TP}{TP + FP} \cdot \quad (2-3)$$

Recall is the proportion of items that actually are positive that the predictor classified correctly,

$$recall = \frac{TP}{TP + FN} \cdot \quad (2-4)$$

Precision is therefore referenced to what the classifier predicted to be positive (True Positives and False Positives), whilst recall is referenced to what is actually positive (since the items that should be predicted as positive include both True Positives and False Negatives). The distinction between these two metrics is that higher precision means fewer false positives, while higher recall implies fewer false negatives (Jurafsky & Martin, 2017). One of these may be more important than the other depending on the objective of the prediction.

In many cases, a trade-off between precision and recall is needed. The most ubiquitous measure that is a trade-off between precision and recall is the F-Measure introduced by Rijsbergen (1979:174)

$$F - measure = \frac{(\beta^2 + 1) \cdot precision \cdot recall}{\beta^2 \cdot precision + recall}. \quad (2-5)$$

For many natural language processing tasks, a variant of the F-measure called the F1 Score, where $\beta = 1$, is used. The F1 Score is thus given by

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \quad (2-6)$$

Accuracy, precision, recall and the F1 Score are some of the most used metrics, as will be evident when looking at the performance of technologies used.

2.1.2 Performance on Out of Vocabulary Words

Machine learning aims to use the training data to generate a predictive model generalised over the population and tested using a testing set. In natural language processing, the in-vocabulary words are the set of words that are in both the training set and the testing test. The unseen samples, those words that are not in the training set but are in the testing set, are referred to as out of vocabulary (OoV) words, see Figure 2-2. Therefore, to test for the generalisation ability of a model, one measures the accuracy, precision, recall and F1 Score on OoV words.

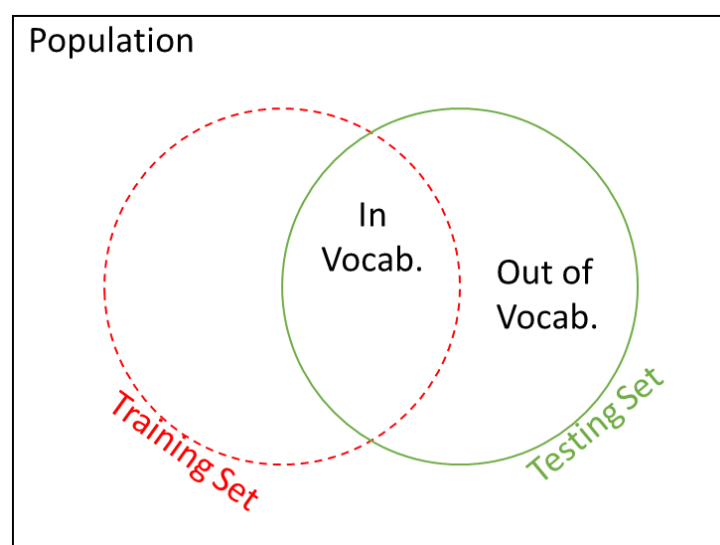


Figure 2-2: Graphical Representation of Out of Vocabulary words

In Figure 2-2, the rectangle represents the entire population of words. The training set is the set of words used to train the model. The testing set is used to measure the predictive performance of the intervention.

2.1.3 Hypothesis Testing

Evaluation of the performance of an intervention involves comparison with a control. The hypothesis is normally that the intervention performs better than the control. To evaluate if this hypothesis is achieved, one tests whether the null hypothesis is rejectable or not. In this case, the null hypothesis is that the intervention is not better than the control. So to test the hypothesis, one must reject the null hypothesis. Rejecting the null hypothesis means testing for statistical significance of the performance of the intervention. The question asked is whether the performance of the intervention is statistically better than that of the control.

The t-test (Rasinger, 2013:192; Welman et al., 2005:231) is a widely used hypothesis testing metric for comparing two interventions, or an intervention against a control. When comparing more than two interventions, the one-way Analysis of Variance (ANOVA) is used to check for statistically significant differences in the results of the interventions (Welman et al., 2005:237).

However, for evaluating HLT resources, the Wilcoxon signed-rank test (Wilcoxon, 1945) is the most appropriate. The Wilcoxon signed-rank test is recommended by Demšar (2006) who found that the widely used t-test was an inappropriate, and statistically unsafe comparison test, for classifiers. The generally accepted threshold for statistical significance is a p-value that is less than 0.05 (Rasinger, 2013:174).

2.1.4 Indirect Segmentation Metrics

As mentioned above, morphological segmentation is used to enhance information retrieval, machine translation, lemmatisation and other natural language processing functions.

2.1.4.1 Bilingual Evaluation Understudy (BLEU)

The bilingual evaluation understudy (BLEU) measure is a machine translation metric that replaced the use of human evaluators in the evaluation of a translated sentence (Papineni et al., 2002). This metric compares the predicted sentence translation to reference translations, penalises for both brevity and verbosity, and gives a rating between 0 and 1, or 0% and 100%. The BLEU has been shown to be well correlated with human evaluators (Callison-Burch et al., 2006).

2.1.4.2 Word Error Rate

The Word Error Rate (WER), a metric that is used in Automated Speech Recognition (ASR), poses the evaluation as a Levenshtein distance (Levenshtein, 1966). This metric compares two ordered lists of items, the prediction to the reference list, e.g a string or a sequence of morphemes. It measures the difference between two lists. It is based on the fact that any list can be transformed to another list by using a series of edit operations (insertion, deletion, substitution and swapping). The distance is then the count of these operations. A generalisation is to assign a weight to each operation, e.g. 1 for insertion and deletion, 2 for swapping or substitution, where the weighted operation counts are summed. The Levenshtein distance is a special case where the operations are given a weight of one (Manning et al., 2009:58; Jurafsky & Martin, 2000:154). Another modality of the Levenshtein distance is to restrict the operations to insertion and deletion. The Word Error Rate is defined as (McCowan et al., 2005)

$$WER = \frac{S + D + I}{N}, \quad (2-7)$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions and N is the total number of items in the reference list. A problem with the word error rate is that it can be difficult to interpret as its value can go beyond 100%.

The word error rate concept is used in evaluating phoneme identification as phoneme error rate (PER) (Heymann et al., 2013; Neubig et al., 2010) and evaluating character identification as character error rate (CER) (Chueh et al., 2006; Smith et al., 2005).

2.1.4.3 Perplexity

One of the indirect metrics to measure a segmentation is perplexity. Perplexity is based on entropy, the randomness in the segmentation.

Definition 2-1. Let X be random variable with events outcomes $x \in \chi$. Let the probability of an event $p(X = x)$ be $p(x)$. The **entropy** of X , $H(X)$, under some base b , is calculated as

$$H(X) = \sum_{x \in \chi} p(x) \log_b \frac{1}{p(x)}. \quad (2-8)$$

Definition 2-2. The **perplexity** of a random variable X , $PP(X)$, with an entropy $H(X)$ is defined as:

$$PP(X) = 2^{H(X)}. \quad (2-9)$$

In language modelling, a reduction in perplexity, and therefore entropy, normally means a better language model. For language model-based segmentation algorithms reduction in perplexity is a good measure of a better segmentation.

2.1.5 Direct Morphological Segmentation Metrics

In this work, a look at metrics that provide a direct measure of morphological segmentation is done. Two concepts are measured directly in morphological segmentation, namely boundary identification and morpheme identification.

For a start, the metrics used either measure the proportion of correct results against the overall results or the proportion of incorrect results, i.e. error. In the former, the metrics used are accuracy, precision, recall and F1 Score, whilst in the latter, the error rates are used.

2.1.5.1 Levels of Measuring Segmentation

Two things can be read from a morphological segmentation:

- The ordered list of segment boundary locations of a word, and
- The ordered list of segments of a word.

From the above, one can measure the following:

- Whether the segmentation has the correct segment boundaries within the word or not. This is morpheme boundary identification,
- Whether a word is segmented correctly or not. This is a measurement of segmentation at the word level, and
- Whether the segmentation has the correct segments or not. This is morpheme identification.

The above measurements are, of course, equivalent, and the interest is whether a prediction is correct or not. This makes segmentation a binary classification task. Therefore the task is a classification of:

- Whether a list of morphemes is the correct list for the word,
- Whether a sequence of letters in a word constitutes a morpheme or not, or
- Whether a location in a word is a morpheme boundary or not.

Section 2.1.1 *Basics of Prediction Measurement* details the morphological segmentation metrics used. These measures apply in all of the above segmentation classification tasks.

2.1.5.2 Morpheme Boundary Identification measurement

The easiest way of measuring the performance of a segmentation system is to measure morpheme boundary identification. In a text corpus, the locations between characters/letters are all possible morpheme boundaries. One can measure true positive, false positive, true negative and false negative morpheme boundaries in this way. This form of measurement assumes surface segmentation, that is, concatenation of the segments results in the segmented word. Linguistic morphological analysis results in segments whose concatenation does not result in the word, hence the use of morpheme identification measurements (Spiegler & Monson, 2010).

2.1.5.3 Morpheme Identification measurements

In a morpheme identification setup, for each word, w_i , there are two lists of morphemes to compare, the reference list, R_i , and the predicted list, P_i , of morphemes (McCowan et al., 2005). For each word, the recall is calculated as

$$\rho_i = \frac{|R_i \cap P_i|}{|R_i|}, \quad (2-10)$$

and the precision is calculated as

$$\pi_i = \frac{|R_i \cap P_i|}{|P_i|}, \quad (2-11)$$

The precision and recall for the corpus of N types is calculated as

$$\rho = \frac{1}{N} \sum_{i=1}^N \frac{|R_i \cap P_i|}{|R_i|}, \quad (2-12)$$

$$\pi = \frac{1}{N} \sum_{i=1}^N \frac{|R_i \cap P_i|}{|P_i|}. \quad (2-13)$$

In a corpus of words, the word **token** refers to the individual duplicated words in the corpus and word **type** refers to the unique words. As an example, a corpus of 100 words has 100 tokens, but less or equal to 100 types. If a corpus has the same number of tokens as types then there is no duplication of words in the corpus.

2.1.5.4 Evaluation Metric for Morphological Analysis (EMMA)

In unsupervised morphological analysis with labelling, the morphological analyser has no way of knowing the labels in the reference analysis, and replacement labels are instead used. Because

of this, a number of techniques have been devised to map the segmenter's analysis to the reference analysis (Sami Virpioja et al., 2011). These metrics use what is referred to as isomorphism - optimising a bidirectional mapping between the reference analysis and the segmenter's analysis - and making an evaluation from that mapping. The Evaluation Metric for Morphological Analysis (EMMA) (Spiegler & Monson, 2010) is one such metric. This metric uses linear programming to optimise the mapping, and the basic morpheme identification calculations, Equations (2-10)-(2-13), to calculate the measurements for the entire test corpus.

2.1.6 Conclusions on Segmentation Metrics

This section gave a brief overview of evaluation techniques used in unsupervised morphological segmentation. It started by giving the basics of prediction measurements followed by measurement of generalisation in prediction.

The main categories of evaluation metrics for segmentation were presented as morpheme boundary identification and morpheme identification. The morpheme boundary identification metrics are the simplest to implement, but they assume surface segmentation. The morpheme identification metrics do not assume surface segmentation but can be used for segmentations whose concatenation does not result in the text being segmented.

For our work, which is unsupervised surface segmentation we choose morpheme boundary identification. Our motivation is the use of the Occam's razor principle (Blumer et al., 1987). The simplest and yet adequate evaluation to use for surface segmentation is morpheme boundary identification.

For hypothesis testing the Wilcoxon signed-rank test is the best test for the metrics identified above.

2.2 Knowledge/Rule-based techniques

Knowledge / Rule-based systems capitalise on the knowledge of linguistic experts. Because of the maturity of linguistic knowledge, this is an obvious choice for automating language processing. In this setup, expert knowledge of the language is encoded as rules that a computer executes.

The *de facto* platform in use is the finite-state automaton.

2.2.1 Finite-State Automata

A finite-state machine/automaton (FSA) is a flexible state machine that is used to model a rule sequence. A state machine is composed of two parts, states and transitions with rules - it takes

the data as input and returns the possible sequences of states that model the data. The transitions control the movement of the state machine between states. The transition rules determine the sequence of states that the machine takes to model the data. A transition links two states, is unidirectional and is activated when all the rules attached to it are satisfied. Transitions rules are modelled as regular expressions (Jurafsky & Martin, 2000:21). The Xerox Finite-state Automata is one of the default applications in use.

One of the seminal works on morphological computation that enabled better implementation of language rules was the language-independent model for word-form analysis and synthesis by Koskenniemi (1984). The model provided a computationally efficient way of moving from lexical representations of words to their surface forms in just a single step. This allowed for the compression of the generative processes in the morphology of a word to a **two level description**. The model is bidirectional (analysis and synthesis) and was tested on a range of languages, and implemented as Finite-state Automata (FSA).

One of the works that used the two level description was the implementation of the two-level finite-state analyser for Arabic, a highly morphologically complex language, by Beesley et al. (1989) and later by Beesley (1996). This work culminated in the Buckwalter Arabic Morphological Analyser (BAMA) by Buckwalter (2002; 2004), a robust finite-state morphological analyser that has seen extensive use for many variants of Arabic.

Closer to home, Pretorius and Bosch (2005) presented a prototype for the morphological analysis of isiZulu using language rules. The system was built using the Xerox **finite-state tools**, i.e. the Finite-State Lexicon Compiler (lexc) and the Xerox Finite-state tool (xfst) (Karttunen, 1993). The prototype analysed 69% of the test tokens presented to it. This work was subsequently adapted for other Nguni languages i.e. isiXhosa, Siswati and IsiNdebele (Bosch et al., 2008).

FSA are usable for languages as complex as Turkish, as shown in the work by Çöltekin (2010). Çöltekin presented TRMorph, a freely available morphological analyser for Turkish. The analyser is **rule-based** and was implemented using the Stuttgart **finite-state** transducer (Schmid, 2005). TRMorph analysed 85% of the words in the Turkish Wikipedia.

Recently a **finite-state morphological analyser** was used for analysing Lakota, an endangered language spoken by one of the largest Native American communities in the United States of America (Curtis, 2014). The analyser properly analysed 59.1% of the 631 words tested. This was a small study to test the feasibility of a morphological analyser for Lakota.

2.2.2 Logic Programming

Another method of encoding expert knowledge is logic (Russell & Norvig, 2014:215). Propositional logic provides for a mechanism to make statements in a shortened form. A symbol, e.g. “A”, represents a statement, e.g. “the sky is red”. $\neg A$ would then represent, “the sky is not red”. The symbol $\neg$ is a negation operation. Propositional logic includes a number of other logic operators, e.g. $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$, for “or”, “and”, “if .. then ..”, and “if and only if”. After encoding knowledge using symbols and operators one can then test for validity of a proposition, using entailment. Entailment returns an indication of whether a statement/proposition is valid, based on the encoded knowledge.

First Order Logic (Russell & Norvig, 2014:290) or predicate logic is an extension of propositional logic. In First Order Logic, a statement is broken down into the object, the predicate and quantifier operation. Quantifier operations use are, $\forall$ and $\exists$ for “all” and “there exists” respectively. An example is that *red(sky)* would represent “the sky is red”, and *mother(jane, john)* would represent “jane is john’s mother”, and $\forall \text{john} \exists \text{jane} \text{mother}(\text{jane}, \text{john})$ would mean that “for all john there exists a jane who is the mother of john”. Inference in First Order Logic involves searching through the encoded knowledge to find the cases under which an input is true (Russell & Norvig, 2014:327).

Spiegler et al. (2010b) used definite clause grammars, an instance of propositional logic, to create an open source morphological corpus for isiZulu called *Ukwabelana corpus*. The labelling scheme for the corpus is described in Spiegler et al. (2010a). The corpus comprises of 10000 labelled word types, 100000 unlabelled word types, 3000 part-of-speech tagged sentences and 30000 raw sentences, a morphological isiZulu grammar and a parsing algorithm. The corpus is available at <http://www.cs.bris.ac.uk/Research/MachineLearning/Morphology/Resources>.

Muggleton & de Raedt (1994) introduced inductive logic programming, an extension of first order logic that, in addition to using encoded expert knowledge, learns from examples. This technique was used by Dzerski & Erjavec (1998) to learn nominal inflections of English, Romanian, Czech, Slovene and Estonian.

2.2.3 Challenges of rule-based systems

The most obvious challenge of developing rule-based systems is the need for expert knowledge, and the length of time needed to develop them. In addition, rule-based systems produce a number of valid segmentations, even though only one output is valid, requiring disambiguation. There are isiZulu examples of these in Bosch (2020). Because of this, rule-based systems have seen augmentation with statistical data, e.g. weighted finite-state automata (Allauzen et al., 2007),

probabilistic context free grammars (Stolcke, 1995) and stochastic logic programming (Muggleton, 2019). This is also where machine learning has a role to play.

Machine learning is a field under statistical based and hybrid techniques. Machine learning is characterised by learning a model from data. Machine learning based systems learn the model of a task from experience, the data. This model is then used in executing the task. A characteristic of machine learning is improvement in performance with more experience, i.e. more data (Mitchell, 1997).

Machine learning techniques are differentiated based on the characteristics of the data they take as learning input, the training data. This is referred to as learning supervision.

2.3 Supervision in Machine Learning

There are four modes of training a machine learning model, i.e. supervised, semi-supervised, unsupervised and reinforcement learning (Manning & Schütze, 1999:232; Murphy, 2012:2).

In supervised learning, the training data contains solution examples that the model must generalise from, e.g. word and word segmentation pairs for a segmentation task. Data in unsupervised training is devoid of such, and generates a model from raw data. Therefore, in a segmentation exercise, the training data would be words only. Semi-supervised systems use something in between, from using limited supervised data with large amounts of unannotated data to unannotated data with rules built into the model.

This difference in supervision can be seen in the work of Spiegler et al. (2008) on learning the morphology of isiZulu using the different levels of supervision. Under the conditions of limited data, and data sparsity, supervised learning outperformed the other levels, followed by semi-supervised, while the unsupervised method gave the lowest performance. The average F1 scores of the respective levels of supervision were 81.24%, 53.6% and 40.93%. The results were consistent for precision and recall as well. This is characteristic of performance between the different levels of supervision.

In reinforcement learning, the system is given feedback on its performance after several attempts at a solution. The system then learns from this feedback and adjusts its strategy as it attempts to build a model. It is useful when the learning system needs to be given occasional reward or punishment based on its performance.

In the following sections, works in supervised, semi-supervised and unsupervised learning are covered. Reinforcement learning will not be covered, as there is little text segmentation work in the field. Let us start with the studies in supervised morphological segmentation.

2.4 Supervised Morphological Segmentation techniques

There exists an extensive body of work in supervised morphological segmentation. This section provides a cursory look at that work with emphasis on the segmentation of morphologically complex languages.

In this section segmentation works using rules lists, Hidden Markov Models, Maximum Entropy Modelling, Markov Random Fields, Support Vector Machines, Markov Chain Monte-Carlo based systems, Memory-based systems and Neural Networks, will be covered. It is important to note that although there are other supervised segmentation techniques that do not fall into the above categories, the categories used are the most prevalent. An example are the works by Quasthoff et al. (2014; 2016) that use the Maximum Affix Overlap algorithm and decision trees respectively to segment isiZulu text.

The simplest form of supervised morphological segmentation is the inference of morphological rules from word-segmentation pairs.

2.4.1 Rules lists

Three works are worth noting for the use of rule lists in supervised segmentation, i.e. the works of Theron and Cloete (1997), Darwish (2002) and Mzamo et al. (2015) because these works are specific to morphologically complex languages. As input, these techniques take a list of pairs, consisting of surface forms paired with some analysis, and generate a segmentation rules list from that. The analysis could be a root, a lemma or a full segmentation.

Theron and Cloete (1997) described a method for automatic context free acquisition of **two-level rules** for morphological analysis. Given a source-target word pair where the target is an inflected form of the source, the method generated segmentation rules in the form of an acyclic finite-state automaton from string edit sequences of the word pair. The study used 5-fold validation to measure both recognition accuracy and generation accuracy. The method was tested on English adjectives, isiXhosa noun locatives and Afrikaans noun plurals. The method achieved 100% for isiXhosa noun locatives. The word lists were small but they proved the concept.

Darwish (2002) presented a supervised statistical based algorithm for extracting roots from Arabic words, called Sebowai. Given word-stem pairs, the system automatically generated the inflection rules, and captured rule statistics for disambiguation. The rules with stochastic information were used for root extraction. The different experiments done averaged to a performance root extraction accuracy of 91.97%. The root extraction was then used in information retrieval. Information retrieval searching using word roots was measured at a precision of 45%.

Similar work on the lemmatisation of isiXhosa was done by Mzamo et al. (2015). In this work, a lexicalised probabilistic graphical lemmatiser (XGL) was introduced. XGL used a hierarchical list of inflection rules that were generated from isiXhosa word-lemma pairs. The inflection rules also contained prevalence statistics that were used for disambiguation. The XGL lemmatiser achieved the highest accuracy compared to selected benchmark lemmatisers, with an accuracy rate of 83.19%.

2.4.2 Hidden Markov Models

Hidden Markov Models (Rabiner, 1989; Baum & Petrie, 1966) have seen extensive use in supervised speech recognition. Markov chains are statistical state-space processes that satisfy the Markov property. State-space processes are modelled by states and transitions between states. Stochastic state-space processes have associated probabilities for the start state, activation of a transition between states, and the output of a state. In the modelling of dependence between states, the Markov property restricts the dependence of the next state to only a subset (n) of the previous states. This is referred to as the n -order Markov model. An n -order Markov model satisfies the following probability equation:

$$P(x_{k+1}|x_{1..k}) = P(x_{k+1}|x_{k-n+1..k}) \quad (2-14)$$

where $x_{1..k} = x_1, x_2, \dots, x_k$ is a sequence indexed from 1 to k and $x_{k-n+1..k}$ is a substring of $x_{1..k}$ with characters from index $k - n + 1$ to k .

In the basic Markov process the states, the transitions and the related probabilities are visible. Hidden Markov Models (HMM) are Markov Models that have hidden states that are not apparent in the data and that must be inferred.

What makes HMMs attractive is their ability to model hidden states that affect the observed process output and the use of the Viterbi algorithm (Viterbi, 1967) which makes searching for the most probable states that are responsible for the output highly efficient.

Baker (1975) and Jelinek (1976) introduced the use of HMMs in speech recognition, as explained in a tutorial by Rabiner (1989). Yamron et al. (1998) presented the use of HMM in text segmentation, specifically in finding topic transitions.

A number of HMM variants have emerged, including the Hierarchical HMM (Fine et al., 1998) which was used by Zhang et al. (2003) in Chinese word segmentation, POS tagging, and word recognition.

Another variant of the HMM is the Hidden Semi-Markov Model (Kevin P Murphy, 2002; Yu, 2010), an HMM that can emit a sequence of observations instead of just one. This was used for speech recognition in English and Arabic (Pylkkönen & Kurimo, 2004; Khelifa et al., 2017).

2.4.3 Maximum Entropy Modelling

A technique that dates back to the 1990s but continues to see wide spread use in natural language processing is a supervised machine learning technique called **maximum entropy modelling (MaxEnt)** (Berger et al., 1996).

Maximum Entropy modelling uses a principle akin to the Occam's razor principle (Blumer et al., 1987), which demands the use of the simplest appropriate solution. MaxEnt modelling takes from the maximum entropy principle, which requires that the model "must use a probability distribution which has maximum entropy subject to whatever is known" (Ratnaparkhi, 1997; Jaynes, 1957). The framework allows for a model designer to specify features that the model constrains with observations from training data in arriving at an appropriate statistical distribution model. Inference uses modified iterative scaling (Della Pietra et al., 1995).

This work was initially demonstrated in modelling segmentation of French in the English-French translation. MaxEnt was also used in POS tagging for English (Ratnaparkhi, 1996) with state-of-the-art accuracy rates of 96.6%. The MaxEnt has been used in a number of studies; some of them are listed below.

Sarikaya et al. (2007) presented a language model, Joint Morphological-Lexical Language Modeling (JMLLM), that jointly uses morphological and lexical information in modelling the probability of an Arabic sentence. The model assumes the availability of morphological analysis of lexical tokens. The morphological and lexical information is combined into a tree structure called the Morphological-Lexical Parse Tree (MLPT). The model uses Maximum Entropy modelling with morphological and lexical features. The model achieved a Word Error Rate (WER) of 29.8% in an Iraqi-Arabic Speech recognition experiment.

MaxEnt was also used as the Maximum Entropy Markov Model (McCallum et al., 2000; Kakade et al., 2002) in the inflection prediction of Russian and Arabic for translation of English to Russian and English to Arabic (Minkov et al., 2007). Maximum Entropy Markov Models (MEMMs) try to solve three limitations of the HMM, i.e.:

1. HMMs only model the observations as raw text and do not allow for incorporation of specific features into the model. Likely features are whether a text has a capital letter or not, part of speech (POS) information, proximity to the beginning or end of a sentence etc.

Part of Speech categories include Noun (N), Verb (V), Adjective (Adj), Noun Phrase (NP) etc.

2. An HMM only works with known states. In some text work, the state could be unknown and only discoverable at training or inference stage.
3. The HMM maximised the likelihood of the observation sequence while the requirement is the maximisation of the state sequence given the observation.

The first difference between HMMs and MEMMs is the conditional probabilities. Whilst the observations are conditioned on the state, $P(o|s)$, in HMM, the MEMM model conditions the state on the observation, $P(s|o)$. This changes the model from a generative model to a discriminative model (Murphy, 2012:685). The conditional distributions are modelled as exponential features inspired by maximum entropy modelling. Variants of the Viterbi and the Baum-Welch algorithms are used for learning and exact inference.

2.4.4 Markov Random Fields

Hidden Markov Models are regarded as a specific type of Dynamic Bayesian Network (Kevin Patrick Murphy, 2002). Bayesian Networks are probabilistic graphical models that use directed graphs, specifically directed acyclic graphs (DAGs). Dynamic Bayesian Networks model sequential stochastic processes using directed graphical models (DGM). The directionality of an arc in directed graphical model specifies the dependency between two random variables, or groups of random variables, represented with vertices, with the arrow pointed to the dependent random variable. See Figure 2-3 for an example of a Bayesian Network, with (2-15) as the corresponding joint probability distribution. HMM are generative linear dynamic Bayesian networks.

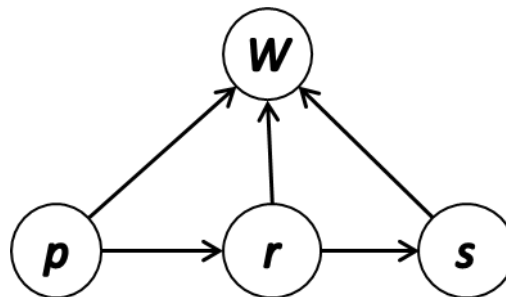


Figure 2-3: An example of a Directed Graphical model

$$P(W, s, r, p) = P(W|s, r, p) \cdot P(s|r) \cdot P(r|p) \cdot P(p) \quad (2-15)$$

HMM are generative, maximising the joint probability of the hidden states and the observations, hence the need for strict conditional independence to make inference tractable. As mentioned above MEMM try to solve this by conditioning the current state on the current observation and the previous states instead of conditioning the current state only on the previous state and conditioning the observation on the current state.

There are cases where directed graphical models are limited in their functionality, and undirected graphical models (UDM) are better models (Murphy, 2012:661). In fact, there are distributions that can be modelled by UDMs and not DGMs (Murphy, 2012:665). Another name for UDMs is Markov Random Fields (MRF) or Markov networks. An example of an undirected graphical model is show in Figure 2-4, with the corresponding distribution

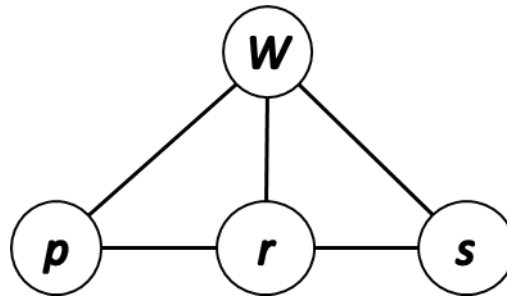


Figure 2-4: An example of an Un-directed Graphical Model

$$P(W, s, r, p | \theta) = \frac{1}{Z(\theta)} \cdot \varphi_{wrs}(W, r, w | \theta) \cdot \varphi_{wpr}(W, p, r | \theta) \quad (2-16)$$

where φ_c is a potential/factor function for the maximal clique c parametrised on the clique random variables and θ are the model parameter. A clique is a subgraph where all the nodes are interconnected and the maximal clique is the largest clique from a number of variables.

Jung et al. (1996) used Markov Networks in the POS-tagging of English with better results at 97.1% accuracy compared to the 96.11% of an HMM model. MRF have also been used in extraction of text from images and segmentation of such extracted text (Tian et al., 2015; Renuka & Terdal, 2017). Dreyer & Eisner (2009) did some work on the generation of un-observed morphological forms in a paradigm for English using MRFs.

Conditional Random Fields (CRFs) (Lafferty et al., 2001) are an undirected graphical model for sequence modelling (Lafferty et al., 2001). They model the joint probability of the entire sequence of states given the entire sequence of observations, as compared to MEMMs that do per state modelling. CRFs can learn using maximum likelihood, $P(D | \theta, m)$, or maximum a posteriori (MAP)

estimations, $P(\theta|D, m)$, where D is the training data, θ are the model parameters and m is the model. Below a number of studies that used CRFs in supervised natural language processing are cited. Linear chain CRFs are the CRFs used in sequence modelling. Like MEMM, CRFs also use exponential features in their conditional representation. CRF maximise the log-likelihood objective function $\mathcal{O}(\theta)$:

$$\mathcal{O}(\theta) = \sum_{i=1}^N \log p_{\theta}(\mathbf{y}^i | \mathbf{x}^i), \quad (2-17)$$

where N is the length of the training data, i is the index for the i -th data point, $(\mathbf{x}^i, \mathbf{y}^i)$ and $1 \leq i \leq N$; $i, N \in \mathbb{Z}^+$. Parameter estimation is done using the iterative scaling algorithm (Della Pietra et al., 1995) or gradient based optimisation (Murphy, 2012:692).

Peng et al. (2004) presented work on the supervised segmentation of Chinese using linear-chain CRFs. This work included identification of new words using segment confidence estimation. Confidence estimation of a segment is done using a technique called Constrained Forward-Backward algorithm (Culotta & McCallum, 2004). In this work a segment of text is a new word if its confidence estimation is high or if it has low confidence but it is surrounded by segments with high confidence. The technique was tested on four Chinese datasets of varying sizes. The dataset had varying out of vocabulary (OoV) rates ranging from 2.2% to 18.1%. The technique's performance was measured at an average F1 Score of 91.85%. For OoV words only, recall was measured, and it ranged between 40.3% and 61.9% and was highly correlated with the dataset OoV rates.

Smith et al. (2005) used CRF in the supervised morphological disambiguation of Korean, Arabic and Czech. The study looked at segmentation/tokenisation, POS tagging and lemmatisation. The approach also used possible false analyses in training as well to optimise the solution. For features, the approach used morpheme trigrams and head morpheme trigrams in one of the models. The model used existing morphological dictionaries from existing morphological analysers. An improvement in training was seen with factored training. Factored training is where the training of the head trigrams is done separately. The study also recorded a rate of 1.7 morphemes per word ($\sigma = 0.77$) in the training set for all three languages with the best performance for the three languages being:

Language	POS Tagging (% accuracy)	Segmentation (% accuracy)	Lemmatisation (% accuracy)
Korean	92.2	90.7	97.6
Arabic	96.1	95.5	94.6
Czech	98.6	89.2	98.7

Andrew (2006) introduced the **Hybrid Markov/Semi-Markov conditional random field (semi-CRF)** algorithm, a relaxation of the Markov property in linear-chain conditional random fields. This paper introduced the log conditional odds feature. Extensive log-linear model feature engineering for Chinese is explained in the paper. Using label unigrams and bigrams, length and word identity, and log-odd features, the study achieved an F1 Score of 96.77% on Chinese.

Hai Zhao et al. (2006) presented feature enhancement to segmentation of Chinese with CRF. The feature enhancements involved two additional tags to the 4-tag character-tagging scheme, extending it to 6 tags. The results of the enhancement were an F1 Score of 95.3%.

Gahbiche-Braham et al. (2012) also used CRFs in the segmentation and POS tagging of Arabic to improve machine translation. The system achieved 0.4% word error rate in segmentation and a BLEU score of 32.9%. The systems implemented were faster than the reference MADA. Whilst MADA processes 90 words per second, the CRF tagger did 2960 words per second for tagging and segmentation.

Hatori et al. (2012) presented a supervised joint model that does word segmentation, POS tagging and dependency parsing together using CRFs. The model was tested on three Chinese data sets. The average performance over the three data sets used was 96.38%, 92.06% and 75.44% for text segmentation, POS tagging and dependency parsing respectively.

Ruokolainen et al. (2013) implemented supervised morphological segmentation using CRF trained with a small annotated corpus. Their approach outperformed semi-supervised morph lexicon approaches, supervised Morfessor and a log-linear model (Poon et al., 2009) on Arabic, Hebrew, English, Finnish and Turkish at the time.

Cotterell et al. (2016) introduced canonical segmentation, which is a segmentation of a word into canonical morphemes using a log-linear model similar to semi-CRFs. The approach used importance sampling for gradient learning (Bengio & Sen  cal, 2003). The approach was

evaluated on English, German and Indonesian with superior F1 Scores than a semi-CRF and weighted finite-state transducer.

2.4.5 Support Vector Machines

Support vector machines (SVM) (Cortes & Vapnik, 1995) are linear classification models that transform the input space into a high dimensional feature-space to maximise the separation of the classes. This separation is done via a hyper-surface in the feature-space. Many transformation models have been designed for SVM, however, there is no need to delve into those models here. Vapnik (1999) provides a good description of the theory. Below a few works in SVMs are listed.

Gao et al. (2004) presented a supervised model for simultaneous sentence segmentation and new word identification for Chinese. Segmentation was done using linear discriminant functions (Hastie et al., 2009:108). New word identification was done during segmentation avoiding error propagation and used support vector machines. The technique achieved precision, recall and F1 Score of 95.2%, 95.9% and 95.5% respectively.

Diab et al. (2004) used an SVM in the automatic tokenisation/segmentation, POS tagging and base phrase chunking of Arabic. Chunking groups of words in a sentence into non-overlapping meaningful segments, or phrases (Jurafsky & Martin, 2000). The tool used was Yamcha¹, a freely available SVM tool. The experiments were done on the Arabic Penn TreeBank 1 (v2.0). Tokenisation involved classifying each letter in a word into five categories, i.e. beginning of a prefix 1 or 2, beginning of the stem, other stem letters, beginning of a suffix and other suffix letters. A collapsed list of POS tags was used for chunking. The experiments showed that an SVM can be used in these tasks with an F1 Score of 99.12% in tokenisation, 95.49% accuracy in POS tagging and an F1 Score of 92.08% for base phrase chunking.

Habash and Rambow (2005) presented a supervised approach to combined segmentation and POS tagging of Arabic. The approach used an existing finite-state morphological analyser, ALMORGEANA (Habash, 2007) to generate candidate morphological analysis for a word, and support vector machines (SVM) for disambiguating segmentation and POS tagging. The features used are inspired by Arabic linguistic morphology, span several words around the candidate word for context and each feature is independently trained with its own SVM. The approach achieved a segmentation F1 Score of 99.1% and POS Tagging accuracy of 97.6%.

¹ Yamcha is available at <http://chasen.org/~taku/software/yamcha/>

Abdelali et al. (2016) presented a support vector machine based Arabic segmenter, called Farasa. Farasa depended on lexicons of valid Arabic stems. The system achieved segmentation accuracy of 98.76%.

2.4.6 Markov Chain Monte-Carlo based segmenters

Sampling methods are useful in modelling cases where the partitioning function (normalisation factor) in a model is intractable. In these cases, conditional sampling can be used to model the conditional distribution and to estimate marginal distributions, called Markov Chain Monte-Carlo. The inference of choice is the Gibbs sampler (Geman & Geman, 1984).

Goldwater et al. (2006; 2009) proposed two supervised MCMC Bayesian word segmentation methods, one using unigrams and one using bigrams. The intention of the paper was to highlight the importance of contextual dependencies in word segmentation. The models used were based on the Dirichlet process, a non-parametric Bayesian process. Non-parametric models do not assume a finite set of states, and at most they assume a countably infinite set. The unigram model was based on the Dirichlet processes (DP) (Ferguson, 1973), and the hierarchical Dirichlet process (HDP) (Teh et al., 2005). Character-wise Gibbs sampling (Geman & Geman, 1984; Brooks, 1998) was used to perform Bayesian inference. The unigrams algorithm showed an F1 Score performance of 53.0% on English. The HDP improved the results to an F-measure of 76.6% in the Brent English child-directed speech utterances corpus of 33399 words. The precision was reported as 61.9% for the Dirichlet Process and 79.4% for the HDP. The study showed that context is important in segmentation.

Lee et al. (2011) used MCMC to validate the importance of the connection between Part of Speech (POS) categories, word affixes and sentence word affix agreement in the segmentation of words in a sentence. The study assumes that words with common affixes are likely to belong to the same POS. Also, grammar agreement is likely to be realised in inter-word affix correlations within a sentence. The approach uses graphic modelling and Gibbs sampling for inference. The approach was tested on Arabic with segmentation accuracy of 86.2%.

Silfverberg and Hulden (2017) presented work on the morphological segmentation of text via Universal Dependencies. Universal Dependencies are word level annotations that mark the linguistic features of a word. The intuition is that linguistic features in a word are expressed as morphemes, and with enough data, it should be possible to identify the morphemes that match a particular feature and to identify such in a word. The study used a symmetric conditional probability objective function which was initially used by da Silva et al. (1999) for multi-word expression extraction. The study used the Gibbs sampler for inference over the large possible

segmentation-label space. The method was tested on Finnish, Spanish and Swedish, and performed better than Morfessor baseline (Creutz & Lagus, 2005). The F1 Scores (%) are tabulated below:

	Morpheme boundaries	Unlabelled morphemes	Labelled morphemes
Finnish	90.89	66.67	85.02
Spanish	86.45	52.08	76.39
Swedish	91.28	58.69	92.73

2.4.7 Memory-Based learning

Also called instance based learning, memory-based learning (Stanfill & Waltz, 1986) takes an input and compares it to the training data set and classifies it to the class of a subset of the training set that is most similar to it. The most obvious type of memory-based learning is the K-nearest neighbour algorithm.

Van den Bosch and Daelemans (1999) presented the use of **memory-based learning** in the morphological analysis of Dutch. The algorithm achieved 93% accuracy and 84% for out of vocabulary words.

Stroppa and Yvon (2005) provided an algebraic framework to a variant of memory-base learning called analogical learning. The technique was evaluated on the lemmatisation and POS-tagging of English, Dutch and German with results comparable to van den Bosch and Daelemans (1999). The second task was the generation of a morphological parse tree for unknown English lemmas. Not enough work was done to report definitive results on the second task.

De Pauw and De Schryver (2008) presented the **Memory-Based** Swahili Morphological Analyser (MBSMA). It was trained using lemmas generated by a rule-based morphological analyser (SALAMA) (Hurskainen, 2004). The output of SALAMA, the Helsinki Corpus of Swahili (HCS), consists of word forms, lemmas and morphosyntactic descriptions (MSD). A sample of 97000 words was extracted from the HCS and 10% was hand annotated to be a gold standard evaluation set. Two versions of the MBSMA were developed, one being the original character based analyser (MBSMA-c) and the other a syllable based analyser (MBSMA-s). The lemmas were generated by the HCS trained MBSMAs and the original HCS lemmas were evaluated against the gold standard and Morfessor. The syllable based MBSMA had the lowest lemmatisation error rate at 11.7%

followed by the HCS lemma output at 12%. The MBSMA-c performed at a word error rate (WER) of 13.6% with Morfessor-baseline being the worst performer at 73.6%.

Quasthoff et al. (2016) used Memory-Based learning in the segmentation of isiZulu and German using small training sets of 8301 and 70000 words respectively. The approach achieved a maximum accuracy rate of 96% for both isiZulu and German.

2.4.8 Neural Networks

An artificial neural network (ANN) or a multilayer Perceptron is a computational network of artificial neurons that is used to model an artificial intelligence function. The neurons in an ANN are arranged in a stack of layers such that a neuron in a layer is connected to neurons in neighbouring layers, and computation is one directional. An ANN has at least three layers, the input layer an output layer and at least one hidden layer between the input and the output layer. A single layer can have many neurons. ANNs are sometimes called feed-forward networks because the computation is one directional from the input layer to the output. A neuron receives input from a number of neurons of the previous layer, does a weighting of each input and a simple non-linear computation of those weighted inputs and passes the result as input to neurons of the next layer. If one considers the inputs to a neuron as a vector, I , and the input weights as a vector, w , and non-linear function, f , the output of the neuron, o , is then calculated as

$$o(I) = f(w^T I), \quad (2-18)$$

and the vector of the outputs of the neurons in a layer $O(I)$, if the vectors of weights are combined into a matrix, W , can be calculated as

$$O(I) = f(W^T I). \quad (2-19)$$

Training of the input weights for the neurons is done using a technique called error back-propagation (Rumelhart et al., 1986) from input-output pairs in supervised training.

What makes ANNs versatile are the many design possibilities regarding both the number of neurons in each layer, the number of layers in an ANN, and the design on the neurons themselves. Three neuron designs that have revolutionised ANNs are auto encoders, the Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). A good text on the latest ANN is by Aggarwal (2018).

Ordinarily, an ANN is trained in a supervised manner with input-output pairs, but what if one wants a compressed representation of the input? That is where auto encoders come in. Auto encoders are neural networks that find a compressed representation of the input. This is achieved by having

one of the hidden layers constricted (with very few neurons) such that the network presents a compressed representation of the input at that layer and training the network with input-input pairs. This constricted layer is referred to as the code layer. The layers between the input layer and the code layer are referred to as the encoder and the layers to the output are referred to as the decoder. Another name for an auto encoder is the encoder-decoder.

The formulation of the ANNs above is memory-less, meaning that the neurons do not use their current output in the calculation of the next output. RNNs provide sophisticated mechanisms for a neuron to take sequential data and use their output as input in future calculations. This allows for time-series models, text processing and biological data analysis.

In addition to the above, our formulation took the input as a vector, not a matrix, as in a picture, or a tensor, where there are relationships between the different values in the input. This is where Convolutional Neural Networks (CNN) are used.

A good overview of Neural Networks for scarce resourced languages is Loubser and Puttkammer (2020). In this section, only an overview of works in morphological analysis is provided.

Collobert and Weston (2008) presented a **single convolutional neural network architecture** that could analyse a sentence and produce POS tags, chunks, named entity tags, semantic roles and semantic similarity words for each word in the sentence, and give a likelihood of the grammatical and semantic correctness of a sentence. The language model is learnt from unlabelled data and the rest from labelled text. Of interest is the semantic representation of a word, the so called “embedding”. An embedding is an estimation of how a particular word semantically relates (synonym, homonyms, hypernyms etc.) to other words. This is modelled as a vector from the context of the word. Context is modelled using proximity. The network achieved word error rates of 14.3% for semantic role labelling, 3.8% for chunking and 2.91% for POS tagging of English.

Wang and Manning (2013) compared linear neural network (Collobert et al., 2011) and conditional random fields. The work showed the mathematical similarity between CRFs and Linear Neural Networks and the performance was similar for named entity recognition (NER) and syntactic chunking.

Durrett and Klein (2015) introduced the combination of neural networks and conditional random fields in neural CRF Parsing of 10 world languages, i.e. English, Arabic, Basque, French, German, Hebrew, Hungarian, Korean, Polish and Swedish. Using neural networks as input to conditional random fields allows for the use of non-linear potential functions (modelled by the feed-forward NN) and the efficient inference provided by the CRF. Evaluating this approach in structured

language parsing showed superior performance over CRF only parsers on all the languages. Cotterell and Duh (2017) used this approach in named entity recognition and syllabification (Krantz et al., 2019) and Peng & Dredze (2016) used it in joint modelling of Chinese word segmentation and named entity recognition. Liu et al. (2020) also used the technique in segment recognition.

Wang et al. (2016) used Long Short Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) neural networks, a variant of a recurrent neural network, in the segmentation of Hebrew and Arabic. The authors tested three context-based LSTM architectures, i.e. a simple window LSTM, a multi-window LSTM and a bidirectional multi-window LSTM, in the prediction of morphological boundaries in a word. The approach casts segmentation as a sequence labelling exercise, with each letter in a word labelled as either the beginning of a morpheme (B), the middle of a morpheme (M), the end of a morpheme (E) or as a single character morpheme (S). This is a labelling used by Ruokolainen et al. (2013). After adjusting for hyper-parameter the best F1 Scores obtained were 95.2% for Hebrew and 96.5% for Arabic. Windows based LSTM showed better results than conditional random fields on a small percentage of the training data, but were comparable when trained on the full data set.

Kann and Schutze (2016) used encoder-decoder neural networks in the morphological re-inflection (MRI) of Arabic, Finnish, Georgian, German, Navajo, Russian, Spanish and Turkish. MRI is the production of a target inflection from a source form and source tag given a target tag. The source form does not have to be a base or lemma. If the source form is a lemma or base MRI is referred to as morphological inflection (MI). A simple example is pluralisation, “tree” -> “trees”. The approach was reported to outperform other systems for all the languages on the SIGMORPHON16 (Cotterell, Kirov, et al., 2016) shared task.

Kann et al. (2016) using neural encoder-decoder with a morph re-ranker to incorporate NLP features in their work on the canonical segmentation of English, German and Indonesian. Canonical segmentation handles morpheme fusion. This work was a porting of the work by Cotterell et al (2016), on the joint modelling of orthography and morphological segmentation, to neural networks. They achieved state of the art results in supervised canonical segmentation at the time with F1 scores of 82%, 87% and 96% for English, German and Indonesian respectively.

Zhang, Yu, et al. (2018) presented a joint sequence-to-sequence neural network model for Chinese segmentation and POS tagging using character level embeddings. The model is a transition system, which predicts a sequence of transition actions. The transition actions are APP and SEP, for appending the current character to the previous word or inserting a space before the current character. Two LSTMs, one for generating character sequence features from

character embeddings and the other for historical word tags, are used. The authors report superior performance at the time on five Chinese datasets.

In Zhang et al.'s (2018) segmentation of Chinese using neural networks the approach used embeddings of already identified words, embeddings of previous characters in the sentence sequence and already identified segmentation tags to predict what segmentation tag the next character should be allocated. The tags assigned to a character are B/I/E/S for identification that the character is at the beginning, intermediate, at the end of a word or is a single character word. The features were used in a transition based framework (Zheng et al., 2013). The approach provided state of the art results when discrete and neural features were used.

Zeldes (2019) compared Deep Neural Networks and Extra Trees Random Forests (ETRF) (Geurts et al., 2006) in the supervised segmentation of Hebrew, a morphologically complex language with templating inflection. The approach was about ensuring that a good set of features were used in the segmentation. The best performance came from the ETRF at an F1 Score of 97.08%.

Wang et al. (2019) incorporated unsupervised segmentation, including an unsupervised word goodness measure, into a supervised neural word segmentation system to improve its performance. Unlike other approaches, which use pre-trained word embeddings (Zhou et al., 2017; Yang et al., 2017), the approach learns the embedding during training. For unsupervised segmentation, the approach uses the accessor variety goodness measure metric and Nested Pitman-Yor Language model (NPYLM) (Mochihashi et al., 2009). The approach outperformed other neural network based Chinese segmenters on out of vocabulary (OoV) boosting recall by up to 4.7% and F1 Score by 0.6%.

Gong et al. (2019) presented to use of multiple criteria framework in the segmentation of Chinese word using switching LSTM neural networks. Each criteria is modelled using an LSTM, and the different LSTMs are linked to a switching unit. The LSTM feed a CRF for character tagging. The approach showed state of the art F1 Scores on eight corpora.

2.4.9 Conclusions on Supervised Morphological Segmentation techniques

The prevalence of a supervised technique seems to depend on the period. A language with similar complexity as isiXhosa has seen more use of variants of Conditional Random Fields (CRFs) than any other technique. Lately, however, neural networks, specifically Long Short Term Memory (LSTM) networks, are dominating, especially when combined with CFRs.

2.5 Semi-Supervised

As explained in section 2.3, semi-supervised algorithms use the best of both worlds, i.e. the structure in annotated data and the statistics of large raw data. Here a few studies that use limited annotated data with large raw data are discussed.

Lee et al. (2003) presented a **seeded semi-supervised algorithm** for the morphological analysis of Arabic. A corpus of 110 000 analysed words was used to estimate a **trigram model** of Arabic. The trigram model was then enhanced in an unsupervised manner with a corpus of 155 million words. The resultant model was tested on 28 449 words and achieved an accuracy of 97%.

Collobert and Weston (2008) presented a **single convolutional neural network architecture** that could analyse a sentence and produce POS tags, chunks, named entity tags, semantic roles and semantic similarity words for each word in the sentence, and give a likelihood of the grammatical and semantic correctness of a sentence. The language model is learnt from unlabelled data and the rest from labelled text. Of interest is the semantic representation of a word, the so called “embedding”. An embedding is an estimation of how a particular word semantically relates (synonym, homonyms, hypernyms etc.) to other words. This is modelled as a vector from the context of the word. Context is modelled using proximity where words with appearing in a similar context are deemed close to each other. The network achieved word error rates of 14.3% for semantic role labelling, 3.8% for chunking and 2.91% for POS tagging of English.

Kohonen et al. (2010) added semi-supervision to an unsupervised morphological segmenter Morfessor Baseline (Creutz & Lagus, 2002). Morfessor Baseline is discussed in section 2.6.3 unsupervised morphological segmentation using *Minimum Description Length*. This version of Morfessor was tested on English and Finnish achieving F1 Scores of 72.85% and 60.26% for the respective languages. This was achieved with the models trained with 10 000 labelled tokens for both languages and 62 185 728 unlabelled tokens for English and 36 207 308 unlabelled tokens for Finnish.

Shalonova and Golenia (2010) presented a weakly supervised approach to the word segmentation of Turkish and isiZulu verbs and nouns. The study used small training sets. The approach consists of two stages, i.e. the identification of the stem and the segmentation of the resultant prefix and suffix into morpheme. Stem identification is done through the **Supervised Stem Extraction (SSE) algorithm** presented in the paper, and the segmentation of affixes into morphemes is through the unsupervised segmentation algorithm **Graph Based Unsupervised Morpheme Segmentation (GBUMS)**. The approach was tested on Turkish and isiZulu nouns

and verbs. The approach showed superiority against Morfessor on nouns but did not perform as well on verbs for both languages. The average F1 Scores for isiZulu and Turkish were 86.2% and 77.85 respectively. F1 Scores for unsupervised Morfessor for both languages were below 29%.

Dreyer and Eisner (2011) extended their earlier work (Dreyer & Eisner, 2009) to a semi-supervised approach to the induction of inflectional paradigms from text. The model reconstructs latent word-bases and inflections. The approach tagged each word with a POS Tag, a base-word/lemma and an inflection. This allows for prediction of out of vocabulary (OoV) words. The approach uses log-linear models and Dirichlet Process Mixture models with Gibbs sampling and is initialised with a few paradigms. The approach was tested on German verbs with an inflection accuracy rate of 92.2%,

2.6 Unsupervised Morphological Segmentation works

As explained in section 2.3, unsupervised learning requires that the machine-learning tool use unannotated training data to model the process. This section covers the nine techniques used in the field and rounds off with a summary of the techniques.

2.6.1 Branching Entropy

The earliest works in unsupervised morphological segmentation used a form of accessor variety, where a morpheme boundary is identified by the possible number of letters that may follow that sequence of letters (Déjean, 1998; Feng, Chen, Deng, et al., 2004). One of the earliest works is that of Ando and Lee (2000) that used the variability of letters following a substring in text segmentation. This work presented a mostly unsupervised statistical word segmentation algorithm for Japanese. The algorithm was trained on unsegmented Japanese text. The algorithm used counts of **character n-grams**. It compared n-grams that straddle a proposed boundary to those that are adjacent. A proposed boundary is valid if the **n-grams that are adjacent to it are more frequent than those straddling it and they are above a particular threshold t**. The algorithm achieved a word precision level of 85%, a recall of 78%, and an F1 Score of 82%, which was better than rule-based morphological analysers available at the time, on the kanji, a type of Japanese characters. The morpheme level metrics reached an F1 Score of 80%.

The next major work on this variability of preceding and succeeding letters as a way to identify morpheme boundaries was done by Feng et al. (2004). They called the method for Chinese text segmentation Accessor Variety. The method considers characters before (predecessors) and after (successors) a string as accessor of the string. The study uses its own statistical criterion, accessor variety, to evaluate the probability of a string being a word/morpheme. The paper uses the intuition that the larger the context, i.e. accessor variety, in which a particular substring

appears in a corpus the higher the probability that it is a morpheme (Harris, 1942). For a particular string, the accessor variety was calculated from the number of different characters preceding it, the number of different characters succeeding it, and the number of sentences in which the string appeared at the beginning and at the end. The performance of this approach resulted in precision and recall of 86.48% and 89.78% respectively.

Jin and Tanaka-Ishii (2006) extended the measure of variability to using entropy instead of counts for unsupervised word segmentation. They called it Branching Entropy. The algorithm is based on the assumption that a point of peak entropy of successive characters in a sequence of characters is the location of a word boundary. The algorithm was tested on Chinese text and achieved a precision rate of 90% and a recall rate of 80%, which results in F1 Score of 84.7%.

Zhikov et al. (2010; 2013) presented an unsupervised word segmentation algorithm that uses branching entropy (Jin & Tanaka-Ishii, 2006) and Minimum Description Length (Rissanen, 1978). The algorithm was evaluated on Japanese, Thai and the CHILDES English corpus. The method achieved results comparable to state of the art at the time with significant improvement in computation speed. The method achieved boundary identification F1 Scores of 88.56%, 90.7% and 90.7% for Japanese, Thai and English respectively, and token F1 Scores of 75.42% and 71.31% for English and Japanese respectively.

Chen et al. (2011) proposed an approach to unsupervised word segmentation using a word induction criterion call WordRank. WordRank estimates the goodness of a word hypothesis using two metrics, the lowest mutual information of bigrams within a prospective word and some word boundary metrics. The approach was tested on English, Chinese and Japanese with F1 Scores of 78.6%, 78.8% and 66.3% respectively.

Wang et al. (2011) presented an unsupervised iterative approach to Chinese word segmentation called ESA. ESA stands for the three phases of the approach, i.e. Evaluation, Selection and Adjustment. The evaluation stage is used to estimate and calculate certain statistics from the corpora including Branching Entropies of the n-grams. Because the strategy uses n-grams of different lengths, the evaluation stage uses a normalisation method called balancing to ensure that comparisons among the different n-grams are proper. The selection stage uses the evaluation stage statistics in identifying locations for segmentation. Having segmented the corpus, the adjustment stage adjusts the statistics accordingly and the iteration starts again. The algorithm is complete when two consecutive iteration return the same results or a specified number of iterations is reached. The approach does require one to set the maximum n-gram length and an exponent in the statistical calculations, which both affect the performance. The approach was evaluated on several Chinese corpora with an average F1 Score of 78.1%.

Magistry and Sagot (2012) and Magistry (2012) presented work on the unsupervised segmentation of Chinese. The approach uses the variation of branching entropy (VBE) (Kempe, 1999) for a subsequence to determine how likely a substring is to be a morpheme. To compare VBE of different string lengths the VBEs are normalised by subtracting the average of VBEs for strings of a particular length. The average F1 Score among four Chinese corpora was 78.65%.

Magistry and Sagot (2013) extended the use of Variation of Branching Entropy (Magistry & Sagot, 2012; Magistry, 2012) and evaluated the use of Minimum Description Length (Rissanen, 1978) in Chinese word segmentation. The algorithm first uses the normalised VBE introduced by Magistry and Sagot (2012) for the initial segmentation. The approach is a further segmentation or merging of segments to reduce the Description Length. The initial work resulted in a reduction in the Minimum Description Length; however, the F1 Score of the segmentations was suboptimal. This meant that the reduction in the description length of a corpus does not necessarily result in the best segmentation of Chinese. Some linguistics informed constraints were introduced in the method resulting in major improvements to the F1 Score. The average F1 Score for the four Chinese corpora used was 80.925%. The description lengths were similar to those achieved by Zhikov et al. (2013).

Bergmanis and Goldwater (2017) presented an improvement to Narasimhan et al.'s (2015) unsupervised word segmenter, MorphoChain. The extension was that of making the system output a word morphological analysis instead of word segmentation. The enhancements made were in the features used. The check for presence in the training data was modified to have an out of vocabulary feature for parents that were not in the training data. The affix feature was modified to use Letter Successor Entropy (LSE), a variant of the right branching entropy, thresholded at an entropy level of 3.0 and a frequency of 100. Augmentation of the compounding feature was also done to the transformation features, with the inclusion of the head word in the feature. To reduce noise from low frequency types in the training data, the training data was pruned to retain words with a frequency of at least 50% of words having an embedding, and to finally decide on the usefulness of a word a modification of Whole Word Morphology (Neuvel & Fulop, 2002) strategy was used. The approach was tested on English, German, Turkish, Arabic, Finnish and Estonian with EMMA F1 Score percentages of 76.3, 56.8, 44.2, 33.1, 54.9 and 76.0 respectively. The EMMA evaluation metric is defined in section 2.1.5.4 *Evaluation Metric for Morphological Analysis (EMMA)*.

El-Kishky et al. (2018) presented SubwordMine for information retrieval. SubwordMine uses Letter Success Entropy to segment words into subwords. These subwords are then used in enriching word embedding that are tested in information retrieval. The work is refined in El-Kishky et al. (2019) with using binomial distribution in the morpheme statistical modelling.

2.6.2 Maximum Entropy Learning

MaxEnt is explained in section 2.4.3. This section provides a list of unsupervised segmentation works done using this technique. Training an unsupervised system using MaxEnt is different from using a supervised system. For a model with features f_i , λ_i are the feature model coefficients and the conditional probability distribution of the outcome Y given the input X is defined as.

$$p(Y|X) = \frac{e^{\sum_i \lambda_i f_i(X,Y)}}{Z}, \quad (2-20)$$

$$Z = \sum_{x,y \in X \times Y} e^{\sum_i \lambda_i f_i(x,y)} \quad (2-21)$$

Unsupervised learning maximises the log-likelihood of the observations.

$$L_\theta(X) = \log \sum_Y P(X, Y) \quad (2-22)$$

This is achieved using gradient descent. The gradients of the feature weights are calculated as:

$$\frac{\partial}{\partial \lambda_i} L_\theta(X) = E_{P(Y|X)}[f_i] - E_{P(X,Y)}[f_i] \quad (2-23)$$

Where $E_P(f)$ denotes the statistical expectation of function f under that probability distribution P .

De Pauw and Wagacha (2007) presented a proof of principle experiment on the unsupervised morphological analysis of a resource-scarce Bantu language Gĩkũyũ. The study used maximum entropy learning (Ratnaparkhi, 1998) to induce morphological features. The algorithm attained an F1 Score of 64.6% in prefix retrieval.

Sarikaya and Deng (2007) used the JMMLM (presented in section 2.4.3 on supervised *Maximum Entropy Modelling*) in the Arabic-English statistical machine translation. This resulted in a BLEU translation score of 30.41% when combined with Morpheme trigram features, an improvement of 0.79% from just the Morpheme trigram language model.

Narasimhan et al. (2015) presented an unsupervised morphological analyser that is also based on log-linear models, called MorphoChain. Some the main features used were semantic similarity, candidate affix distributions, affix correlations and morpheme attachment transformations. Normalisation of the objective function, the likelihood function, was replaced with contrastive estimation, which reduced the normalisation from a sum over all the words to a sum over the word neighbourhoods. Contractive estimation is explained in section 2.6.8: *Contrastive Estimation*. The

approach was tested on English, Turkish and Arabic with F1 Scores of 76.2%, 61.2% and 79.9% respectively.

2.6.3 Minimum Description Length

Descriptive Length Gain (DLG) is a criterion used in unsupervised lexical acquisition. It is a measure of how good a substitution is at reducing the minimum description length (MDL) of a corpus. Consider an N long corpus $D = m_1..m_N$ which is made of types $m_i \in M$, where M is the dictionary of types. If $\#m$ is the count of $m \in M$ in the corpus, the minimum description length is calculated in two parts i.e. the length of the corpus given the types, $L(D|M)$, and the length of the code book $L(M)$.

$$\begin{aligned}
 DL(D|M) &= L(D|M) + L(M) \\
 &= - \sum_{m \in M} \#m \cdot \log_2 p(m) - \sum_{m \in M} \log_2 p(m) \\
 &= - \sum_{m \in M} \#m \cdot \log_2 \left(\frac{\#m}{N} \right) - \sum_{m \in M} \log_2 \left(\frac{\#m}{N} \right)
 \end{aligned} \tag{2-24}$$

The description length gain (DLG) of modelling the same corpus with morphs $m' \in M'$, where M' is the dictionary of morphs, is calculated as follows:

$$DLG(D) = DL(D|M) - DL(D|M') \tag{2-25}$$

An algorithm that uses DLG, maximising the DLG, was developed by Kit (1998) and was evaluated (Kit & Wilks, 1999) on the English Brown corpus resulting in a precision level of 79.33% and recall of 63.01%.

Goldsmith (2000; 2001) presented the unsupervised learning of morphological segmentation of European languages using the minimum description length (MDL) as the global segmentation evaluator, in an algorithm called Linguistica. Minimum description length optimisation involves minimising the MDL. Linguistica produced analysis comparable to the human morphologist for English, French, Spanish, Latin and Italian. Results were only specified for English and French with accuracies of 82.9% and 83.3% respectively.

Teahan et al. (2000) presented a supervised scheme for estimating morpheme boundaries in Chinese text using text compression techniques. Text compression is similar to minimisation of description length. The error rate, recall and precision achieved by the technique were 10.8%, 93.4%, and 89.6% respectively.

Creutz and Lagus (2002) presented methods for unsupervised segmentation of words into morpheme-like units. The first method used the Minimum Description Length (MDL) optimisation and the second method used Maximum Likelihood (ML) optimisation, called Morfessor-Baseline and Morfessor-Categories ML respectively. Maximum Likelihood optimisation minimises the log-likelihood of the data given a model M' . Below N' is the number of segments in the new segmentation of the data D .

$$L_{\theta}(D|M') = - \sum_{m' \in M'} \#m \cdot \log_2 \left(\frac{\#m'}{N'} \right) \quad (2-26)$$

The Morfessor variants performed well compared to the state-of-the-art system at the time, Linguistica, on both Finnish and English. Accuracy results for the MDL algorithm, the ML algorithm and Linguistica were 49.6%, 47.3% and 43.1 % respectively on Finnish.

Creutz and Lagus (2005; 2007) presented an algorithm for unsupervised word segmentation of words called Morfessor Categories-MAP. The algorithm used the **probabilistic maximum a posteriori (MAP) model**. MAP models maximise the posterior probability of the model M given the data D , $P(M|D)$, which can be calculated as follows:

$$P(M|D) = \frac{P(D|M) \cdot P(M)}{P(D)} \quad (2-27)$$

The algorithm was tested on Finnish and English and compared to Linguistica (Goldsmith, 2001), its predecessors Categories-ML and Morfessor Baseline (Creutz & Lagus, 2002). Very good results were obtained for Finnish, but it showed an F1 Score lower than the original Categories-ML for English.

Gronroos and Virpioja (2014) presented a new variant to Morfessor (Creutz & Lagus, 2002; Creutz & Lagus, 2005), FlatCat which applies Hidden Markov Model to morphological segmentation. FlatCat did not reach the accuracy of Morfessor Categories-MAP, but performed better compared to systems implementing Conditional Random Fields (CRFs).

Golénia et al. (2009) presented an unsupervised word segmentation algorithm called Unsupervised Graph Decomposition (UNGRADE). The algorithm operates in two phases. The first phase identifies a pseudo stem using a window based Minimum Description Length algorithm. The second phase takes the resultant affixes (prefix and suffix) and segments them into the constituent affixes (morphs) using a graph algorithm called Graph-Based unsupervised sequence segmentation (Shalnova et al., 2009). UNGRADE was tested on English, German, Finnish, Turkish and Arabic with F1 Scores of 37.11%, 33.44%, 36.51%, 36.64% and 54.37% respectively.

2.6.4 Paradigmatic Clustering

A paradigm is a grouping of words according to their main meaning (Booij, 2012). Normally words in a paradigm group share a base word from which they are all derived. Several metrics can be used to measure the similarity between words to cluster them into a paradigm.

Gaussier (1999) presented an unsupervised method for learning suffixation rules from an inflectional lexicon. This is done by **clustering words into relational families**. The similarity measure in this study is based on the commonality of stems between words resulting in suffix pairs, and the prevalence of the suffix pairs. Such rules were then used to identify even more suffixes. This was done by extracting derivational morphology from words in a relational group. The algorithm was evaluated on English with an accuracy of 85%.

De Roeck and Al-Fares (2000) presented an unsupervised algorithm for clustering Arabic words that share the same root. The algorithm uses the Dice and Jaccard coefficients (Manning & Schütze, 1999:299) with modifications for the language to measure similarity between words. The algorithm achieved a maximum clustering accuracy of 94.04%.

Lignos et al. (2009) introduced an unsupervised morphological analyser that learns rules from unannotated text. The rules are stored as base/lemma and transform pairs. The work was then improved by Lignos (2010) to better handle compound words and an addition of transforms that are based on other transforms. The best segmentation F1 Scores achieved were 56.62% for English, 48.93% for Finnish, 38.78% for German and 33.31% for Turkish.

Muller and Schutze (2015) experimented with different word representations in morphological tagging. They compared Singular Value Decomposition, Language Model based word clustering, neural network embeddings and accumulated tag counts, a form of sparse representation. The work was done for six languages, i.e. Czech, English, German, Hungarian, Latin and Spanish. They found that **Language-Model based clusters** outperformed other representation for POS tagging in all the languages evaluated.

Mendez-Cruz et al. (2016) presented an unsupervised morphological segmentation method that uses **affixality measurements** (Méndez-Cruz et al., 2012). Affixality measurement is made up three statistics measurements that together give a score of a segment being a morpheme. For a particular segment, the statistics used are the entropy of segments attaching to it similar to right-branching entropy, an economy of a segment which is a measure of how many other morphemes can replace the segment in its context, and the number of squares of the segment which is a measure of how many times it is found in combination with three other segments. The method

outperformed other unsupervised morphological segmentation algorithms (Morfessor and ParaMor) in the segmentation of Spanish words.

Chavula and Suleman (2017) presented work on the unsupervised morphological segmentation of Bantu languages using morphological clusters created using a weighted word similarity measure. The similarity measure is based on Ordered Weighted Aggregator operators. The Chavula-Suleman similarity measure is biased towards the similarity of the middle characters in strings and reduces the importance of similarity the further away from the centre. It is based on the assumption that stems are distributed more towards the centre in Bantu words. This was then used to cluster words into paradigms, identify stems and eventually identify affixes. The study measured the purity of the clusters identified, and the morphological segmentation induced from the stems identified. The tests were done for two Malawi languages, Citumbuka and Chichewa. The clustering purity results obtained were 0.7 and 0.79 for the respective languages, compared to 0.59 and 0.66 using the Dice coefficient. Precision for morphological segmentation was measured at 53% for Chichewa and 54% for Citumbuka, an improvement for Citumbuka when compared to Morfessor-Baseline whose precision was 50%, but a reduction from for Chichewa' Morfessor-Baseline precision of 58%.

2.6.5 Context based segmentation

Whilst paradigmatic clustering groups words according to their paradigm, identifying the root and the segmentation, context based approaches group words according to their context. The idea is that words that appear in the same context are probably of the same part of speech and therefore are inflected in a similar manner. Therefore, in the cases below the approach is that of identifying POS common inflections thereby being able to analyse the words.

One of the earliest works that used this approach was Déjean (1998). Instead of using a word context, this work used character level context. The paper describes an unsupervised method for the automatic discovery of syntactic structures from an untagged corpus in twenty languages. The structures discovered are morphemes, chunks, relationships between chunks and clauses. The method differentiates between frequent and infrequent morphemes and finally segments words in the corpus. The method has three steps. The first step identifies the top 100 frequent morphemes using **boundary identification**. A boundary between morphemes is identified by the number of different letters that follow a sequence of letters, similar to Accessor Variety (see section 2.6.1: *Branching Entropy*). Having identified the top 100 morphemes, the method then uses words that have these morphemes to find other strings that replaced these top 100 morphemes, assuming those strings also to be morphemes. Having identified this extended list

of morphemes, the method then can analyse the entire corpus. The training process used 100 000 words. Déjean mentioned that 5% of the words were not segmented by the method.

Yarowsky and Wicentowski (2000) presented an unsupervised algorithm that induces inflectional morphology of English from a corpus. The algorithm uses distributional patterns in the monolingual text. The algorithm combined techniques for **relative corpus frequency, contextual similarity, string similarity and induced flectional probabilities**. The algorithm was tested on English past tense cases and achieved an accuracy of 99.2%.

Schone and Jurafsky (2000) presented an unsupervised morphology induction algorithm that uses **Latent Semantic Analysis (LSA)** (Deerwester et al., 1990). The intuition of LSA is that the meaning of a word can be induced from its contextual use. Statistical context is inferred from words that frequently appear with the word in question. This means that words that appear in the same context should have similar meanings. This analysis is extended to the use of identifying morphemes by Schone and Jurafsky. The approach rivalled the state-of-the-art at the time, *Linguistica*. The algorithm was tested on the English CELEX corpus with an F1 Score of 84.3% compared to *Linguistica*'s 81.6%.

Schone and Jurafsky extended their initial work to induce inflectional morphologies by adding **orthography and syntactic distribution**. The improved algorithm was tested in the CELEX corpora for German, Dutch and English. The new algorithm achieved F1 Scores of 88.1% for English, 92.3% for German and 85.8% for Dutch. This was well above the results of 85.2%, 88.3% and 82.2% for English, German and Dutch respectively from the earlier Schone and Jurafsky (2000) algorithm.

Belkin and Goldsmith (2002) presented experiments that use unsupervised statistical methods to infer syntactic categories, morpheme identities, from an untagged corpus. The method derives categories using the **occurrence of words adjacent to other words**. It is a form of word clustering that is based on **co-occurrence**. The heart of the algorithm is the use of eigenvectors of the bigram graph. The work does not provide results other than the list of English suffixes that the method identifies.

2.6.6 Non-parametric Bayesian techniques

The techniques above are parametric. Whereas parametric methods assume a fixed set of features and parameters, non-parametric Bayesian methods do not. These mostly use Markov Chain Monte Carlo simulation with Gibbs (Geman & Geman, 1984) sampling inference. Techniques that show promise are Pitman-Yor process based models (Pitman & Yor, 1997) and Dirichlet processes (Ferguson, 1973).

Goldwater et al. (2005) presented a framework that augments standard generative models with an adaptor that modifies the distribution to generate power-laws. Inference is done using the Gibbs sampler (Geman & Geman, 1984). The adaptor used the Pitman-Yor process and improved the performance of verb suffix identification using unsupervised learning. The paper however does not provide empirical results. This work was applied to probabilistic context free grammars to create **adaptor grammars** by Johnson et al. (2007). Adaptor grammars are an unsupervised extension to Probabilistic Context Free Grammars (PCFGs) that lightly break the independence and introduce links with previous text in the string being analysed.

Johnson (2008a) used adaptor grammars in the segmentation of Sesotho speech. Different grammar models were used with the word-morph model giving the best word F1 Score at 55.6% and the colloc2-syll model producing the best morpheme F1 Score at 39.3%. Adaptor Grammars were also used by Johnson (2008b) in the unsupervised word segmentation of English speech. The best word segmentation F1 Score was measured 78%. Botha and Blunsom (2013) evaluated the use of adaptor grammars in the identification of roots and morphological segmentation of non-concatenative languages, particularly languages with templatic morphology, i.e. Hebrew and some Arabic variants. The approach resulted in an accuracy of 74% for identifying Hebrew roots and an F1 Score of 78.1%, an improvement to previous results in these languages, for segmentation.

Sirts and Goldwater (2013) also tested the use of **adaptor grammars** for morphological segmentation in different supervision modes: supervised, semi-supervised and unsupervised training. The unsupervised model for learning the grammar rules introduced by Sirts and Goldwater included using over-articulated meta-grammars that are then filtered with a small labelled set to optimise boundary identification. The approach was tested on English, Finnish, Turkish and Estonian with segmentation F1 Scores of 74.4%, 70%, 61.7% and 71.7% respectively. The implementations outperformed Morfessor and Morsel (2010).

Mochihashi et al. (2009) extended the use of the Pitman-Yor process to a Nested Pitman-Yor Language Models (NPYLM) in the segmentation of English, Chinese and Japanese speech. This work also introduced the blocked Gibbs sampler with dynamic programming for inference. The best results were an F1 Scores of 75.7% for English, an average F1 Score of 81.2% for the two Chinese corpora and 66.6% for Japanese.

Neubig et al. (2010; 2012) presented work on unsupervised learning from continuous speech instead of text of a language model and word boundaries. The work used a novel Weighted Finite-state Transducer (WFST) based sampling technique to simultaneously learn word boundaries

and the n-gram language model in an unsupervised way. The work used the Hierarchical Pitman-Yor language model. The work was tested on Japanese with a phoneme error rate of 8.1%.

Another attempt at translating to morphologically rich languages (MRL) saw the use by Chahuneau et al. (2013) of a discriminative model to predict inflection of targets from rich source-side annotations. In this approach, the morphological analysis of the morphologically rich language used an **unsupervised Bayesian model** instead of a supervised analyser. The baseline translator was a hierarchical phrase-based statistical machine translator (HP-SMT) (Chiang, 2005) using a 4-gram language model. A word class-based language model was one enhancement and synthetic inflection was the other enhancement. This approach showed translation improvements when translating from English to Russian, Hebrew and Swahili. The BLEU results for translation from English using baseline translations, class-based LM and synthetic inflection were 14.7%, 15.7% and 16.2% for Russian, 15.8%, 16.8% and 17.6% for Hebrew, and 18.3%, 18.7% to 19% for Swahili. In this approach translation is done to the stem of the MRL, the stems are then inflected to form a sentence. Unsupervised morphological analysis uses the **Dirichlet** distribution to model morpheme distribution, the **Beta distribution** to model affix distribution, a **geometric distribution** to model the vocabulary and the **Gibbs sampling** (Geman & Geman, 1984; Brooks, 1998) to model word segmentation. The approach is similar to using adaptor grammars (Goldwater et al., 2006; Johnson, 2008a).

Cao et al. (2014) proposed a four-level Dirichlet Process (DP-4) based model for learning word segmentation (WS), keyword extraction (KE) and term extraction (TE) for Chinese. For Dirichlet Process based model a sampler is required. In this paper the authors introduced the sentence-wise Gibbs sampling technique and used the dynamic programming strategy proposed by Mochihashi et al. (2009). The algorithms achieved an F1 Score of 82.1% for word segmentation, and 37.4% recall for out of vocabulary words, and accuracy levels of 70.4% and 57.1% for term extraction and keyword extraction respectively.

Zhang et al. (2015) presented an approach to simultaneously/joint segmentation, POS tagging and dependency parsing. The system was tested on Arabic and Chinese and use a **randomised greedy algorithm** based on one that was initially designed for parsing only (Zhang et al., 2014). The algorithm showed improvements to the state of the art on three data sets. Performance on this study resulted in the following percent F1 Scores:

	Segmentation	POS Tagging	Parsing
Modern Standard Arabic	99.52	97.43	87.23
Classical Arabic	94.35	84.44	-
Chinese	98.04	94.47	82.01

Uchiumi et al. (2015) presented the Pitman-Yor Hidden Semi-Markov Model (PYHSMM) for the joint segmentation and part of speech tagging. This model is an extension of the Nested Pitman-Yor Language Model (NPYLM) introduced by Mochihashi et al. (2009) to include hidden states. The method was tested on Japanese, Chinese and Thai. Segmentation average accuracy was measured at 71%, 82.37% and 82.1% for Japanese, Chinese and Thai respectively and POS tagging for Chinese gave an average precision of 46.9% across the data sets.

2.6.7 Neural Networks

Recent neural networks architectures have provided good performance even though they are not strictly unsupervised. There are two techniques used in neural networks that have boosted their performance, i.e. word and character-level embeddings (Mikolov, Corrado, et al., 2013) and deep architectures (Bengio & Senécal, 2003).

Collobert (2011) proposed a discriminative algorithm for natural language parsing based on **deep recurrent convolutional neural networks**. The method used the **embedding modelled** in Collobert and Weston (2008). The algorithm performed similarly to existing algorithms at an F1 Score of 89.1% for English.

Collobert et al. (2011) proposed a **unified neural network architecture** for use in natural language processing tasks including POS tagging, named entity recognition, segmentation and semantic role labelling. The proposed neural network model learns internal representation of a language from unlabelled training data. The model was tested on a tagging system with satisfactory results. The system vastly outperformed benchmarks when it comes to computing resource usage and speed. The model achieved 77.24% accuracy for POS tagging, and F1 Scores of 94.29% for parsing, 89.31% for named entity recognition, and 77.92% for semantic role modelling for English.

Ling et al. (2015) introduced a model for creating word embeddings from characters, C2W. The approach used **bidirectional Long Short-Term Memories (LSTMs)** (Hochreiter & Schmidhuber, 1997; Graves & Schmidhuber, 2005). LSTMs are neural networks that have memory whose decay can be controlled. The approach was tested on language modelling and part-of-speech tagging for English, Portuguese, Catalan, German and Turkish. The language modelling perplexity results

for the respective languages were 57.39, 40.92, 34.92, 41.94 and 32.88. The approach showed superior performance for all the languages except Turkish in tagging. The POS tagging percent accuracy results were 97.36%, 97.47%, 98.92%, 98.08% and 91.59% for English, Portuguese, Catalan, German and Turkish respectively.

Word embeddings are a representation of words as points in a high dimensional geometric space such that similar words are closer to each other (Mikolov, Corrado, et al., 2013). Lin et al. (2015) tested the use of this concept in unsupervised POS induction. The system used multivariate Gaussian distributions over word embeddings to model POS clustering. They found superior performance when using **simple Gaussian Hidden Markov Models** with a **Conditional Random Field** auto-emitter. The study used the V-measure metric, a metric that evaluates the homogeneity and completeness of a cluster (Rosenberg & Hirschberg, 2007).

Instead of using complete words in generating the word embedding, Kim et al. (2016) created a **neural language model** that only takes character level sequence input. The model does predictions at word-level. The model is said to be able to encode rich semantic and orthographical features. The model was on par with existing English models and outperformed existing models for some morphologically rich languages, i.e. Arabic, Czech, French, German, Spanish and Russian.

Wang et al. (2017) used a Neural Network named “Sleep-Wake Networks” (SWAN) inspired by connectionist temporal classification and sequence transducers. The approach was tested on the segmentation of English text and achieved 30.5% character error rates (CER).

2.6.8 Contrastive Estimation

The idea behind contrastive estimation is to replace the normalisation denominator in probability density estimation, the partitioning function, with one that does not sum across the entire dataset but a subset of the dataset that is close to the event that needs to be normalised. For example, the normalisation function Z in Maximum Entropy Learning (see section 2.6.2) can be intractable for large data sets but can be efficiently estimated using contrastive estimation.

Smith and Eisner (2005) introduced contrastive estimation, an unsupervised machine learning technique that uses Conditional Random Fields and data perturbation to improve modelling accuracy. The model uses perturbation to create false samples. The paper considered different perturbation methods including transposition of letters, deletion and both. The best results came from transposition. The approach was tested on POS tagging for English and achieved 90.4% accuracy.

Poon et al. (2009) used contrastive estimation with log-linear models for unsupervised word segmentation of Arabic. According to the paper, the algorithm achieved a reduction in the F1 Error rate of 11% compared to Morfessor-Baseline. The F1 Error is defined as $1 - F1\ Score$.

Narasimhan et al. (2015) presented an unsupervised morphological analyser that is also based on log-linear models. Some the main features used were semantic similarity, candidate affix distributions, affix correlations and morpheme attachment transformations. Normalisation of the objective function, the likelihood function was replaced with contrastive estimation. The approach was tested on English, Turkish and Arabic with F1 Scores of 76.2%, 61.2% and 79.9% respectively.

Zoph et al. (2016) presented work on the use of noise contrastive estimation (NCE) to improve speed in neural language modelling. NCE converts an unsupervised method to a supervised model by introducing false/wrong/noise injected samples onto the training set. This improves estimation by treating the training process as a classification problem between the true samples and the false samples. Ordinarily, a number of false samples are created for each true sample, increasing the training set size, thereby slowing down training. The enhancement introduced in this paper was the use of the same false sample for a batch of true samples. This approach was tested on language modelling a large Arabic corpus resulting in the perplexity of 43.2 compared to 51.3 from an earlier study on the same corpus. The enhanced language model was used in an Arabic-English translation pipeline resulting in a BLEU score of 41.0% compared to 39.7% from a previous study on the same translation corpus.

2.6.9 Language Model based segmentation

A language model is a model that assigns probabilities to sequences of words $p(w_{1..N})$ (Jurafsky & Martin, 2000:36). This is achieved by using the chain rule of probability and assuming that the conditional probabilities are a Markov Chain of order n , such that

$$p(w_{1..N}) = \prod_{i=1}^N p(w_i | w_{i-n+1..i-1}). \quad (2-28)$$

Because the n-gram probabilities are determined from type counts of a corpus, which is finite, there are valid types that are not in the corpus. This is referred to as sparseness as the counts of the missing types would be zero, an incorrect estimate. In addition, counting from the corpus produces poor estimates for near-zero probability types (Jurafsky & Martin, 2000:207). Compensating for these deficiencies is called smoothing. There are a number of smoothing methods with the simplest being Good-Turing Discounting and the best performing smoothing

being the modified Kneser-Ney smoothing (Chen & Goodman, 1998). Chen and Goodman provide good detail of the different smoothing techniques. Heafield et al. (2013) provide a scalable implementation of the algorithm. Pickhardt et al. (2014) extended the modified Kneser-Ney smoothing to skip-grams.

Language models are used in handwriting recognition, augmentative communication for the disabled, spelling error correction and other tasks (Jurafsky & Martin, 2000:192). Sarikaya et al. (2007) used a language model, Joint Morphological-Lexical Language Modelling (JMLLM), for speech recognition. Sarikaya and Deng (2007) used the JMLLM in the Arabic-English statistical machine translation.

Most language models are parametric, meaning that they have a set depth, a maximum n-gram size. However, there is a need for non-parametric language models, especially if one wants to relax the Markov order or to have a variable Markov order. The Dirichlet Language Model (Mackay, 1995) and the Nested Pitman-Yor Language Models (NPYLM) (Teh, 2006b) are such models. The NPYLM is a generalisation of the Dirichlet Language Model and defines the interpolated Kneser-Ney model under certain conditions, specifically combining type and token frequencies (Goldwater et al., 2005). Non-parametric models use statistical processes to generate the model, and the Pitman-Yor and Dirichlet are such processes. Work on language models using other non-parametric statistical processes continues. Studies in segmentation using non-parametric methods are detailed in section 2.6.6: *Non-parametric Bayesian techniques*.

Recently, neural language models have become the benchmark in language modelling (Bengio & Senécal, 2003; Mikolov, Corrado, et al., 2013; Mikolov, Yih, et al., 2013). Kim et al. (2016) extended the ideal to a **neural language model** that only takes character level sequence input. The model does predictions at word-level. The model is said to be able to encode rich semantic and orthographical features. The model was on par with existing English models and outperformed existing models for some morphologically rich languages, i.e. Arabic, Czech, French, German, Spanish and Russian. Mikolov, Corrado, et al. (2013) improved on previous neural word embeddings by introducing techniques for learning word vectors from large data sets efficiently. The architecture is called word2vec and has seen extensive use in neural language modelling.

Non-parametric language models use sampling and are heavy on computation. Neural language models require vast amounts of data and they are also computation-intensive, hence the persistent use of classical language modelling techniques, as used in Erdmann et al. (2019). This work used a language model guided by language heuristics to segment a number of variants of Arabic resulting in an average accuracy of 80.5%. Teh (2006a) showed that the hierarchical

Pitman-Yor language model recovers the exact formulation of the interpolated Kneser-Ney smoothing algorithm.

In classic terms, the probability of a token, say a word $w = w_i$, given a context, say $h = w_{i-n+1..i-1}$ is calculated as

$$p(w|h) = p(w_i|w_{i-n+1..i-1}) = \frac{\#hw}{\#h} = \frac{\#w_{i-n+1..i}}{\#w_{i-n+1..i-1}} \quad (2-29)$$

where $\#x$ is the number of times the n-gram x is seen in the data, the count of x .

Of the classical language smoothing techniques, the Good-Turing is one of the simplest. The Good-Turing Discounting is based on the assumption that bigrams are binomially distributed (Jurafsky & Martin, 2000:215), and it does a re-estimation of the n-gram probability of scarce tokens from the number of n-grams with higher counts. The smoothed count (c^*) is:

$$c^* = (c + 1) \cdot \frac{N_{c+1}}{N_c} \quad (2-30)$$

where c is the unsmoothed count and N_c the number of n-grams that occur c times.

The modified Kneser-Ney smoothing is an interpolated smoothing technique. The easiest interpolated smoothing method is called Deleted Interpolation where the conditional probability, $p(w_i|w_{i-2}, w_{i-1})$, is calculated from unigrams, bigrams and trigrams as

$$\hat{p}(w_i|w_{i-2}, w_{i-1}) = \lambda_1 p(w_i|w_{i-2}, w_{i-1}) + \lambda_2 p(w_i|w_{i-1}) + \lambda_3 p(w_i) \quad (2-31)$$

where

$$\sum_i \lambda_i = 1. \quad (2-32)$$

The easiest choice is $\lambda_1 = \lambda_2 = \lambda_3 = 1/3$. A generalised interpolation technique is such that

$$\hat{p}(w_i|w_{i-n+1..i-1}) = \lambda_0 p(w_i) + \sum_{j=1}^{n-1} \lambda_j p(w_i|w_{i-j..i-1}). \quad (2-33)$$

Considering recursion, (2-33) can be rewritten to

$$\begin{aligned} \hat{p}(w|h) &= \hat{p}(w_i|w_{i-n+1..i-1}) = \alpha(w|h) + \lambda(h) \cdot p(w|\underline{h}) \\ &= \alpha(w_i|w_{i-n+1..i-1}) + \lambda(w_{i-n+2..i-1}) \cdot p(w_i|w_{i-n+2..i-1}). \end{aligned} \quad (2-34)$$

where $\alpha(x)$ is the discounting function and $\lambda_1(x)$ is the interpolation function and $\underline{h}$ is the history truncated by one token. Equation (2-34) is the model that modified Kneser-Ney (mKN) smoothing uses.

Modified Kneser-Ney (mKN) smoothing uses three innovations in calculating a conditional probability at a particular level, say n . The first one is adjusted counts. Instead of using the prevalence of a sequence as a count, the number of times the sequence is seen as the first n words in sequence of length $n + 1$ is used where possible. Second, the counts are discounted. There are three discounts, for each level. One for counts of 1, another for counts of 2 and the third for counts larger than 2. These discounts are based on the total counts for each level. The third innovation is the recursion with dependence on only one level lower. In this way a conditional probability calculation at a particular level requires adjusted counts based on one level higher, discounts calculated from statistics at its level and conditional probabilities from one level lower.

2.6.10 Summary of Unsupervised Morphological Segmentation

A comparison of the performance of the different techniques would have enhanced the study and the choice of technique to use, however, the curse of dimensionality looms large in the cohort of unsupervised segmentation papers cited in this study. There are many techniques used, in several different languages, evaluated on differing testing corpora and using a plethora of evaluation metrics. This leaves persistence of use as a better criterion for choosing the technologies to use.

Accessor variety is one of the earliest techniques used in unsupervised morphological segmentation. This evolved further to use mutual information (Sun et al., 1998; Ye et al., 2013), and different forms of Branching Entropy (Ando & Lee, 2000; Sun et al., 1998). This technique continues to be used in a number of variants (Wu & Zhao, 2018; El-Kishky et al., 2019).

Minimum Description Length (MDL) (Rissanen, 1978) has seen extensive use, primarily as a measure of fit of the training data to heuristic models and statistical models (Kit, 1998; Creutz & Lagus, 2002; Golénia et al., 2010).

Another popular approach is clustering and paradigmatic models. A paradigm is a grouping of words according to their form/meaning correspondence (Booij, 2012). This involves clustering related words into a paradigm using a similarity measure, identifying the stem, and considering the rest as a sequence of affixes (Gaussier, 1999; Chavula & Suleman, 2017). The similarity measures used are Latent Semantic Analysis (Deerwester et al., 1990), Dice and Jaccard coefficients (Manning & Schütze, 1999), affixality measurements (Méndez-Cruz et al., 2016) and Ordered Weighted Aggregator operators (Chavula & Suleman, 2017). Word context is also

another technique that is used to identify similar words (Belkin & Goldsmith, 2002; Schone & Jurafsky, 2000).

Non-parametric Bayesian techniques have also shown promise, including Pitman-Yor process based models (Goldwater et al., 2005; Uchiumi et al., 2015) and adaptor grammars (Sirts & Goldwater, 2013). These use Markov Chain Monte Carlo (MCMC) simulation with Gibbs Sampling (Geman & Geman, 1984) for inference. Contrastive Estimation (Smith & Eisner, 2005; Narasimhan et al., 2015) is another non-parametric model that is showing elegance and promising results.

A number of studies have used a combination of the above techniques and measures (Schone & Jurafsky, 2000; Nogwina, 2016) and recently language heuristics have been used to guide the segmentation.

2.7 Techniques and Benchmark of choice for isiXhosa segmentation

In selecting techniques to use in the rest of the study, simplicity of implementation and high performance are selected as the prime criteria.

The requirements of the simplicity of implementation, coupled with the high performance, make the branching entropy unsupervised techniques, especially the work by Magistry and Sagot (2013) on the use of the Normalised Variation of Branching Entropy, the preferred types of candidates for segmenting isiXhosa. This is a variant of the accessor variety which is seeing recent use in the text segmentation. Thus this technique will be investigated in the following chapters, extended where necessary, implemented and evaluated in the unsupervised segmentation of isiXhosa.

Another class of unsupervised segmentation techniques that meet the two previous criteria in segmenting a complex language, are hierarchical non-parametric character-level language models applied to unsupervised text segmentation.

Ordinarily, non-parametric character-level language model based techniques use a hierarchical Pitman-Yor process for language modelling and Gibbs sampling for inference. This is however demanding on computational resources. A possible modification is worth investigating. Instead of using non-parametric Bayesian language modelling one could use modified Kneser-Ney (mKN) smoothed character-level language models to estimate the hierarchical Pitman-Yor process. The mKN was shown to estimate the hierarchical Pitman-Yor language models by Teh. The search space could then be reduced using language heuristics, similar to adaptor grammars. This approach was used by Erdmann et al. (2019). As such, in further chapters, work on this approach

will be investigated, appropriate language heuristics will be identified, appropriate generative language models will be identified, implemented and evaluated in the unsupervised segmentation of isiXhosa.

For a benchmark, availability of the algorithm, wide use in the literature and suitable license conditions are pre-requisites. A tool that has been used as a benchmark in several studies in this literature review (de Pauw & de Schryver, 2008; Kohonen et al., 2010; Shalanova & Golenia, 2010; Chavula & Suleman, 2017; Méndez-Cruz et al., 2016) is Morfessor-Baseline (Creutz & Lagus, 2002). This tool is also freely available and accordingly, this study will use it as a benchmark.

2.8 Conclusions

This chapter examined several methods used in the morphological segmentation of text. The chapter started by detailing metrics used in evaluating text segmentations. For evaluation purposes, accuracy and the F1 Score are the standard metrics of use and are explained in this chapter.

That was followed by a review of the techniques used in morphological segmentation. The review starts by looking at knowledge and rule-based systems. The rest of the chapter focuses on machine learning based segmentation including supervised, semi-supervised and unsupervised segmentation techniques.

Several supervised techniques are used in unsupervised segmentation scenarios with the appropriate optimisation technique. Because computation in unsupervised methods can be intractable, the use of non-parametric Bayesian methods is also common. A new technique that attempts to address that intractability is contrastive estimation and its variant, noise contrastive estimation.

The versatility of Artificial Neural Networks has enabled the use of several ANN techniques in text segmentation, including recurrent neural network, convolutional neural network and auto encoder. Recently word and character-level embeddings have provided major improvements.

In choosing techniques that could be used to segment isiXhosa, the criteria specified of training on raw text, ease of implementation, use in complex languages and good morphological segmentation performance leads to the branching entropy and heuristics guided language model-based segmentation techniques. The Normalised variation of Branching Entropy (Magistry & Sagot, 2013) is an unsupervised technique that has performed well in Chinese, a language without word delimiters, and has ease of implementation. Using heuristics to guide a language

model segmenter (Erdmann et al., 2019) showed good performance in Arabic, a complex language, the language model can be implemented using modified Kneser-Ney smoothing to approximate non-parametric language modelling without computationally heavy MCMC, and is unsupervised.

For evaluation, as stated in Sect 2.1.6: *Conclusions on Segmentation Metrics* the morpheme boundary identification metrics are the best fit for the work of unsupervised surface segmentation. In addition, overwhelmingly the studies cited use either accuracy, F1 Score or both in their evaluation. As such this work will follow that precedence and use accuracy and F1 Score in the measurements.

For a benchmark, the chapter identifies Morfessor-Baseline, because it is a surface segmenter that continues to be used as a benchmark by many studies cited in this chapter and there isn't another benchmark with such ubiquity.

3. INFORMATION THEORETIC FUNDAMENTALS

This chapter provides an overview of information theory which underpins the branching entropy, unsupervised, text segmentation techniques. Information theory is the basis for the branching entropy techniques that will be used in the segmentation of isiXhosa text.

Claude Elwood Shannon established the field of information theory in his paper titled “A Mathematical Theory of Communication” (Shannon, 1948). The theory initially considered questions in telecommunications on the transmission of information across a noisy channel, particularly the shortest possible error-free data source encoding as well as the maximum error-free information transmission rate. Shannon introduced two fundamental measures of information, namely entropy and mutual information. The metrics introduced in that paper are now widespread and their use in more than just telecommunications is widely prevalent (Prokopenko et al., 2009; Itti & Baldi, 2009; Williams & Beer, 2010; Scutari, 2018).

3.1 Entropy

Entropy is the measure of the uncertainty of a random variable, in this case the output of a data source (Cover & Thomas, 1991:13). The convention in this document uses uppercase letters for random variables, lowercase letters for outcomes and Greek script letters for the dictionary of all possible outcomes, with respective examples: X , x and χ .

Definition 3-1. Let X be random variable with events outcomes $x \in \chi$. Let the probability of an event $p(X = x)$ be $p(x)$. The **entropy** of X , $H(X)$, under some base b , is calculated as

$$H(X) = \sum_{x \in \chi} p(x) \log_b \frac{1}{p(x)}. \quad (3-1)$$

The unit of entropy is the bit when the logarithm base b is two (2), and the nat if the base is ten (10). For brevity, this document uses logarithm base two by default, but the theory applies to decimal logarithm base 10 as well. $H(X)$ is bound by zero (0) at the minimus, when there is no uncertainty, and $\log_b |\chi|$, at maximum uncertainty. $|\chi|$ denotes the size of the dictionary χ . $H(X)$ is zero (0) when $p(x) = 1$ for one of the outcomes and $p(x) = 0$ for all the other events. $H(X)$ is largest when all the outcomes are equiprobable, i.e. $p(x) = \frac{1}{|\chi|}$. Entropy is also considered a measure of the information of a random variable.

Another way of looking at equation (3-1) is to consider it as the expectation of a function $f(x)$ under the probability distribution $p(x)$. In this case

$$f(x) = \log \frac{1}{p(x)}, \quad (3-2)$$

so that

$$H(X) = E_p[f(x)] = E_p \left[\log \frac{1}{p(x)} \right]. \quad (3-3)$$

The measure $f(x)$ above is referred to as the *surprisal* of the outcome $X = x$. It measures the degree of surprise at discovering the outcome x . It is also interpreted as the *information content* of the outcome x . Thus the surprisal and information content of an outcome is highest when its probability is low. Therefore, $H(X)$ is the average surprisal / information content associated with a random variable X .

The original explanation by Shannon was as follows. If X is the random variable of a source output in information coding and the coding scheme allocates the shortest code to the most probable outcome using a code of length $f(x)$, then $H(X)$ is the average length of the shortest possible error-free coding of the source output.

It is important to note that each event's contribution to the entropy, $p(x) \cdot \log \frac{1}{p(x)}$, is non-negative.

Entropy is easily extendable to two or more random variables with the concepts of joint entropy and conditional entropy.

3.1.1 Joint Entropy

Joint entropy is the uncertainty associated with more than one random variable.

Definition 3-2. The **joint entropy** of two random variables X and Y , with events $x \in \chi$ and $y \in \gamma$ respectively, and compound events $(x, y) \in \chi \times \gamma$ with probability distribution $p(x, y)$, is the expectation of the surprisal function $f(x, y)$ such that

$$f(x, y) = \log \frac{1}{p(x, y)}, \quad (3-4)$$

defined under the compound space and is calculated as

$$H(X, Y) = E_p[f(x, y)] \quad (3-5)$$

$$= \sum_{x \in \chi} \sum_{y \in \gamma} p(x, y) \log \frac{1}{p(x, y)}. \quad (3-6)$$

Again, the contribution of each event to the joint entropy, $p(x, y) \cdot \log \frac{1}{p(x, y)}$, is non-negative.

3.1.2 Conditional Entropy and Residual entropy

Having defined the joint surprisal, conditional measures of information are defined. First the conditional surprisal is defined, which is the surprisal of one variable when another variable is known.

Definition 3-3. *Conditional surprisal is defined as*

$$f(x|y) = \log \frac{1}{p(x|y)}. \quad (3-7)$$

From the conditional surprisal, define the conditional entropy.

Definition 3-4. *The conditional entropy, $H(X|Y)$, is defined as*

$$H(X|Y) = E_p [f(x|y)] = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{1}{p(x|y)}. \quad (3-8)$$

Conditional entropy, $H(X|Y)$, measures the residual uncertainty about X when Y is known. With this Equation (3-8) can be restated as

$$H(X|Y) = \sum_{y \in \mathcal{Y}} p(y) \sum_{x \in \mathcal{X}} p(x|y) \log \frac{1}{p(x|y)}, \quad (3-9)$$

$$= \sum_{y \in \mathcal{Y}} p(y) H(X|y), \quad (3-10)$$

where

$$H(X|y) = \sum_{x \in \mathcal{X}} p(x|y) \log \frac{1}{p(x|y)}. \quad (3-11)$$

The joint entropy can also be formulated from the conditional entropy as follows

$$H(X, Y) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} p(x, y) \log \frac{1}{p(x|y)p(y)}, \quad (3-12)$$

$$= \left[\sum_{y \in \mathcal{Y}} p(y) \log \frac{1}{p(y)} \right] + \left[\sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} p(x, y) \log \frac{1}{p(x|y)} \right], \quad (3-13)$$

$$= H(Y) + H(X|Y), \quad (3-14)$$

$$= H(X) + H(Y|X). \quad (3-15)$$

When X and Y have marginal entropies $H(X)$ and $H(Y)$, then the joint entropy of these variables is the marginal entropy of one variable, say X , plus the residual entropy of the other variable when the first variable is known, $H(Y|X)$ in our case. This is referred to as the *chain rule of entropy*. Therefore, joint entropy is the uncertainty of the whole system, while conditional entropy is the uncertainty about the system that is attributable only to a single variable. Thus $H(Y|X)$, is the amount of uncertainty in the joint system that is attributable to Y , and that cannot be accessed via X . Thus even for known X , the variable Y still retains information about the joint system ($X \times Y$) not known in X namely $H(Y|X)$. Of course, X behaves the same, it exclusively retains $H(X|Y)$ information about ($X \times Y$) to itself. Shared information between the variables is called mutual information (cf. section 3.2). This also means that adding a variable to a system only adds to the residual information, the information that the new variable does not share with the system.

In a joint system, the portion of the information that can only be accessed via each variable as the sum of the residual entropies on the individual variable is determined using the residual entropy:

$$R(X, Y) = H(X|Y) + H(Y|X). \quad (3-16)$$

Statistical independence between the variables is only possible when the joint entropy equals the residual entropy,

$$R(X, Y) = H(X, Y) \Leftrightarrow X \perp Y. \quad (3-17)$$

In Equation (3-17), the symbol $\perp$ denotes statistical independence.

3.1.3 Cross Entropy

Suppose there is a random variable X , with outcomes $x \in \mathcal{X}$, and $p(x)$ as the true distribution. Suppose one estimates X using another random distribution $q(x)$. How much information content does our estimate of the distribution provide? What is the average number of bits needed to encode X using q . In this case, the surprisal is

$$f_q(x) = \log \frac{1}{q(x)}, \quad (3-18)$$

and cross entropy can then be defined as below.

Definition 3-5. If a random variable X with outcomes $x \in \chi$, whose accepted probability mass distribution $p(x)$ is estimated using another probability mass distribution $q(x)$, the uncertainty about X under the estimate probability mass distribution, **the cross entropy**, is

$$H_{p,q}(X) = E_p[f_q(x)], \quad (3-19)$$

$$= \sum_{x \in \chi} p(x) \log \frac{1}{q(x)}. \quad (3-20)$$

This notation differs slightly from the conventional notation. The conventional notation used is $H(p, q)$ which emphasizes the fact that entropy is a function of a probability distribution. $H_{p,q}(X)$ is inspired by the notation of expectation, E_p , as entropy is an expectation over a metric under a probability distribution. This makes the notation for entropy $H_p(X)$, but the subscript is implied in most cases as the metric uses the same probability distribution.

3.1.4 Relative Entropy

The different types of entropies above have been entropies for distributions under different random variables, except for cross entropy. However, to measure the difference in entropy of the same random variable under different random distributions, requires an extension. This is considered by the *Kullback-Leibler divergence* (Kullback & Leibler, 1951) and is also known as the *relative entropy*.

Suppose there is a random variable X , with outcomes $x \in \chi$, and whose true distribution is $p(x)$. Suppose this random variable is estimated by another random distribution $q(x)$. The following definition determines the difference of the approximate encoding from the true encoding.

Definition 3-6. The **Kullback-Leibler (KL) divergence** between two probability distributions is

$$D(p \parallel q) = \sum_{x \in \chi} p(x) \log \frac{p(x)}{q(x)}. \quad (3-21)$$

The KL divergence can also be calculated from entropies as

$$D_{KL}(p \parallel q) = \left[\sum_{x \in \mathcal{X}} p(x) \log \frac{1}{q(x)} \right] - \left[\sum_{x \in \mathcal{X}} p(x) \log \frac{1}{p(x)} \right], \quad (3-22)$$

$$= H_{p,q}(X) - H_p(X). \quad (3-23)$$

Importantly, the contribution of each event, $p(x) \log \frac{p(x)}{q(x)}$, to the relative entropy is not always positive. It is only positive when $p(x) > q(x)$.

The most important characteristic of relative entropy is that it is always non-negative. The proof for this is due to *Jensen's inequality* (Jensen, 1906) and can be found in Cover and Thomas (1991:26) and Murphy (2012:58)

$$D(p \parallel q) \geq 0. \quad (3-24)$$

The Kullback-Leibler divergence is equal to zero only when the two distributions are equal,

$$D(p \parallel q) = 0 \Leftrightarrow p(x) = q(x) \quad \forall x \in \mathcal{X}. \quad (3-25)$$

What this inequality implies is that the cross entropy of any two distributions of the same random variable is always larger than or equal to the entropies of the individual distributions,

$$H_p(X) \leq H_{p,q}(X) \leq H_q(X). \quad (3-26)$$

Another property to note is that relative entropy is not a distance measure as it is not symmetrical and does not satisfy the triangle inequality,

$$|D(p \parallel q)| \neq |D(q \parallel p)|, \quad (3-27)$$

$$|D(p \parallel z)| + |D(z \parallel p)| \not\geq |D(q \parallel p)|. \quad (3-28)$$

3.1.5 Conditional Relative Entropy

The relative entropy of two conditional probabilities over the same random variables is referred to as the conditional relative entropy.

$$D(p(x|y) \parallel q(x|y)) = E_{p(x,y)} \left[\log \frac{p(X|Y)}{q(X|Y)} \right], \quad (3-29)$$

$$= \sum_{y \in \mathcal{Y}} p(y) \sum_{x \in \mathcal{X}} p(x|y) \log \frac{p(x|y)}{q(x|y)}. \quad (3-30)$$

Conditional relative entropy is the average relative entropy between the two conditional probability functions.

3.2 Mutual Information

The second information metric that Shannon introduced in his paper is mutual information. Mutual information tells us how much information the variables share, how much knowing one variable tells us about the other. For two variables mutual information measures how similar the joint distribution of random variables, $p(X, Y)$, is to their factored distribution, $p(X)p(Y)$. It is a measure of how independent the two variables are on average.

Definition 3-7. *The mutual information between two random variables is calculated as*

$$I(X; Y) \triangleq D_{KL}(p(X, Y) \parallel p(X) \cdot p(Y)), \quad (3-31)$$

$$= E_{p(x,y)} \left[\log \frac{p(x, y)}{p(x)p(y)} \right], \quad (3-32)$$

$$= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}. \quad (3-33)$$

The above metric is the average of the joint outcomes metric, called the point-wise mutual information (PMI) which measures how independent two events are, shown in Equation (3-34).

$$PMI(x, y) \triangleq \log \frac{p(x, y)}{p(x)p(y)}. \quad (3-34)$$

PMI can take any number below or above zero but the mutual information, $I(X; Y)$, is always non-negative.

Mutual information can be decomposed into entropies, as

$$I(X; Y) = H(X) + H(Y) - H(X, Y). \quad (3-35)$$

Proof:

$$I(X; Y) = \sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}, \quad (3-36)$$

$$= \left[\sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} p(x, y) \log \frac{1}{p(x)} \right] + \left[\sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(y)} \right], \quad (3-37)$$

$$= \left[\sum_{x \in \mathcal{X}} p(x) \log \frac{1}{p(x)} \right] - \left[\sum_{(x,y) \in \mathcal{X} \times \mathcal{Y}} p(x,y) \log \frac{1}{p(x|y)} \right], \quad (3-38)$$

$$= H(X) - H(X|Y), \quad (3-39)$$

$$= H(X) + H(Y) - H(X,Y) \blacksquare \quad (3-40)$$

The next section deals with conditional mutual information.

3.2.1 Conditional mutual information

Conditional mutual information defines the mutual information between two random variables conditioned on a third variable $I(X; Y|Z)$. It is the reduction in the uncertainty of one variable due to the knowledge of the second given knowledge of the third variable. Given the previous definitions of random variables X and Y , if Z is a random variable with event $z \in \mathcal{Z}$, and the joint probability distribution of the three variables being $p(x, y, z)$ then the conditional mutual information $I(X; Y|Z)$ is defined as follows

Definition 3-8. *The conditional mutual information of random variables X and Y when Z is known is*

$$I(X; Y|Z) \triangleq E_{p(x,y,z)} \left[\log \frac{p(X, Y|Z)}{p(X|Z)p(Y|Z)} \right], \quad (3-41)$$

$$= H(X|Z) - H(X|Y, Z). \quad (3-42)$$

Proof:

$$I(X; Y|Z) = E_{p(x,y,z)} \left[\log \frac{p(X|Y, Z)p(Y|Z)}{p(X|Z)p(Y|Z)} \right], \quad (3-43)$$

$$= E_{p(x,y,z)} \left[\log \frac{p(X, Y, Z)p(X)}{p(X|Z)p(Y|Z)p(Z)p(X)} \right], \quad (3-44)$$

$$= E_{p(x,y,z)} \left[\log \frac{p(X, Y, Z)p(X)}{p(X|Z)p(Y, Z)p(X)} \right], \quad (3-45)$$

$$= E_{p(x,y,z)} \left[\log \frac{p(X, Y, Z)}{p(X)p(Y, Z)} - \log \frac{p(X|Z)}{p(X)} \right], \quad (3-46)$$

$$= I(X; Y, Z) - I(X; Z) \blacksquare \quad (3-47)$$

The proof of Equation (3-47) is in Cover and Thomas (1991:22).

The next section deals with the chain rules for the above metrics, allowing for the extension of these metrics to more than two random variables.

3.3 Chain rule of entropy, relative entropy and mutual information

Consider $\mathbf{X} = X_1 \times X_2 \times \dots \times X_n$, a product ensemble of n random variables $X_1, X_2, \dots, X_n$ with events $x_1 \in \chi_1, x_2 \in \chi_2, \dots, x_n \in \chi_n$ and a joint distribution $p(x_1, x_2, \dots, x_n)$. In a stochastic process $x_i \in \chi \mid i \in \{1, 2, \dots, n\}$.

Definition 3-9. Stochastic processes are time or sequence indexed random variables (Cover & Thomas, 1991:60)

Stochastic processes are generally numerically indexed with increasing integers, i.e. $X_1, X_2, \dots, X_n$. This stream is characterised by a joint probability distribution $p(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n)$, where $x_i \in \chi \mid i \in \{1, 2, \dots, N\}$ and χ is the vocabulary of possible outcomes. For brevity, the notation of the probability distribution is shortened to $p(x_1, x_2, \dots, x_n)$.

Using the chain rule of probability (Murphy, 2012:29), the joint distribution can also be stated as

$$p(x_1, x_2, \dots, x_n) = \prod_{i=1}^n p(x_i | x_1, \dots, x_{i-1}). \quad (3-48)$$

The sections starts with the chain rule of entropy, then cover that of relative entropy and finish with that of mutual information.

3.3.1 Chain Rule of Entropy

The entropy, $H(\mathbf{X})$ of n random variables $\mathbf{X} = X_1, X_2, \dots, X_n$ can be presented as sum of conditional entropies.

Theorem 3-1. Chain rule of entropy. The entropy of $H(\mathbf{X} = X_1 \times X_2 \times \dots \times X_n)$ can be expressed as

$$H(\mathbf{X}) = \sum_{i=1}^n H(X_i | X_1, \dots, X_{i-1}). \quad (3-49)$$

The proof below is from Cover and Thomas (1991:21) and uses two techniques, i.e. the repeated application of the two-variable expansion rule [Equations (3-14) and (3-15)], and the use of Equation (3-48).

Proof.

$$H(\mathbf{X}) = - \sum_{x_1 \in \chi_1, x_2 \in \chi_2, \dots, x_n \in \chi_n} p(x_1, x_2, \dots, x_n) \log p(x_1, x_2, \dots, x_n). \quad (3-50)$$

Using the chain rule of probability

$$= - \sum_{x_1 \in \chi_1, x_2 \in \chi_2, \dots, x_n \in \chi_n} p(x_1, x_2, \dots, x_n) \log \prod_{i=1}^n p(x_i | x_1, \dots, x_{i-1}), \quad (3-51)$$

$$= - \sum_{x_1 \in \chi_1, x_2 \in \chi_2, \dots, x_n \in \chi_n} \sum_{i=1}^n p(x_1, x_2, \dots, x_n) \log p(x_i | x_1, \dots, x_{i-1}). \quad (3-52)$$

Swapping the sums and marginalising out $x_i, \dots, x_n$, gives us

$$H(\mathbf{X}) = \sum_{i=1}^n \left[- \sum_{x_1 \in \chi_1, x_2 \in \chi_2, \dots, x_i \in \chi_i} p(x_1, x_2, \dots, x_i) \log p(x_i | x_1, \dots, x_{i-1}) \right], \quad (3-53)$$

$$= \sum_{i=1}^n H(X_i | X_1, \dots, X_{i-1}) \blacksquare \quad (3-54)$$

For stochastic processes, this is referred to as the block entropy $H(n)$ (Prokopenko et al., 2009), defined as

$$H(n) = - \sum_{x^n \in \chi^n} p(x^n) \log p(x^n). \quad (3-55)$$

Next is the chain rule for relative entropy.

3.3.2 Chain Rule of Relative Entropy

For the purposes of this section, the notation of relative entropy is modified from $D(p \parallel q)$ to $D_{p,q}(X)$, such that

$$D_{p,q}(X) = D(p(X) \parallel q(X)). \quad (3-56)$$

This allows a formulation of the relative entropy for multiple random variables.

Theorem 3-2. Chain rule of relative entropy. Relative entropy for multiple random variables can be restated as

$$D_{p,q}(\mathbf{X}) = \sum_{i=1}^n D_{p,q}(X_i | X_1, \dots, X_{i-1}). \quad (3-57)$$

Proof: Using the proof of Cover and Thomas (1991:23) for two random variables, having extended it to multiple variables one gets

$$D_{p,q}(\mathbf{X}) = \sum_{x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2, \dots, x_n \in \mathcal{X}_n} p(x_1, x_2, \dots, x_n) \log \frac{p(x_1, x_2, \dots, x_n)}{q(x_1, x_2, \dots, x_n)}, \quad (3-58)$$

$$= \sum_{x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2, \dots, x_n \in \mathcal{X}_n} p(x_1, x_2, \dots, x_n) \log \prod_{i=1}^n \frac{p(x_i | x_1, \dots, x_{i-1})}{q(x_i | x_1, \dots, x_{i-1})}, \quad (3-59)$$

$$= \sum_{x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2, \dots, x_n \in \mathcal{X}_n} \sum_{i=1}^n p(x_1, x_2, \dots, x_n) \log \frac{p(x_i | x_1, \dots, x_{i-1})}{q(x_i | x_1, \dots, x_{i-1})}. \quad (3-60)$$

Having swapped the sums and marginalising $x_i, \dots, x_n$ gives

$$D_{p,q}(\mathbf{X}) = \sum_{i=1}^n \sum_{x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2, \dots, x_i \in \mathcal{X}_i} p(x_1, x_2, \dots, x_i) \log \frac{p(x_i | x_1, \dots, x_{i-1})}{q(x_i | x_1, \dots, x_{i-1})}, \quad (3-61)$$

$$= \sum_{i=1}^n D_{p,q}(X_i | X_1, \dots, X_{i-1}) \blacksquare \quad (3-62)$$

The chain rule for mutual information follows.

3.3.3 Chain Rule of Mutual Information

Fano (1961:58) presented the multivariate mutual information as,

$$I(\mathbf{X}) = H(\mathbf{X}) - \sum_{i=1}^n H(X_i | \mathbf{X}_{\setminus i}) \quad (3-63)$$

$$I(\mathbf{X}) = I(\mathbf{X}_{\setminus i}) - I(\mathbf{X}_{\setminus i} | X_i), \quad (3-64)$$

$$= \sum_{i=1}^n (-1)^{i-1} \sum_{\mathbf{X}^p \subseteq \mathcal{P}_i(\mathbf{X})} I(\mathbf{X}^p). \quad (3-65)$$

Here $\mathbf{X}_{\setminus i}$ is the product ensemble of $\mathbf{X} = X_1 \times X_2 \times \dots \times X_n$ with X_i excluded, and $i, k \in 1, 2, \dots, n$. $\mathcal{P}_i(\mathbf{X})$ is a set of all the ordered combinations of random variables from $\mathbf{X}$ of size i , i.e. the partial ordered partitions of $\mathbf{X}$ of size i . This can also be presented in terms of entropies of subsets (Ince, 2017 Equation (13))

$$I(\mathbf{X}) = - \sum_{i=1}^n (-1)^{n-i} \sum_{\mathbf{X}^p \subseteq \mathcal{P}_i(\mathbf{X})} H(\mathbf{X}^p). \quad (3-66)$$

Another way to look at multivariate mutual information is to decompose it into sets of local mutual information.

Definition 3-10 (Local Mutual Information). Let $x^n = x_1, x_2, \dots, x_n$ be the outcome of a multivariate random variable $X^n = X_1, X_2, \dots, X_n$. the local mutual information $i(x^n)$ between the n outcomes of x^n is defined as

$$I(X^n) = \sum_{x^n \in X^n} p(x^n) \cdot i(x^n) \quad , i(x^n) = i(x_1; x_2; \dots; x_n)$$

Induction is used to reformulate multivariate Mutual Information for variables $X^n = \mathbf{X} = X_1 \times X_2 \times \dots \times X_n$. For $n = 1$

$$I(X^1) = \sum_{x_1} p(x) \cdot \log \frac{1}{p(x_1)} \Rightarrow i(x) = \log \frac{p(x_1 | \text{None})}{p(x_1)} = \log \frac{1}{p(x_1)}. \quad (3-67)$$

It follows from independence that $p(x | \text{None}) = 1$ since the probability that the outcome x exists is 1. For $n = 1$

$$I(X^2) = \sum_{x_1, x_2} p(x_1, x_2) \cdot \log \frac{p(x_2 | x_1)}{p(x_2)} \Rightarrow i(x^2) = i(x_1; x_2) = \log \frac{p(x_2 | x_1)}{p(x_2)}. \quad (3-68)$$

For $n = 3$

$$I(X^3) = \sum_{x_1, x_2, x_3} p(x_1, x_2, x_3) \cdot \left[\log \frac{p(x_3 | x_1, x_2)}{p(x_3)} - \log \frac{p(x_3 | x_2)}{p(x_3)} - \log \frac{p(x_3 | x_1)}{p(x_3)} \right] \quad (3-69)$$

$$= \sum_{x_1, x_2, x_3} p(x_1, x_2, x_3) \cdot [i(x_1, x_2; x_3) - i(x_2; x_3) - i(x_1; x_3)] \quad (3-70)$$

$$\Rightarrow i(x^3) = i(x_1; x_2; x_3) = i(x_1, x_2; x_3) - i(x_2; x_3) - i(x_1; x_3) \quad (3-71)$$

where $i(x = x_a, x_b, \dots, x_k; x_n)$ and x is some multivariate outcome of length, is defined as

$$i(x; x_n) = \log \frac{p(x_n | \mathbf{x})}{p(x_n)}. \quad (3-72)$$

For $n = 4$

$$I(X^4) = \sum_{x_1, x_2, x_3, x_4} p(x_1, x_2, x_3, x_4) \cdot \left[\begin{array}{c} -i(x_1; x_4) - i(x_2; x_4) - i(x_3; x_4) \\ +i(x_1, x_2; x_4) + i(x_1, x_3; x_4) + i(x_2, x_3; x_4) \\ -i(x_1, x_2, x_3; x_4) \end{array} \right] \quad (3-73)$$

$$\Rightarrow i(x^4) = i(x_1; x_2; x_3; x_4) = \begin{array}{c} -i(x_1; x_4) - i(x_2; x_4) - i(x_3; x_4) \\ +i(x_1, x_2; x_4) + i(x_1, x_3; x_4) + i(x_2, x_3; x_4) \\ -i(x_1, x_2, x_3; x_4) \end{array}. \quad (3-74)$$

Using induction, $i(\mathbf{x}^n = x_1; x_2; \dots; x_n)$ is then formulated to

$$i(\mathbf{x}^n) = - \sum_{j=1}^{n-1} (-1)^j \sum_{\mathbf{x}^p \subseteq \mathcal{P}_j(\mathbf{x}^{n-1})} i(\mathbf{x}^p; x_n) \blacksquare \quad (3-75)$$

3.4 Predictive metrics for stationary stochastic processes

Stochastic processes as defined in **Definition 3-9** are random variables that are a sequence, a stream (Cover & Thomas, 1991:60). They may be time indexes but are generally numerically indexed with increasing integers, e.g. $X_1, X_2, \dots, X_n$. This stream is characterised by a joint probability distribution $p(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n)$, where $x_i \in \chi \mid i \in \{1, 2, \dots, n\}$ and χ is the vocabulary of possible outcomes. For brevity, shorten the notation of the probability distribution to $p(x_1, x_2, \dots, x_n)$. To establish predictive metrics, start by defining a number of important random processes.

Definition 3-11. A **stationary process** is a stochastic process where the probability distribution of any fixed length sequence is shift invariant, e.g.

$$p(x_1, x_2, \dots, x_n) = p(x_{1+l}, x_{2+l}, \dots, x_{n+l}) \quad n, l \in \mathbb{Z} \quad (3-76)$$

Definition 3-12. A **Markov process** is a stationary process whose probability distribution can be reduced to a function of conditional distributions of a fixed length, l , such that;

$$p(x_1, x_2, \dots, x_n) = p(x_1) p(x_2|x_1) p(x_3|x_1, x_2) \dots p(x_n|x_1, \dots, x_{n-1}), \quad (3-77)$$

$$= \prod_{i=1}^n p(x_i|x_{i-l+1}, \dots, x_{i-1}). \quad (3-78)$$

Recent times have seen a lot of work into information theoretic metrics for predicting complexity, self-organisation and emergence of systems (Fernández et al., 2014; Rosas et al., 2018; Prokopenko et al., 2009). This section focuses on those that have promise in segmentation.

In prediction, the objective is to predict the future depending on a seen past. Denoting the future and the past as $\vec{X}$, the past as $\tilde{X}$, and the future as $\hat{X}$, then prediction involves predicting the probability of some future given the past, $p(\vec{X}|\tilde{X})$.

The following metrics are used in quantifying predictability of a stochastic process.

3.4.1 Entropy Rate

Definition 3-13. For a product ensemble $X_1, X_2, \dots, X_n$ with a probability distribution $p(X_1, X_2, \dots, X_n)$ the **entropy rate** (Cover & Thomas, 1991:63) is defined as

$$h_\mu = \lim_{n \rightarrow \infty} \frac{H(X_1, X_2, \dots, X_n)}{n} \quad (3-79)$$

For a stationary process, the entropy rate exists and can be expressed according to conditional entropies as below (Cover & Thomas, 1991:64).

$$h_\mu = \lim_{n \rightarrow \infty} H(X_n|X_1, \dots, X_{n-1}) \quad (3-80)$$

The entropy rate can be approximated using the block entropy rate

$$h_\mu(n) = \frac{H(X_1, X_2, \dots, X_n)}{n} \quad (3-81)$$

$$= H(X_n|X_1, \dots, X_{n-1}) \text{ for a stochastic process} \quad (3-82)$$

Boffeta et al. (2002) redefined the entropy rate. Their approach is based on the block entropy for stochastic processes.

In a stochastic process, Prokopenko (2009) defines the block entropy, $H(n)$, of a data stream of length n , $x^n = x_1, x_2, \dots, x_n$ where $x_i \in \chi \mid i \in \{1, 2, \dots, N\}$ as

$$H(n) = - \sum_{x^n \in \chi^n} p(x^n) \log p(x^n). \quad (3-83)$$

In the above, χ^n contains all possible sequence of length n . This is then used in formulating the entropy gain.

Definition 3-14. The entropy gain $h_\mu(n)$ is then defined as

$$h_\mu(n) = H(n) - H(n-1). \quad (3-84)$$

The entropy gain is the average uncertainty about the n -th symbol given the $n-1$ previous symbols. This leads to a re-definition of entropy rate h_μ , which is

$$h_\mu = \lim_{n \rightarrow \infty} h_\mu(n) = \lim_{n \rightarrow \infty} \frac{H(n)}{n} = \lim_{n \rightarrow \infty} H(X_n | X_1, \dots, X_{n-1}) \quad (3-85)$$

The entropy rate is also referred to as the entropy density, the density of information per unit time or symbol depending on the index. The entropy gain, $h_\mu(n)$, is always positive because $H(n) \geq H(n-1)$.

Of importance is that $h_\mu = 0$ for periodic and deterministic processes, even those with infinite-memory but no randomness. $h_\mu > 0$ for unpredictable processes, e.g. independent and identically distributed process, as well as Markov processes. $h_\mu(n)$ estimates the number of parameters k in the system to model sequences of length n such that

$$h_\mu(n) = \frac{k}{2} \log_2 n. \quad (3-86)$$

This measure is used in the calculation of excess entropy, explained below, but starting with an entropy measure that is under-utilised, namely redundancy.

3.4.2 Redundancy

As indicated above, the entropy of a random variable X^n with outcomes $x_i \in \chi$ has a maximum observable value of $n \log |\chi|$. Redundancy is the difference between the maximum possible entropy rate and the data's entropy rate.

Definition 3-15. *Redundancy is calculated as*

$$R = \log|\chi| - h_\mu \quad (3-87)$$

This is another measure of the predictability of the random process.

3.4.3 Excess Entropy

Excess entropy, E , is the measure of the intrinsic redundancy of a source (Grassberger, 1986).

Definition 3-16. *Excess entropy of a process is defined as*

$$E = \sum_{n=1}^{\infty} [h_\mu(n) - h_\mu]. \quad (3-88)$$

Excess entropy measures the structure in a source. It “measures the amount of apparent randomness at small values n that is explained by considering correlation over larger blocks” (Crutchfield & Feldman, 2003). Crutchfield and Feldman provide a finite-sum estimate of excess entropy for length n as

$$E(n) = H(n) - nh_\mu(n). \quad (3-89)$$

Excess entropy is also expressed as the mutual information between future events and past events of a process, precisely the amount of information stored in one half that is useful to predict the other half (Grassberger, 1986; Crutchfield & Feldman, 2003),

$$E = I_{pred}(\tilde{X}; \vec{X}), \quad (3-90)$$

$$= H(\vec{X}) - H(\vec{X}|\tilde{X}), \quad (3-91)$$

$$= H(\tilde{X}) + H(\vec{X}) - H(\vec{\tilde{X}}), \quad (3-92)$$

where $\vec{\tilde{X}}$ depicts the sequence data covering the past and the future. In this expression, it is referred to as predictive information (Prokopenko et al., 2009). This is the measure of information that the past provides about the future. Excess Entropy can also be considered the cost of amnesia.

3.4.4 Transient Information

Transient Information quantifies how block entropy, $H(n)$, converges to $E + nh_\mu$ (Feldman & Crutchfield, 2004).

Definition 3-17. Transient information is defined as

$$T = n \sum_{n=1}^{\infty} E + n h_{\mu} - H(n). \quad (3-93)$$

3.4.5 Transfer Entropy

Transfer entropy (Schreiber, 2000) measures the effect that one process has on another. Consider two processes X and Y . This is looking for information that the history $\tilde{Y}$ brings to the current value of X , X_t .

Definition 3-18. Transfer entropy is defined as

$$T_{Y \rightarrow X} = I(X_t; \tilde{Y} | \tilde{X}). \quad (3-94)$$

If $\tilde{Y}$ is the previous j states of Y , such that $\tilde{Y} = Y_{t-j}, \dots, Y_{t-2}, Y_{t-1}$ and $\tilde{X}$ is the previous k states of X such that $\tilde{X} = X_{t-k}, \dots, X_{t-2}, X_{t-1}$ then the transfer entropy, $T_{Y \rightarrow X}$, is the amount information that the previous j states of Y provide about the current state of X , X_t , when both are conditioned on the previous k states of X . Conditioning on $\tilde{X}$ removes its effect on X_t .

From Equations (3-41) and (3-47), where $x_t, \tilde{y}$ and $\tilde{x}$ are outcomes of random variable ensembles $X_t, \tilde{Y}$ and $\tilde{X}$ respectively, restate transfer entropy as,

$$T_{Y \rightarrow X} = E_{p(x_t, \tilde{y}, \tilde{x})} \left[\log \frac{p(x_t | \tilde{y}, \tilde{x})}{p(x_t | \tilde{x})} \right], \quad (3-95)$$

$$= H(X_t | \tilde{X}) - H(X_t | \tilde{Y}, \tilde{X}). \quad (3-96)$$

From the above two things are apparent. Firstly, $T_{Y \rightarrow X}$ is zero only when $\tilde{Y}$ does not have an influence on X_t ,

$$T_{Y \rightarrow X} = 0 \Leftrightarrow p(x_t | \tilde{y}, \tilde{x}) = p(x_t | \tilde{x}) \quad \forall x_t, \tilde{y}, \tilde{x}. \quad (3-97)$$

Secondly, the limit of $T_{Y \rightarrow X}$ as $j, k \rightarrow \infty$ generalises to the entropy rate between two processes.

3.5 Predictive Information Metrics

Bialek et al. (2001a; 2001b) argued that whilst the mutual information of the past and future contains predictive information, it also contains information that is useless to prediction. They

provided a decomposition that extracts the predictive information from the excess information $E = I_{pred}(\tilde{X}; \vec{X})$.

The premise is that the entropy of a sequence, e.g. a time series, is extensive and proportional to its duration, meaning the entropy rate, h_μ , of the past exists and is quantifiable.

From Equation (3-89) the block entropy can be represented as

$$H(n) = E(n) + nh_\mu(n) \quad \text{and,} \quad (3-98)$$

$$\lim_{n \rightarrow \infty} H(n) = E + nh_\mu(n). \quad (3-99)$$

From $E(n)$, the predictability of the source can be determined. Excess entropy $E(n)$ is subextensive, meaning that it grows with n less rapidly than a linear function. That said, it increases with the amount of complexity in the process. Bialek et al. (2001b), provided three scenarios that E can take and what they say about the predictability of a source. The Excess entropy increases when

- E is finite for periodic and random process, being zero for IID processes.
- The finite estimate $E(n)$ diverges logarithmically for complex processes. The coefficient of this divergence estimates the number of parameters or rules in the process.

One of the key criteria of predictability of a system is whether the entropy rate falls below the measurement channel capacity or not. If that is so, the process can be predicted from its data, otherwise, it is not predictable (Shannon, 1948; Crutchfield & Feldman, 2003).

It is important to note that $h_\mu(n)$ is considered the estimate of the process' unpredictability and decreases monotonically to the limit h_μ , the process's "long-time measure of unpredictability of sequences" produced by it. Crutchfield and Feldman derive a predictability gain metric;

Definition 3-19. Predictability gain is defined from estimates of entropy rate as

$$\Delta^2 H(n) = \Delta h_\mu(n) = h_\mu(n) - h_\mu(n-1) \quad \text{with} \quad (3-100)$$

$$\Delta^2 H(1) = H(1) - \log|\chi| \quad (3-101)$$

A large value of $|\Delta^2 H(n)|$ indicates that the n 'th ensemble is informative. $\Delta^2 H(n)$ is always non-positive. This makes $H(n)$ a concave function of n . Crutchfield and Feldman also presented, the total predictability, G of a process;

Definition 3-20. Total predictability of a process with a defined $H(n)$ is

$$G = \sum_{n=1}^{\infty} \Delta^2 H(n). \quad (3-102)$$

Total predictability quantifies the amount of information that a predictor must acquire before it can predict the process.

3.6 Conclusions

In this chapter, Information theoretic measures were introduced. These measures provide the foundation to branching entropy

The chapter detailed the different types of entropy metrics as well as the mutual information metric. Chain rules for the entropy, relative entropy and mutual information metrics were detailed for stochastic processes.

Lastly, the chapter presented the state-of-the-art metrics, namely entropy rate and predictive gain, in measuring the predictive information contained in a stationary process.

4. SEGMENTATION WITH BRANCHING ENTROPY

This chapter provides some theoretic underpinning of branching entropy in unsupervised text segmentation. Different variants of the branching entropy technique are presented.

A new variant of the branching entropy techniques is introduced, Bracing Joint Entropy, which removes the need for modelling entropies in two directions, left to right and right to left, but instead provides a single measure.

The chapter also links the branching entropy techniques with predictive information metrics, and proves that these techniques capture certain types of information.

Lastly, the chapter investigates a combined way of training the directional branching entropies in a unified way, including models based on modified Kneser Ney smoothed probabilistic models.

4.1 Branching Entropy

Branching Entropy is used in two ways in text segmentation. The first method (Ando & Lee, 2000) involves identification of morpheme boundaries in a word. The second is a recent form that uses branching entropy in identifying possible morphemes (Magistry & Sagot, 2013).

Branching entropy techniques are based on the principle that with progression through the letters in a word, the predictability of the next letter increases, so a decrease in uncertainty. The consequence of detecting an increase in uncertainty, measured in terms of entropy, is that a morpheme boundary has occurred. Letter progression sequences are analysed using substring methods.

Substring Attachment

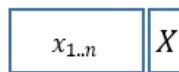


Figure 4-1: Right Branching from a substring

Definition 4-1. From Figure 4-1, consider a substring $x_{1..n}$, the uncertainty regarding the following character X , **Branching Entropy**, is defined as;

$$h(x_{1..n}) = H(X|x_{1..n}) \quad (4-1)$$

$$= - \sum_{x \in \mathcal{X}} P(x|x_{1..n}) \log P(x|x_{1..n})$$

The substring could have attachments on either side. A left side attachment is measured with Left Branching Entropy, $h_{\leftarrow}(x_{1..n})$, and conversely right side attachment as Right Branching Entropy, $h_{\rightarrow}(x_{1..n})$.

4.1.1 Morpheme Boundary Identification using Branching Entropy

Huang & Powers (2003), Jin and Tanaka-Ishii (2006) and Zikhov et al. (2010) used branching entropy as a way to identify a morpheme boundary in word segmentation. These works use the branching entropy without modification.

To extend their work, the work here repurposes a metric introduced by Magistry and Sagot (2012) and Magistry (2012; 2013), variation of branching entropy, for the identification of morpheme boundaries. This metric was initially used by Jin and Tanaka-Ishii (2006) as one of their criteria, B_{increase} , in boundary location identification. This was then used by Magistry and Sagot (2012) and Magistry (2012; 2013) to normalise the branching entropy.

For a word of length L , there are $L - 1$ possible boundary locations. For each of these possible boundary locations, a boundary branching entropy measure is calculated from a language corpus, according to **Definition 4-2** below.

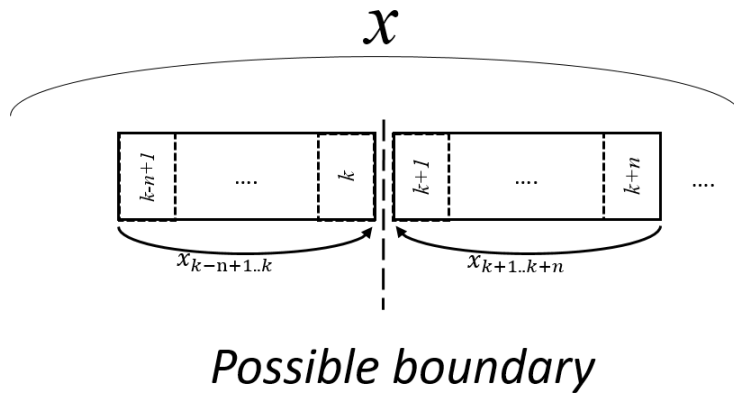


Figure 4-2: Location of a possible boundary in a substring

Definition 4-2. From (Magistry, 2012; Zhikov et al., 2013) and Figure 4-2, consider a word $x = x_{1..N}$ of length N . Determine the **Boundary Branching Entropy** across the x_k and x_{k+1} letters, represented as $x_{k;k+1}$, from the right branching entropy of the substring of length n to the

left of the boundary location, $h_{\rightarrow}(x_{k-n+1..k})$, and the left branching entropy of the substring of length n to the right of the boundary location, $h_{\leftarrow}(x_{k+1..k+n})$, as

$$H(x_{k;k+1}|x_{k-n+1..k}:x_{k+1..k+n}) = h_{\rightarrow}(x_{k-n+1..k}) \text{ op } h_{\leftarrow}(x_{k+1..k+n}) \quad (4-2)$$

$$\begin{aligned} &= - \sum_{x \in \chi} P(x|x_{k-n+1..k}) \log P(x|x_{k-n+1..k}) \text{ op} \\ &\quad - \sum_{x \in \chi} P(x|x_{k+1..k+n}) \log P(x|x_{k+1..k+n}) \end{aligned} \quad (4-3)$$

In the definition above, $x_{k;k+1}$ is the word location between letters x_k and x_{k+1} , $x_{i..j}$ is the word substring from the i -th letter to the j -th letter, χ is the vocabulary of letters in the language, $h_{\rightarrow}$ and $h_{\leftarrow}$ are the right branching and left branching entropies defined above and the operator op could be any binary operator. The *min* and the *sum* operations are generally used.

To infer boundaries, the peaks in the boundary branching entropy profile of the word are considered morpheme boundaries. Then for location $x_{k;k+1}$ to be a boundary, the two substrings $x_{k-n+1..k}$ and $x_{k+1..k+n}$ must combine into an increase in branching entropy at this location.

Working from Harris's hypothesis (1955) on morpheme boundaries, Magistry and Sagot (2012) and Magistry (2012; 2013) introduced the variation of branching entropy. The hypothesis is that, in general, the access variety decreases with increasing context size. If one translates this to branching entropy, generally $h(x_{1..n}) \geq h(x_{1..n+1})$. However, Magistry and Sagot do not use this metric in morpheme boundary identification but apply it in morpheme identification only.

Definition 4-3. *The Variation of Right Branching Entropy for a substring $x_{k-n+1..k}$ is calculated as*

$$\delta h_{\rightarrow}(x_{k-n+1..k}) = h_{\rightarrow}(x_{k-n+1..k}) - h_{\rightarrow}(x_{k-n+2..k}) \quad (4-4)$$

The variation of the left-branching entropy is defined analogously.

In this work, VBE is also used to identify morpheme boundaries. The VBE profile of a word is calculated and the peaks are identified as morpheme boundaries. Sect. 7.2 shows that this approach gives superior F1 scores.

4.1.2 Morpheme Identification using Branching Entropy

Instead of focusing on a location in determining the boundary location in a word, another investigated using a measure of “wordhood” or “morphemehood” of a substring (Zhang et al.,

2017; Magistry, 2013). The latter measures determine the likelihood of a substring being a morpheme.

In Figure 4-3, let χ be the dictionary set of letters in a language L . Let A, C and B be random variables with a, c and b as their outcome respectively. Let $a, c \in \chi$ and $b \in \chi^+$ such that $b = b_1, b_2, b_3, \dots, b_n$ and $b_i \in \chi$. χ^+ denotes a string of letters from χ . Let $\langle a, b, c \rangle$ be an $n + 2$ long observation of a substring in L .

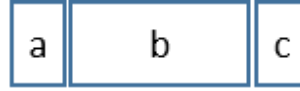


Figure 4-3: A graphical representation of left and right attachment to a substring

From access variety (Feng, Chen, Deng, et al., 2004) one can consider the wordhood of a substring to be a function of the variability of letters attaching to the substring on both sides.

Let the wordhood metric be $a(w_i)$. For a word $w = w_{1..n}$, inference of the best segmentation, s^* , is then

$$s^*(w) = \arg \max_{s \in Segs(w)} \sum_{w_i \in s} a(w_i) \cdot len(w_i) \quad (4-5)$$

Above $Segs(x)$ are possible segmentations of w , w_i is a possible morpheme from segmentation s , and $len(w_i)$ is the length of the segment.

To speed up inference a Viterbi-like optimisation using memorisation is used where, if the score of the best segmentation $s^*(w_{1..i})$ of a prefix $w_{1..i}$ is known, the best segmentation is

$$s^*(w) = \arg \max_{\substack{i=1..n \\ s^*(w_{1..i}), w_{i+1..n}}} s^*(w_{1..i}) + a(w_{i+1..n}) \cdot (n - i) \quad (4-6)$$

The first measure of this variability that this section considers is the newly Bracing Joint Entropy, being the joint entropy of the attaching letters conditioned on the substring.

4.1.2.1 Wordhood from Bracing Joint Entropy

The Bracing Joint Entropy considers the joint entropy of the attachment letters on both sides of the substring, as shown in Figure 4-3.

Definition 4-4. Considering Figure 4-3, the **Bracing Joint Entropy**, $BJE(b)$, of a substring b is calculated as

$$BJE(b) = H(A, C|B = b) \quad (4-7)$$

$$= - \sum_{a \in A} \sum_{c \in C} p(a, c|b) \log_2 p(a, c|b) \quad (4-8)$$

BJE measures how variable the combined side attachments are, given the substring they are attaching to. This measures strict variability around a substring.

Bracing Joint Entropy removes the complexity of keeping two entropy measurements per string as described in **Definition 4-2** for Branching Entropy, resulting in half the model size. This reduces memory and processing requirements for the algorithm.

4.1.2.2 Wordhood from individual left- and right- branching entropies

Whilst Bracing Joint Entropy measures the combined variability of the attachment letters, individual entropies from each side of the possible morpheme might give a better measure. This is the measure used in Magistry (2013).

In this setting, Bracing Joint Entropy is a stricter metric than the sum of the individual entropies because a joint entropy is no greater than the sum of the individual entropies (Williams, 2011),

$$H(A, B) \leq H(A) + H(B), \quad (4-9)$$

The BJE matches the separate entropy sum only when A is independent of B , but not as strict as the minimum of the two individual branching entropies. Thus the wordhood from individual branching entropies is calculated as

$$wBE(b) = \overleftarrow{H(A|B = b)} \text{ op } \overrightarrow{H(C|B = b)} \quad (4-10)$$

$$= \sum_{a \in A} p(a|b) \log_2 \frac{1}{p(a|b)} \text{ op } \sum_{c \in C} p(c|b) \log_2 \frac{1}{p(c|b)} \quad (4-11)$$

Here $\overleftarrow{H(A|B = b)}$ is the left branching entropy (LBE) and $\overrightarrow{H(C|B = b)}$ is the right branching entropy (RBE), the operator **op** is either the *sum* or the *minimum* of the two directional branching entropies.

4.1.2.3 Normalisation across substring sizes

To enable comparison between substrings of different lengths, normalisation is required. In this work, variation normalisation, statistical normalisation and a combination of both is used.

Variation normalisation uses the intuitive notion that metrics generally decrease with n-gram size (Harris, 1942; Magistry, 2013). Let the metric be denoted with ϖ . ϖ could be the bracing joint entropy or the branching entropy. For a case of combined side attachments metric, like the bracing joint entropy, the variation of the metric, $\delta\varpi$, for a substring $b = b_{1..n}$ of length n is calculated as the average of the metric for substrings one character shorter.

$$\delta\varpi(b_{1..n}) = \varpi(b_{1..n}) - \frac{\varpi(b_{1..n-1}) + \varpi(b_{2..n})}{2} \quad (4-12)$$

For a metric that uses an individual metric, like the branching entropy, for each attachment side of a substring, the following variation metrics are used, according to their direction.

$$\overleftarrow{\delta\varpi(b_{1..n})} = \overleftarrow{\varpi(b_{1..n})} - \overleftarrow{\varpi(b_{1..n-1})} \quad (4-13)$$

$$\overrightarrow{\delta\varpi(b_{1..n})} = \overrightarrow{\varpi(b_{1..n})} - \overrightarrow{\varpi(b_{2..n})} \quad (4-14)$$

The original metric, or the variation of the metric, can also be **statistically normalised** using either the mean of the metric for a length or the z-score which also uses the standard deviation (Magistry, 2013).

$$N_\mu\varpi(b_{1..n}) = \varpi(b_{1..n}) - \mu_n \quad (4-15)$$

$$N_z\varpi(b_{1..n}) = \frac{\varpi(b_{1..n}) - \mu_n}{\sigma_n} \quad (4-16)$$

In the above, μ_n is the mean of the metric for substrings of length n and σ_n is the standard deviation for the same length, where

$$\mu_n = \sum_b p(b_{1..n})\varpi(b_{1..n}) \quad (4-17)$$

$$\sigma_n = \sqrt{\sum_b p(b_{1..n})[\varpi(b_{1..n}) - \mu_n]^2} \quad (4-18)$$

$$\sigma_n = \sqrt{\sum_b p(b_{1..n}) \varpi(b_{1..n})^2 - \mu_n^2} \quad (4-19)$$

Where $p(b_{1..n})$ is the probability of substring b given that it is of length n .

4.2 Branching Entropy Metrics and Predictive Information

The purpose of this section is to show that the different variants of branching entropy capture different aspects of predictive information. Here only the right branching entropy is presented since the results, and calculations for the left branching entropy are analogous.

4.2.1 Branching Entropy and Entropy Rate

Branching entropy is defined in **Definition 4-1** as

$$h(x_{1..n}) = H(X|x_{1..n}) = - \sum_{x \in \mathcal{X}} P(x|x_{1..n}) \log P(x|x_{1..n}). \quad (4-20)$$

Theorem 4-1 shows how Branching Entropy relates to predictive information.

Theorem 4-1. *Branching Entropy captures the block entropy rate such that*

$$h_\mu(n+1) = \sum_{x_{1..n} \in \mathcal{X}^n} p(x_{1..n}) h(x_{1..n}) \quad (4-21)$$

Proof: From equation (3-82)

$$h_\mu(n+1) = H(X_{n+1}|X_1, \dots, X_n) \quad (4-22)$$

$$= - \sum_{x^{n+1} \in \mathcal{X}^{n+1}} p(x_{1..n+1}) \log p(x_{n+1}|x_{1..n}). \quad (4-23)$$

$$= \sum_{x^n \in \mathcal{X}^n} p(x_{1..n}) \left[- \sum_{x_{n+1} \in \mathcal{X}} p(x_{n+1}|x_{1..n}) \log p(x_{n+1}|x_{1..n}) \right] \quad (4-24)$$

$$= \sum_{x^n \in \mathcal{X}^n} p(x_{1..n}) H(X_{n+1}|x_{1..n}) \quad (4-25)$$

$$= \sum_{x^n \in \mathcal{X}^n} p(x_{1..n}) h(x_{1..n}) \blacksquare \quad (4-26)$$

Therefore, the branching entropy is the local / point-wise analogue to block entropy rate. The next variant of branching entropy metric to consider is the variation of branching entropy (VBE).

4.2.2 Variation of Branching Entropy and Predictive Gain

The variation of branching entropy (VBE) is defined in **Definition 4-3** as

$$\delta h_{\rightarrow}(x_{1..n}) = h_{\rightarrow}(x_{1..n}) - h_{\rightarrow}(x_{2..n}) \quad (4-27)$$

Theorem 4-2. *The Variation of branching entropy(VBE) captures Predictive Gain such that*

$$\Delta h_{\mu}(n+1) = \sum_{x_{1..n} \in \mathcal{X}^n} p(x_{1..n}) \delta h(x_{1..n}) \quad (4-28)$$

Proof: From Equation (3-100)

$$\Delta h_{\mu}(n+1) = h_{\mu}(n+1) - h_{\mu}(n) \quad (4-29)$$

$$= \sum_{x_{1..n} \in \mathcal{X}^n} p(x_{1..n}) h(x_{1..n}) - \sum_{x_{1..n-1} \in \mathcal{X}^{n-1}} p(x_{1..n-1}) h(x_{1..n-1}) \quad (4-30)$$

Since the process one is dealing with is stochastic and therefore time shift invariant, $p(x_{1..n-1}) = p(x_{2..n})$ if $x_{1..n-1} = x_{2..n}$. This leads leading to a change of the second terms such that

$$\Delta h_{\mu}(n+1) = \sum_{x_{1..n} \in \mathcal{X}^n} p(x_{1..n}) h(x_{1..n}) - \sum_{x_{2..n} \in \mathcal{X}^{n-1}} p(x_{2..n}) h(x_{2..n}) \quad (4-31)$$

Since $h(x_{2..n})$ is not a function of x_1 , the second term could be regarded as a result of marginalising x_1 out where

$$\sum_{x_1 \in \mathcal{X}} p(x_{1..n}) h(x_{2..n}) = h(x_{2..n}) \sum_{x_1 \in \mathcal{X}} p(x_{1..n}) = h(x_{2..n}) p(x_{2..n}). \quad (4-32)$$

This leads equation (4-31) to

$$\Delta h_{\mu}(n+1) = \sum_{x_{1..n} \in \mathcal{X}^n} p(x_{1..n}) h(x_{1..n}) - \sum_{x_{2..n} \in \mathcal{X}^{n-1}} \sum_{x_1 \in \mathcal{X}} p(x_{1..n}) h(x_{2..n}) \quad (4-33)$$

$$= \sum_{x_{1..n} \in \mathcal{X}^n} p(x_{1..n}) h(x_{1..n}) - \sum_{x_{1..n} \in \mathcal{X}^n} p(x_{1..n}) h(x_{2..n}) \quad (4-34)$$

$$= \sum_{x_{1..n} \in \mathcal{X}^n} p(x_{1..n}) [h(x_{1..n}) - h(x_{2..n})] \quad (4-35)$$

$$= \sum_{x_{1..n} \in \mathcal{X}^n} p(x_{1..n}) \delta h(x_{1..n}) \blacksquare \quad (4-36)$$

This means that using variation of branching entropy (VBE) captures the predictive gain in the text as its local / point-wise analogue. The above proof has assumed right branching – an analogous proof applies for left branching.

The inference algorithm used in segmenting using VBE chooses the segments $x_{1..n}$ of text with the highest VBE. This is either done through bias normalisation or through the z-score normalisation, since $\Delta h_\mu(n+1)$ is the average of $\delta h(x_{1..n})$.

4.2.3 Normalisation and Predictive Information

Next, consider the normalisation functions, and what they mean for inference using branching entropy.

There are two branching entropy metrics that are normalisations of branching entropy and variation of branching entropy. They are normalisation by bias and normalisation by the z-score. From (4-15) and (4-16),

$$N_\mu h(b_{1..n}) = h(b_{1..n}) - \mu_n$$

$$N_z h(b_{1..n}) = \frac{h(b_{1..n}) - \mu_n}{\sigma_n}$$

Both of these normalisation schemes use the bias or the average of metric per length n , μ_n . The z-score normalisation uses the standard deviation σ_n per length n , and these from (4-17) (4-19) are

$$\mu_n = \sum_b p(b_{1..n}) h(b_{1..n})$$

$$\sigma_n = \sqrt{\sum_b p(b_{1..n}) [h(b_{1..n}) - \mu_n]^2}$$

From (4-21) the mean of the branching entropy at a particular length n is the entropy rate for a length one unit longer, $n+1$, and it is

$$\mu_n^{BE} = \sum_{b_{1..n} \in \chi^n} p(b_{1..n}) h(b_{1..n}) = h_\mu(n+1) \quad (4-37)$$

Therefore $\mu_n = h_\mu(n+1)$ for Branching Entropy. For Variation of Branching Entropy (VBE), the bias is calculated according to (4-22) and is the Predictive Gain for a unit longer length.

$$\mu_n^{VBE} = \sum_{b_{1..n} \in \chi^n} p(b_{1..n}) \delta h(b_{1..n}) = \Delta h_\mu(n+1) \quad (4-38)$$

The second moment of branching entropy, $h_\mu^2(n+1)$, and variation of branching entropy, $\Delta h_\mu^2(n+1)$ are then defined as

$$h_\mu^2(n+1) = \sum_{b_{1..n} \in \chi^n} p(b_{1..n}) [h(b_{1..n})]^2 \quad (4-39)$$

$$\Delta h_\mu^2(n+1) = \sum_{b_{1..n} \in \chi^n} p(b_{1..n}) [\delta h(b_{1..n})]^2 \quad (4-40)$$

The above gives us a way to define the variance of the branching entropy and variation of branching entropy as

$$\sigma_n^{BE} = \sqrt{h_\mu^2(n+1) - [\mu_n^{BE}]^2} \quad (4-41)$$

$$\sigma_n^{VBE} = \sqrt{\Delta h_\mu^2(n+1) - [\mu_n^{VBE}]^2} \quad (4-42)$$

In the above, μ_n is the mean of the metric for substrings of length n and σ_n is the standard deviation for the same length. The normalised metrics are therefore

$$N_\mu^{BE}(b_{1..n}) = BE(b_{1..n}) - h_\mu(n+1) \quad (4-43)$$

$$N_z^{BE}(b_{1..n}) = \frac{BE(b_{1..n}) - h_\mu(n+1)}{\sigma_n^{BE}} \quad (4-44)$$

$$N_\mu^{VBE}(b_{1..n}) = VBE(b_{1..n}) - \Delta h_\mu(n+1) \quad (4-45)$$

$$N_z^{VBE}(b_{1..n}) = \frac{VBE(b_{1..n}) - \Delta h_\mu(n+1)}{\sigma_n^{VBE}} \quad (4-46)$$

The objective in segmentation is to choose among segments of different lengths. Because $h_\mu(n) \geq h_\mu(n+1)$ and $\Delta h_\mu(n) \leq \Delta h_\mu(n+1)$ there needs to be a way to normalise the metrics across lengths depending on how length is used in inference.

4.2.4 Branching Entropy Inference and Predictive Information

Inference is done differently for the different metrics. As a start, there are eight (8) metrics to consider for the branching entropy and four metrics for bracing joint entropy. The eight are the base metric, i.e. Branching Entropy and Bracing Joint Entropy, the variation of each, and the two normalisations for the base metrics and the variation metrics:

Branching entropy metrics	Bracing joint entropy metrics
• Branching Entropy • Variation of Branching Entropy • Bias normalised Branching Entropy • Z-score normalised Branching Entropy • Bias normalised Variation of Branching Entropy • Z-score normalised Variation of Branching Entropy	• Bracing Joint Entropy • Variation of Bracing Joint Entropy • Bias normalised Bracing Joint Entropy • Z-score normalised Bracing Joint Entropy • Bias normalised Variation of Bracing Joint Entropy • Z-score normalised Variation of Bracing Joint Entropy

4.2.4.1 Inference in BE and VBE

For the un-normalised branching entropy metrics, BE and VBE, inference is about locating morpheme boundaries in the word. In a word of length $|w|$, this is done by first calculating a BE or VBE profile of the $|w| - 1$ possible boundary locations in the word. The metric is calculated for both directions. The final metric is either a sum or a minimum of the two.

For BE, the peaks in this profile are inferred to be boundaries. For VBE, a value of zero or higher in the profile is considered a boundary, because Predictive Gain is a decreasing/declining metric. If one keeps the boundary context size, n , remains the same, there should be no need for normalisation by length because the average of the BE metric for the length, the block entropy rate $h_\mu(n+1)$, is the same for the size. However, the requirement for normalisation arises due to the following.

However, the context size shrinks for locations/boundaries close to the edges, specifically for locations less than n from the edges,. This means that the un-normalised BE and VBE metrics are incomparable at locations less than n from the word edges. Because the block entropy rate declines as n increases, locations less than n from the word edges have higher BE on average, and therefore increase the possibility of boundaries at the edges. Because the Predictive Gain, $\Delta h_\mu(n + 1)$, is an increasing metric, boundaries at the locations less than n from the word edges have a lower chance of becoming boundaries. Therefore, normalisation is required at locations closer to the word edge. This is an important finding that the information theory provides and future work should investigate this.

What makes using word profile peaks to infer a morpheme boundary interesting is that a boundary needs agreement from both straddling substrings, and these are two different substrings.

4.2.4.2 Inference in normalised BE and VBE

Ordinarily, the normalised metrics are used in direct morpheme identification instead of morpheme boundary inference. The algorithm for inferences of morphemes in a sequence is according to equation (4-5) and (4-6), where s is a possible segmentation, w_i is the possible morpheme of that segmentation, $a(w_i)$ is the normalised metric as

$$s^*(w) = \arg \max_{s \in Segs(w)} \sum_{w_i \in s} a(w_i) \cdot len(w_i)$$

Above $Segs(w)$ are possible segmentations of w , w_i is a possible morpheme from segmentation s , and $len(w_i)$ is the length of the segment. What this means is that for a corpus of words the algorithm chooses segments of different lengths that increase the predictive information whilst constrained by the need to combine into words.

This allows for comparison of possible morphemes w_i of different lengths. From an information theoretic perspective normalisation using bias is clear as the first moment of branching entropy and variation of branching entropy, the block entropy rate and predictive gain respectively is defined. What is not clear is the meaning of the second moment, as the z-score uses the variance of the metrics, the shifted second moment of the metrics. Intuitively, however, using the z-score should normalise out any average effects from the calculation, including spread, such that possible morphs of different sizes are directly comparable. This however might mean neutralisation of any effect, even linguistic phenomena, that applies to all segments of a particular length. This might be an issue for isiXhosa as most closed class morphemes are bigrams.

4.2.5 Conclusions on Branching Entropy Metrics and Predictive Information

In this section, proof that branching entropy metrics are related to predictive information and capture the different aspects of predictive information, i.e. block entropy rate and predictive gain, were presented.

In addition, inference using unnormalised metric was shown to tend to oversegment the edges of a word in the case of Branching Entropy, and undersegments the edges in the case of Variation of Branching Entropy.

It was also shown that z-score normalisation has unintended consequences, that of smoothing out any n-gram length specific linguistic phenomena, which might be a problem for isiXhosa as most of its morphemes that are in closed classes have two letters.

4.3 Bi-directional Entropy Language Modelling

The metrics presented above depend on a language model. *Section 2.6.9 Language Model based segmentation* provides some detail of language modelling and its use in text segmentation. In that section, the reason for the need for smoothing in language models was also explained.

Language modelling as a task has an extensive research history, so there is no need to delve into the general aspects of it in this document. However, Branching Entropy, and metrics derived from it, require modelling in two directions. This can be easily achieved by using two language models, one for each direction as Magistry does (Magistry, 2013), but there is a cost involved. In this section, work on combining the two languages models into one and extending them into an entropy model is detailed. This model is such that a dictionary entry points to a vector of two values, the right branching and the left branching values. This reduced the memory footprint. The model is called the Bi-directional Branching Entropy Language Model (BELM). Also, because smoothing assists in improving language processing performance, a BELM that is smoothed with modified Kneser-Ney smoothing (Chen & Goodman, 1998) is implemented. In addition, the language models were also extended to include an option for using all possible n-gram levels in one model instead of having a maximum n-gram level limit, i.e. an infinite-gram. All the other calculations are then done as specified above, based on these language models. The algorithms are briefly described below.

The implementation presented in this section used the Python programming language (van Rossum, 2007), and the model runs in a single thread. The interest is not much on speed other than proving that this kind of model is possible. There are better implementations of uni-directional

language models (Stolcke, 2002; Heafield, 2011), and the implementation could be ported to optimised programming languages in the future.

4.3.1 Un-smoothed bi-directional Branching Entropy language model flow

The input to the unsmoothed modelling is a list of one directional (left-to-right) n-gram strings with frequency counts (n-gram, f) or a mapping of n-gram strings to frequency counts, and the process returns a single Bi-directional Branching Entropy Language Model (BELM) with branching entropy values for both directions. Figure 4-5 shows the process of the modelling.

The first is the creation of a mapping, $M: x \rightarrow (f, null)$, from the n-gram string to a tuple containing the frequency and a space for the appropriate reverse count, and an empty mapping H for the branching entropy model.

The Mapping M is then updated with reverse frequencies such that each n-gram x is mapped to a list of two counts such that,

$$M(x) \rightarrow \begin{pmatrix} \# \vec{x} \\ \# \overleftarrow{x} \end{pmatrix} \quad (4-47)$$

To find the appropriate location for a reverse count a reverse key is generated for each n-gram $x_{1..n}$. Figure 4-4 shows how the reverse key is generated from the normal key.

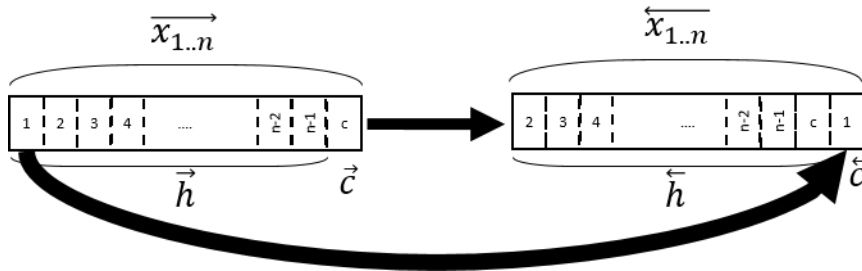


Figure 4-4: Representation of Reverse Key Transformation

The reverse key is generated from a normal key. The normal key, $x_{1..n}$, is split into a context, h and the last letter c , such that $x_{1..n} = hc$. This implies that the context $h = x_{1..n-1}$ and the last letter $c = x_n$. Language models estimate the conditional probability $p(c|h)$. For the reverse key $x_{1..n} = ch$, such that $h = x_{2..n}$ and $c = x_1$. The value of the reverse key is stored in the same location as if it was hc . This allows for handling the mapping as if it was a single left to right mapping, allowing

for sorting by context which is required for calculating the directional probabilities $\overrightarrow{p(c|h)}$ and $\overleftarrow{p(c|h)}$ which are then used for the directional branching entropies.

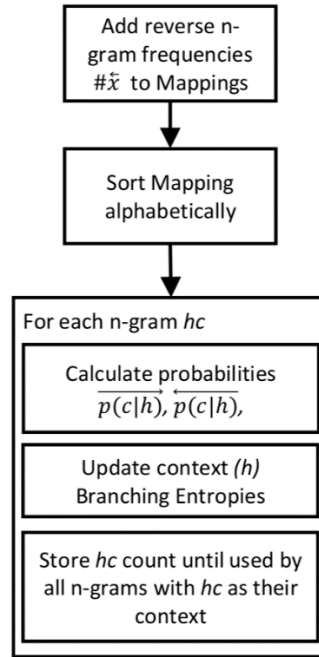


Figure 4-5: Representation of the Un-Smoothed BELM process

The keys to the mapping are then sorted alphabetically. The sorting allows the process to do a single pass through the n-gram strings frequency counts, as context always precedes the full counts. The probabilities are calculated such that

$$\overrightarrow{p(c|h)} = \frac{\#hc}{\#h} \quad (4-48)$$

$$\overleftarrow{p(c|h)} = \frac{\#ch}{\#h} \quad (4-49)$$

A buffer for increasing contexts is kept until exhaustion and then calculating the conditional probabilities is unrolled, adding to the branching entropies. This is important when dealing with large data amounts counts as in this case.

The resulting BELM model is a mapping H such that

$$H(x_{1..n}) \rightarrow \begin{pmatrix} h_{\rightarrow}(x_{1..n}) \\ h_{\leftarrow}(x_{1..n}) \end{pmatrix} \quad (4-50)$$

The branching entropies are calculated according to (Zhikov et al., 2013). The probabilities, $\overrightarrow{p(c|h)}$ and $\overleftarrow{p(c|h)}$, are discarded after use.

4.3.2 The modified Kneser-Ney smoothed bi-directional Branch Entropy language model

For this study, there was a requirement to establish whether smoothing had any effect on the performance of the branching entropy segmenter. Smoothing is necessary where there is data sparsity, which is the case in higher-order character n-grams, i.e. long n-gram strings.

A bi-directional mKN smoothed BELM was also implemented. Input to it is also a list of one directional (left-to-right) n-gram strings with frequency counts (n-gram, f) or a mapping of n-gram strings to frequency counts and the process returns a single bi-directional Branching Entropy Language Model (BELM) with branching entropies for both directions. The process is shown in Figure 4-6.

The conditional probabilities are calculated according to (Chen & Goodman, 1998) and the branching entries are according to (Zhikov et al., 2013).

Having created the n-gram frequency mapping with reverse counts as in section 4.3.1 above, two empty conditional probabilities models are also created, one for the current level, P , and one for the previous n-gram length level, P' .

Next, all the keys from the mapping M are extracted and this is used to create the adjusted counts required in mKN smoothing. Because the work requires modelling both directions, the $\bullet x$ and $x\bullet$ counts are needed for extrapolation. This is done by simply removing the last letter from each key for $x\bullet$ and then doing counts, and the first letter from the keys for $\bullet x$ and doing the counts. The new counts, the adjusted counts, are used to update mapping M accordingly. See Chen and Goodman (1998), and Heafield (2011) for the meaning of the adjusted counts, $\bullet x$ and $x\bullet$.

Next, mapping, M , is sorted by length and n-gram context. The sorting by n-gram lengths and n-gram contexts ensured that lower level n-grams, shorter n-grams, are processed first as their results are required for the interpolation of higher-level, longer, n-grams. This also ensured that discount values could be calculated independently for each level. Lastly, this ensured that n-grams are clustered by context, which is required for mKN smoothing.

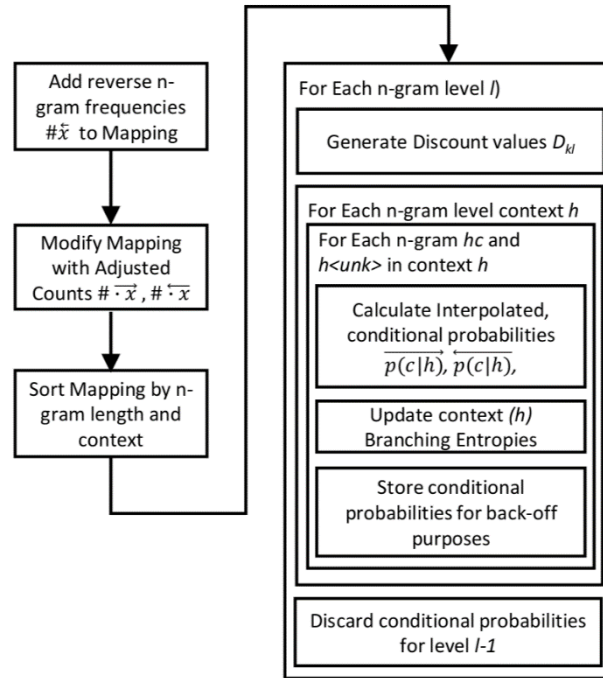


Figure 4-6: Graphical representation of the mKN Smoothed BELM Process

At a particular n-gram length level, n , conditional probabilities from the previous n-gram length level, $n - 1$ are needed, so the conditional probabilities model of only that level, P' are kept. The conditional probabilities of the current level are calculated using interpolation, stored in P and the branching entropy model is updated at the same time. When the level is exhausted, the conditional probabilities model of the previous level, P' , is discarded and is replaced with one from the current level, P and the modelling continues on to the next level.

All this ensured that only two passes through the n-gram counts are required. One for generating adjusted counts and collection interpolation statistics and the second for calculating the probabilities and updating the BELM.

4.3.3 Summary of Bidirectional Entropy Language Models

In this section, an overview is given of the work involved in creating the Bidirectional Entropy Language Models. This highlights how the directional counts are combined into a single entry and how the adjusted counts can be done easily.

The conditional probabilities models are discarded and only the branching entropy models persist. Entries not in the model are regarded to have zero entropy.

4.4 Conclusions

This chapter presented information theory based, unsupervised, morphological segmentation techniques.

Two approaches to unsupervised morphological segmentation are presented, namely Branching Entropy and the introduction of a new technique of Bracing Joint Entropy. It is shown that Branching Entropy extracts and uses predictive information from the data to predict segmentations.

As part of the development of branching entropy techniques, the bi-directional branching entropy language modelling technique, that combines both branching entropy directions into one model and used for both un-smoothed and smoothed language modelling is presented.

The evaluation of the techniques identified in this chapter will follow in Chapter 7.

5. ISIXHOSA SEGMENTS

This chapter establishes the heuristics that guide the language models used to assess the non-parametric language model based segmentation. This relies on a deep understanding of the grammar rules of the isiXhosa words. The purpose of this is to cover the language's grammar such that segmentation heuristics, which generalise over the word categories, can be established.

To build a model for the segmentation of isiXhosa words, the morphology of its words must be understood. In this chapter, the question of what the appropriate surface analysis for a category of isiXhosa words should be for automated surface segmentation is investigated. According to Bender (2013:1), "knowledge of linguistic structure leads to better features for machine learning." The understanding of isiXhosa, in the context of surface segmentation, is therefore important.

The intention, therefore, is not to establish all the rules of the language (as in implementing a rule-based system), but to find a set of heuristics that ensures identification of all the possible segmentation boundaries. This implies that the heuristics might decompose a possible morpheme into pieces, but this is considered a worthwhile compromise.

This chapter first describes what surface segmentation is in isiXhosa. This is followed by an enumeration of isiXhosa word categories because segmentation is specific to a word category in isiXhosa. This is followed by details of how each category should be surface segmented.

The detailed work is then summarised into a set of heuristics that can be used later in the study. Lastly, a summary of dependencies of the components in the language is established.

5.1 Surface Segmentation

Morphological analysis is the task of splitting a language token, in this case, a word into its constituent units (Manning & Schütze, 1999), e.g. the segmentation of a word into morphemes; and thereafter classifying these. Morphemes are the smallest meaning-bearing component of a word (Kosch, 2006). This is a preprocessing step in information retrieval, translation and other operations for languages rich in inflection and derivation.

A differentiation is made by Hammarström et al. (2011) between morphological segmentation, which splits words into constituent morphemes, and morphological analysis, which splits words and also classifies the identified morphemes. This differentiation originated from Zwicky (1966).

The focus of this work is specifically on surface segmentation which does not handle morpheme boundary fusion. Surface segmentation segments a word such that a concatenation of the

segments results in the word. As an example the morphological segmentation of *ukwanda* is *u-ku-and-a*, however, in this study, the word is surface segmented to *u-kw-and-a*.

5.2 Word Categories of isiXhosa

Since different categories of isiXhosa words are made up of different free or bound morphemes, it is better to look at what surface segmentation would be for each word category. For isiXhosa, word categories have evolved over the years (Boyce, 1844; Pahl, 1982; Louw et al., 1984).

Louw et al.'s (1984) approach to isiXhosa word categorisation is morphologically based and provides for thirteen word categories, namely noun [*isibizo*], pronoun [*isimelabizo*] (absolute, demonstrative, quantitative, and emphatics absolute), verbal [*isenzi*] (auxiliary verb [*isenzi esilabalabayo*]), adjective [*isiphawuli*], relative [*isibaluli*], enumerative relative [*isichazi sobalo*], possessive [*isimnini*], copulative [*isibanjalo*], verbal relative [*isibaluli esakhiwe kwisenzi*], descriptive [*isihlomelo*], ideophone [*isifanekisozwi*], conjunction [*isihlahnganisi*] and interjection [*isikhuzo*].

The *Greater Dictionary of Xhosa: Volume 3 (Q to Z)* (Pahl et al., 1989) provides for a traditional nomenclature of isiXhosa word categories i.e. noun, pronoun, verb, qualificative, conjunction, adverb and ideophone.

The *Greater Dictionary of isiXhosa: Volumes 1: (A to J) and 2: (K to P)* (Mini & Tshabe, 2003:xxxiv; Tshabe & Shoba, 2006:xxv) provides a slightly different categorisation with twelve word categories for isiXhosa i.e. noun [*isibizo*], pronoun [*isimelabizo*], verb [*isenzi*], adjective [*isiphawuli*], copulative [*isibanjalo*], qualificative [*isichazil*], possessive [*isimnini*], quantificative [*isahluli*], enumerative [*isiquki*], ideophone [*isifanekisozwi*], adverb [*isihlomelo*], and form word [*isikhapi*].

For surface segmenting isiXhosa words, eight main word categories will be used in this document, i.e. noun, pronoun, qualificative, predicate, descriptive, ideophone, conjunction and interjection. This main categorisation is a fusion of the above categories.

Each main category consists of sub-categories, e.g. the different copulatives in The *Greater Dictionary of Xhosa: Volume 3* (Pahl et al., 1989:xxxvi) and parts of speech in Louw et al. (1984:16). In this work, the hierarchy of isiXhosa categories used is that developed by Mzamo (2015). The flow of this section also follows that work :

- Noun (isibizo)
- Pronoun (isimelabizo)
 - Absolute pronoun (esoqobo)
 - Demonstrative (esokukhomba/ esokwalatha/ isikhombisi/ isalathisi/)
 - Quantitative pronouns (ezoquko)
 - Differentiative Pronoun (isahluli)
 - Superlative (esobalaselo)
- Qualificative (isichazi)
 - Adjective (Isiphawuli)
 - Relative (Isibaluli)
 - Possessive (isimnini)
 - Enumerative (esobalo)
- Predicate (Isivisa)
 - Verb (isenzi)
 - Copulative (isibanjalo)
 - Indicative copulative (isibayiyo)
 - Qualificative copulative (isibanjani/ isibanjalo sochazo/ isichazi)
 - Locative Demonstrative Copulative (isibandawo/ izibanjalo-zalathandawo)
 - Possessive copulative (esinesibandakanyi/ isibanayo)
 - Distributive (esinokwaba)
- Descriptive/Adverb (isihlomelo)
 - Locatives (Descriptive of place) (isalathandawo)
 - Descriptive of Manner (sihlomelo sobunjani)
 - Descriptive of Time
- Ideophone (isifanekisozwi)
- Conjunction (isihlanganisi)
- Interjection (isikhuzo)

Figure 5-1: The hierarchy of IsiXhosa word categories we use

The hierarchy of categories is important because in each main category there are nuances specific to the sub-categories.

5.3 IsiXhosa Segmentation Specifics

Each word in isiXhosa has a root and affixes, i.e. suffixes, prefixes or circumfixes. A root carries the main meaning of the word, and always appears with at least one suffix. The shortest word form appears as a stem. A stem is a root suffixed with a vowel, hence the term - terminal vowel, e.g. *hamba* (go) has the root *hamb* and the terminal vowel *a*. A circumfix is the "simultaneous affixation of a prefix and suffix to a root or a stem to express a single meaning" (Kosch, 2006). An example of a circumfix in isiXhosa is the combination *a-...-anga* used for negation, e.g. *akahambanga* (he/she **did not** go). Most stems, which are the meaning carrying constituents of words, consist of two syllables (Meinhof, 1932:36).

Since isiXhosa is an agglutinating language displaying a measure of polysynthesis, each word is made up of numerous morphemes (Kosch, 2006). Word creation also uses fusion, where morpheme edges are fused. An example is the conversion of *umntu* (person) with the suffix *-ana* to form the diminutive *umntwana*. In this case, the *u* and *a* vowels are fused into *wa*. This is the reason for the deviation in this study from traditional grammar. The difference is that morphemes, such as noun stems, are broken into further pieces, e.g. *nt-u*, because the last vowel in a noun stem is the one that could undergo fusion.

Affixes are made up of one or more morphemes. An example is the prefix string *su-ku-ndi-* in the word *sukundibamba* (don't hold me). The sequence of the morphemes is strict in isiXhosa words, and depends on the word type (Louw et al., 1984). Most of the affixes in isiXhosa are bound morphemes, so they never appear independently as meaningful words (Pahl, 1982).

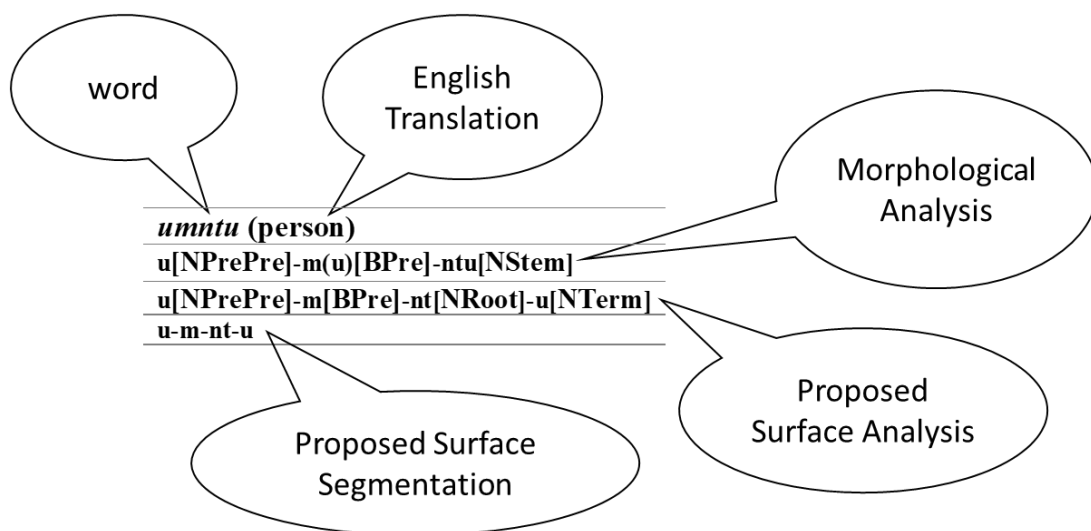


Figure 5-2: Glossing Description

Segmentation for each word category will be specified using a form of glossing. A gloss gives information about the meaning and grammatical properties of individual words and parts of words (Comrie et al., 2008). For clarity, the glosses (tagset) used are based on the ZulMorph (Pretorius & Bosch, 2018) tagset². Glossing tags are in square brackets, e.g. [BPre] for the basic prefix. In addition, hyphens, '-', are used to separate segments, e.g. *u[NPrePre]-m[BPre]-ntu[NStem]*. Because the primary objective is to get to a surface segmentation from the morphological glossing, this chapter will not specify noun class, so a gloss of the form *u[NPrePre][1]-m[BPre][1]-ntu[NStem][1]* will be glossed *u[NPrePre]-m[BPre]-ntu[NStem]*. Some new tags might be used if

² <https://portal.sadilar.org/FiniteState/demo/zulmorph/doc.html#tagset>

the need arises. See Figure 5-2 for the description of the glossing. If the morphological analysis matches the proposed segmentation, it will not be duplicated. The glosses used are tabled in Annexure A.

This section describes the appropriate manner for segmenting the surface form of each word in detail. As the effect of nouns permeates all other word categories in isiXhosa (McLaren, 1948:14), they are the first category considered. The following sections follow the same structure as Mzamo (2015), which was about lemmatisation, and the wording might seem similar.

This similarity has to do with the definition of the word category and how that category is used and formed in the language. The suggested surface segmentations that are specified for each category, which are the main crux and innovation of this chapter, do not appear in Mzamo (2015).

5.3.1 Noun (Isibizo)

There are two types of substantives in isiXhosa: nouns and pronouns. Nouns are dominant in that they affect the morphology of many of the other word categories. The part of the isiXhosa nouns that affects all the other isiXhosa words is the noun prefixes which expose the class of the noun (Pahl, 1982:9; McLaren, 1948:16; Louw et al., 1984:18). The noun class structure is a system of classifying Bantu nouns according to prefixes specific to a class. These prefixes distinguish between singular and plural forms. In this work the classification specified in the *Greater Dictionary of isiXhosa: Volume 2* (2003:1110) is used.

Table 5-1: Noun Class Prefixes

Class	Singular	Class	Plural
1	<i>u-m(u)-</i>	2	<i>a-ba-, a-be-, ab-</i>
1a	<i>u-</i>	2a	<i>oo-</i>
3	<i>u-m(u)-</i>	4	<i>i-mi-</i>
5	<i>i-li-, i-</i>	6	<i>a-ma-, a-me-</i>
7	<i>i-si-, i-sa-</i>	8	<i>i-zi-</i>
9	<i>i-, in-</i>	10	<i>i-zi(n)-, ii(n)-</i>
11	<i>u(lu)-</i>	10	<i>i-zii-, ii-</i>
14	<i>u-bu-</i>		
15 and 17	<i>u-ku-, ukw-</i>		

Some parts of the prefix are hidden. In Table 5-1 one can see that some class prefixes are segmented. This is to show that noun prefixes are made up of two parts, i.e. an initial vowel called the pre-prefix (Pahl, 1982:9; Louw et al., 1984:19) and the noun prefix proper (McLaren, 1948:27) called the basic prefix (McLaren, 1948:19). This noun class structure affects how other word categories relate to nouns, as will be shown later in the chapter. This effect is referred to as

concordial agreement (Mncube, n.d:13; McLaren, 1948:16; Louw et al., 1984:17). The basic prefixes are listed in Table 5-2.

Table 5-2: Noun Basic Prefixes

Class	Singular		Plural
1 and 1a	<i>mu-</i>	2 and 2a	<i>ba-</i>
3	<i>mu-</i>	4	<i>mi-</i>
5	<i>li-</i>	6	<i>ma-</i>
7	<i>si-</i>	8	<i>zi-</i>
9	<i>i-</i>	10	<i>zi-</i>
11	<i>lu-</i>	10	<i>zi-</i>
14	<i>bu-</i>		
15 and 17	<i>ku-</i>		

This basic prefix forms the base for concords in the formation of other isiXhosa word categories, e.g. relatives, adjectives, enumeratives and copulatives.

The basic segmentation for a noun would therefore involve identifying the pre-prefix, the basic prefix and the root stem.

<i>umntu</i> (person)
u[NPrePre]-m(u)[BPre]-ntu[NStem]
u[NPrePre]-m[BPre]-nt[NRoot]-u[NTerm]
u-m-nt-u

The noun basic prefix involves dropping the pre-prefix from the noun class prefixes. Class 2a is an exception to this rule.

On the suffix, IsiXhosa nouns can take an augmentative *-kazi* or a diminutive suffix *-ana*. Such a segmentation is as shown below.

<i>umthikazi</i> (large tree)
u[NPrePre]-m(u)[BPre]-thi[NStem]-kazi[AugSuf]
u[NPrePre]-m[BPre]-thi[NStem]-kaz[AugSuf]-i[NTerm]
u-m-thi-kaz-i

<i>umntwana</i> (child)
u[NPrePre]-m(u)[BPre]-ntu[NStem]-ana[DimSuf]
u[NPrePre]-m[BPre]-ntw[NRoot]-an[DimSuf]-a[NTerm]
u-m-ntw-an-a

5.3.1.1 Noun Concords

Concordial agreement permeates isiXhosa sentences and links words in a sentence. In isiXhosa words are characterised by their concordial element which specifies the noun class to which they refer (Louw et al., 1984:57). The concordial element is a prefixal morpheme that relates the word to a noun class. This referential characteristic allows the word to be used in the place of a noun, a pronominal character.

These concordial morphemes normally bind to stems of other word categories to form the word, e.g. “*ndi-mi*” (I stand), for subject concords, and “*u-ndi-mele*” (he/she waited for **me**), for object concord. Concords are a closed group and shown in Table 5-3 and Table 5-4 for subject and object concords respectively.

Table 5-3: Subject Concords

Person/Class	Singular		Plural
1st Person	<i>ndi-</i>		<i>si-</i>
2nd person	<i>u-</i>		<i>ni-</i>
3rd person			
Class 1 and 1a	<i>u-</i>	Class 2 and 2a	<i>ba</i>
3	<i>u-</i>	4	<i>i-</i>
5	<i>li-</i>	6	<i>a-</i>
7	<i>si-</i>	8	<i>zi-</i>
9	<i>i-</i>	10	<i>zi-</i>
11	<i>lu-</i>	10	<i>zi-</i>
14	<i>bu-</i>		
15 and 17	<i>ku-</i>		

Because most word categories relate to the subject and object via the concord they also have concords, but those concords are formulated from the noun basic prefix, and the subject and object concords. One characteristic of concordial agreement is that it allows for the omission of the noun in the use of the language.

Table 5-4: Object Concords

Person/Class	Singular		Plural
1st Person	<i>-ndi-</i>		<i>-si-</i>
2nd person	<i>-ku-</i>		<i>-ni-</i>
3rd person			
Class 1 and 1a	<i>-m-</i>	Class 2 and 2a	<i>-ba</i>
3	<i>-wu-</i>	4	<i>-yi-</i>
5	<i>-li-</i>	6	<i>-wa-</i>
7	<i>-si-</i>	8	<i>-zi-</i>
9	<i>-yi-</i>	10	<i>-zi-</i>
11	<i>-lu-</i>		
14	<i>-bu-</i>		
15 and 17	<i>-ku-</i>		

Surface segmentation as proposed below preserves the number of segments, and the concatenation of the segments results in the word.

***uhambile* (he/she is gone)**

u[SC]-hamba[VStem]-ile[VTPerf]

u[SC]-hamb[VRoot]-il[VTPerf]-e[VT]

u-hamb-il-e

***undihambele* (he/she visited me)**

u[SC]-ndi[OC]-hamb[VRoot]-a[VT]-ele[VTPerf]

u[SC]-ndi[OC]-hamb[VRoot]-el[VTPerf]-e[VT]

u-ndi-hamb-el-e

5.3.2 Pronoun (Isimelabizo)

The main pronoun sub-categories for isiXhosa are, absolute pronouns, demonstrative pronouns, quantitative pronouns and superlatives.

5.3.2.1 Absolute Pronoun (Esoqobo)

This is a closed group of full words. Absolute pronouns are characterised by a stem for each noun class and a common suffix *-na*, e.g. *thi-na*, *wo-na*, *so-na*, *ko-na*, *ye-na*. The list of absolute pronouns is shown below.

Table 5-5: Absolute Pronouns

Person/Class	Singular		Plural
1st Person	<i>m-na</i>		<i>thi-na</i>
2nd person	<i>we-na</i>		<i>ni-na</i>
3rd person			
Class 1 and 1a	<i>ye-na</i>	Class 2 and 2a	<i>bo-na</i>
3	<i>wo-na</i>	4	<i>yo-na</i>
5	<i>lo-na</i>	6	<i>wo-na</i>
7	<i>so-na</i>	8	<i>zo-na</i>
9	<i>yo-na</i>	10	<i>zo-na</i>
11	<i>lo-na</i>	10	<i>zo-na</i>
14	<i>bo-na</i>		
15 and 17	<i>ko-na</i>		

As such, absolute pronouns will segment as the absolute pronominal stem followed by the common suffix *-na*, e.g:

***mna* (me)**

m[PronStem]-na[PronSuf]

m-na

***wena* (you)**

we[PronStem]-na[PronSuf]

we-na

The above absolute pronouns are used to form some locatives, possessives, copulatives and adverbs (Pahl, 1982:38). This formation is characterised by the dropping of the absolute pronoun suffix:

Na+yena > *naye*
Ya+zona > *yazo*
Nga+bona > *ngabo*

This will be handled later in the document.

5.3.2.2 Demonstrative Pronouns (Isilathisi)

IsiXhosa also uses demonstrative pronouns. There are three types; firstly for demonstrating at positional types: here [this], secondly near [that]; and lastly at a distance [that yonder], e.g. *lo*, *lowo*, *lowaa* (Louw et al., 1984:60; McLaren, 1948:57; Pahl, 1982:39). These are also a closed set depending on the class of the related noun and are enumerated in Louw et al. (1984:61).

Demonstratives are formed from the demonstrative element /a- and the class subject concord as shown in the examples below.

<i>esi</i> (this one)
(l)a[Dem]-si[SC]
e[SC]-s[DemRoot]-i[DemTerm]
e-s-i

<i>leyo</i> (that one)
(l)a[Dem]-y[Drift]-o[SC]
le[SC]-y[Drift]-o[SC]
le-y-o

<i>lowa</i> (that one yonder)
(l)a[Dem]-o[SC]-w[Drift]-a[SC]
lo[Dem]-w[Drift]-o[SC]
lo-w-a

In the examples above, the *a* in /a- assimilates to the subject concord vowel *i* to form *e*. In the second case above, the pronoun is made from the concatenation /a- and the subject concord –o. In this case, a vocalic/drift consonant is added to vocalise –o to –yo, and –a to –wa for the yonder pronoun.

5.3.2.3 Quantitative Pronoun (Esoquuko)

Quantitative pronouns have the stems: *-nke* (*all*), *-dwa* (*only*) and *-numeral* (Louw et al., 1984:68; Pahl, 1982:43). Examples are:

Table 5-6: Quantitative Pronouns

Quantitative stem	Examples	Translated
-nke	<i>so-nke</i> <i>zo-nke:</i>	all of us all of them
-dwa	<i>ye-dwa:</i> <i>so-dwa:</i>	only him us alone
-numeral	<i>so-babini:</i> <i>zo-mbini:</i>	the two of us the two of them

The surface segmentation for quantitative pronouns is as follows:

<i>sonke</i> (all of us)
so[QC]-nke[QuantStem]
so[QC]-nk[QuantRoot]- e[QuantTerm]
so-nk-e

<i>zombini</i> (that one yonder)
zo[QC]-mbini[QuantStem]
zo[QC]-mbin[QuantRoot]- i[QuantTerm]
zo-mbin-i

5.3.2.4 Differentiative Pronoun (Esokwahlula)

Differentiative pronouns convert a differentiative qualificative with stem *–mbi* [another] and *–phi*, (which of these) (McLaren, 1948:67) into a pronoun. For the differentiative pronouns derived from the qualificative stem *–phi*, the prefixes *a-*, *e-* and *o-* are added (Pahl et al., 1989:703). These pronouns are also a closed set that is related to the noun class and are listed in Addendum 15 of Volume 2 of the Greater Dictionary of IsiXhosa (2003).

Differentiative Pronouns can be surface segmented as shown below:

<i>bambi</i> (other persons)
ba[SC]-mbi[DiffStem]
ba[SC]-mb[DiffRoot]- i[DiffTerm]
ba-mb-i

<i>omphi</i> (which person)
o[DiffPref]-m[SC]-phi[DiffStem]
o[DiffPref]-m[SC]-ph[DiffRoot]- i[DiffTerm]
o-m-ph-i

5.3.2.5 Superlative Pronoun (Esobalulo)

Emphatic absolute pronouns or Superlatives, are based on absolute pronouns and serve the purpose of emphasis (Louw et al., 1984:74) by showing that a particular noun should be highlighted in some manner (Pahl, 1982:39). These are noun class specific, only exist in the third person and are listed in Addendum 6 of Volume 2 of the Greater Dictionary of IsiXhosa (2003):

Superlatives are created by prefixing an absolute pronoun with a superlative prefix *a-*, *e-* or *o-* depending on the class. These pronouns are therefore segmented as follows.

oyona (thee “person”)
o[SupPref]-ye[PronStem]-na[PronSuf]
o-ye-na

eyona (thee “person”)
e[SupPref]-yo[PronStem]-na[PronSuff]
e-yo-na

5.3.3 Qualificative (Isichazi)

This word category describes the quality of a noun. Qualificatives differ in the origin of their stem. Such stems belong to other word categories, like a verb or pronoun, except for adjectives and enumeratives. The sub-categories are adjectives relatives, possessives and enumeratives.

5.3.3.1 Adjective (Isiphawuli)

Adjectives are characterised by a noun class concord prefix, the adjective concord, and a stem. The characteristic of the substantive is carried by the stem of an adjective e.g. *-de* (height or length), or *-hle* (beauty or goodness). The adjective concords are listed in Table 5-7.

Table 5-7: List of Adjective Concords (Louw et al., 1984:77)

Class	Singular	Plural
1 and 1a	<i>om-</i>	2 and 2a <i>aba-</i>
3	<i>om-</i>	4 <i>emi-</i>
5	<i>eli-</i>	6 <i>ama-</i>
7	<i>esi</i>	8 <i>ezi-</i>
9	<i>en-</i>	10 <i>ezin-</i>
11	<i>olu-</i>	10 <i>ezin-</i>
14	<i>obu-</i>	
15 and 17	<i>oku-</i>	

There are only nineteen adjective stems in isiXhosa and are listed in Pahl (1982:46) and Louw et al. (1984:78).

The adjective concord is made up of a qualificative prefix *a-* and the basic prefix. The qualificative prefix assimilates the basic prefix’s vowel such that *a- + -m(u) > om*. This results in the following segmentation of adjectives.

***omde* (tall one)**

a[QualPref]-m(u)[BPre]-de[AdjStem]

o[QualPref]-m[BPre]-d[AdjRoot]-e[AdjTerm]

o-m-d-e

***abafutshane* (short ones)**

a[QualPref]-ba[BPre]-futshane[AdjStem]

a[QualPref]-ba[BPre]-futshan[AdjRoot]-e[AdjTerm]

a-ba-futshan-e

***entle* (a beautiful one)**

a[QualPref]-i(n)[BPre]-hle[AdjStem]

e[QualPref]-n[BPre]-tl[AdjRoot]-e[AdjTerm]

e-n-tl-e

Adjectives allow for negation (Louw et al., 1984:81). An often neglected adjective negation form adds a circumfix *-nga-* .. *-anga-* and is segmented as follows.

***ongamdanga* (the one who's not tall)**

a[QualPref] -nga[NegPre]-m(u)[BPre]-de[AdjStem] -nga[NegCirc]

o[QualPref]-nga[NegPre]-m[BPre]-d[AdjRoot] -ang[NegCirc]-a[AdjTerm]

o-nga-m-d-ang-a

***olungeludanga* (the one who's not tall)**

a[QualPref]-lu[BPre]-nga[NegPre]-lu[AuxVStem]-de[AdjStem]-nga[NegCirc]

o[QualPref]-lu[BPre]-nge[NegPre]-lu[BPre]-d[AdjRoot]-ang[NegCirc]-a[AdjTerm]

o-lu-nge-lu-d-ang-a

It should be noted that similar to nouns, adjectives can take an augmentative suffix *-kazi* or diminutive suffix *-ana* (Louw et al., 1984:81).

5.3.3.2 Relative (Isibaluli)

Relatives have a similar purpose as adjectives but differ in the source of the stems. The relative is formed from relative stems and stems of other word categories (Louw et al., 1984:83), making the relative a noun qualifier that may inherit stems from other word categories. Relatives are produced by prefixing a stem from another word category with the relative concord.

The relative concords are enumerated below:

Table 5-8: Relative Concords (Pahl et al., 1989:685; Louw et al., 1984:84)

Person/Class	Singular		Plural
1st Person	<i>endi-</i>		<i>esi-</i>
2nd person	<i>o-</i>		<i>eni-</i>
3rd person			
Class 1 and 1a	<i>o-</i>	Class 2 and 2a	<i>aba-</i>
3	<i>o-</i>	4	<i>e-</i>
5	<i>eli-</i>	6	<i>a-</i>
7	<i>esi-</i>	8	<i>ezi-</i>
9	<i>e-</i>	10	<i>ezi-</i>
11	<i>olu-</i>	10	<i>ezi-</i>
14	<i>obu-</i>		
15 and 17	<i>oku-</i>		

Relatives from relative stems are segmented as:

<i>obumnyama</i> (that is black)
a[RelPref]-bu[SC]-mnyama[RelStem]
o[RelPref]-bu[BPre]- mnyama[RelRoot]-a[RelTerm]
o-bu-mnyam-a

Relatives from noun stems are segmented as follows;

<i>ongumthuthi</i> (who is a courier)
a[RelPref]-ngu[CopPre] u[NPrePre]-m[BPre]-thuthi[NStem]
o[RelPref]-ngu[CopPre]-m[BPre]-thuth[NRoot]-i[NTerm]
o-ngu-m-thuth-i

It should be noted that similar to nouns, relatives can take an augmentative suffix *-kazi* or diminutive suffix *-ana* (Louw et al., 1984:87-88). Relatives from possessives are segmented as follows;

<i>elabo</i> (theirs)
a[RelPref]-la[PossConc]-bo[PronStem]
e[RelPref]-la[PossConc]-bo[PronStem]
e-la-bo

Relatives from a connective are segmented as follows;

endinayo (that I have)

a[RelPref]-ndi[SC]-nayo[PosStem]

e[RelPref]-ndi[SC]- nay[PosRoot] -o[PosTerm]

e-ndi-nay-o

5.3.3.3 Enumerative (Esobalo)

Enumeratives are considered qualificatives that usually follow the substantive which they qualify (Louw et al., 1984:89; 1989:xxxvi; Pahl, 1982:45). But, the same *Greater Dictionary of Xhosa* lists the stems **-mbi (other)** and **-phi? (where)** as both qualificative and pronouns in Addendum 15 (Pahl et al., 1989:702).

Louw et al. (1984) add two more stems in the list of enumerative stems to the two stems listed above, i.e. **-nye** and **-ni?**. These stems are adequately covered in section 5.3.2.4: *Differentiative Pronoun*. To form the enumerative stems **-nye** and **-ni?** One prefixes the basic class prefix to the stem.

Enumeratives are surface segmented as follows;

ezinye (that are one)

ezi[AC]-nye[EnumStem]

e[QualPref]-zi[SC]-ny[EnumRoot] -e[EnumTerm]

e-zi-ny-e

omni (that are one)

om[AC]-ni[EnumStem]

o[QualPref]-m[SC]-n[EnumRoot] -i[EnumRoot]

o-m-n-i

5.3.3.4 Possessive (Isimnini)

The possessive is a qualificative because it qualifies either a noun or a pronoun in terms of possession (Louw et al., 1984:99; 1989:xxxvi; Pahl, 1982:45).

The possessive modifies a noun with another noun using a possessive concord. It is the possessor that is prefixed with the possessive concord. The possessive concord is formulated by adding the possessive concordial morpheme **-a-** to a class concord, e.g. *u-* + *-a-* > *wa*.

Table 5-9: List of Possessive Concorde (Louw et al., 1984:100)

Class	Singular	Class 2 and 2a	Plural
Class 1 and 1a	<i>wa-</i>	Class 2 and 2a	<i>ba-</i>
3	<i>wa-</i>	4	<i>ya-</i>
5	<i>la-</i>	6	<i>a-</i>
7	<i>sa</i>	8	<i>za-</i>
9	<i>ya-</i>	10	<i>za-</i>
11	<i>lwa-</i>	10	<i>za-</i>
14	<i>ba-</i>		
15 and 17	<i>kwa-</i>		

Possessives from nouns are segmented as follows;

<i>wenkosi</i> (that are one)
wa[PC]-i[BPre]-nkosi[NStem]
we[PC]-nkos[NounRoot]-i[NTerm]
we-nkos-i

Possessives formed from pronouns are unique because the absolute pronoun loses its suffix, –*na*, when it converts to a possessive pronominal stem. The possessive pronominal stems are listed in the table Table 5-10. An example of a pronominal possessive is shown below.

<i>yakhe</i> (that are one)
ya[PC]-khe[PosStem]
ya[PC]-kh[PosRoot]-i[PosTerm]
ya-kh-e

Table 5-10: Possessive Pronominal Stems (Pahl et al., 1989:690)

Person/Class	Singular	Plural
1st Person	<i>-m</i>	<i>-ithu</i>
2nd person	<i>-kho</i>	<i>-inu</i>
3rd person		
Class 1 and 1a	<i>-khe</i>	Class 2 and 2a <i>-bo</i>
3	<i>-wo</i>	4 <i>-yo</i>
5	<i>-lo</i>	6 <i>-wo</i>
7	<i>-so</i>	8 <i>-zo</i>
9	<i>-yo</i>	10 <i>-zo</i>
11	<i>-lo</i>	10 <i>-zo</i>
14	<i>-bo</i>	
15 and 17	<i>-ko</i>	

From the possessive prefixes and possessive pronominal stems, a list of pronominal possessives is generated as in Addendum 9: Pronominal Possessives of the *Greater Dictionary of Xhosa: Volume 3* (Pahl et al., 1989:691).

5.3.4 Predicate (Isivisa)

The predicate is the word that carries the central meaning of a sentence. In isiXhosa it “can be the statement, the relation or the declaration” (Louw et al., 1984:111)., There are two types of predicates in isiXhosa, i.e. verbs and copulatives.

They are always used relative to a substantive, either a subject or an object. This relation is reflected in the concord, the subject concord if related to the subject or object concord if related to the object. The concords are listed in Table 5-3 and Table 5-4.

Because each predicate category has a complex structure they should be handled separately.

5.3.4.1 Verbs

IsiXhosa verbs contain morphemes for person, number, class, mood, tense and form (McLaren, 1948:80), e.g. *ndi-balek-is-iwe* (I was rushed), *be-si-ngxam-is-ana* (we rushed each other), *a-ka-buy-a* (he/she didn’t return), *ba-ku-buy-isw-a* (they will be returned).

Verb extension, conjugation, change in verb form, mood, aspect and tense can cause verb inflection.

5.3.4.1.1 Verb Extensions

Verbs may be extended to show a relationship between the subject, the object and the intensity of the action. These extensions are: Applied, Neuter, Causative, Connective, Intensive, Passive and Reduplication (Louw et al., 1984:122; Pahl, 1982:81). Table 5-11 shows the segmentation manners for each extension type

Table 5-11: Verb Extension examples

Basic verb stem	<u><i>bamba</i> (hold)</u> <u>bamb[VRoot]-a[VT]</u> <u>bamb-a</u>
With applied extension	<u><i>bambela</i> (hold for)</u> <u>bamb[VRoot]-el[ApplExt]-a[VT]</u> <u>bamb-el-a</u>
With Neuter extension	<u><i>bambeka</i> (able to hold to)</u>

	<u>bamb[VRoot]-ek[NeutExt]-a[VT]</u> <u>bamb-ek-a</u>
With Causative extension	<u>bambisa (hold with)</u> <u>bamb[VRoot]-is[CausExt]-a[VT]</u> <u>bamb-is-a</u>
With Connective/reciprocal extension	<u>bambana (hold each other)</u> <u>bamb[VRoot]-an[RecipExt]-a[VT]</u> <u>bamb-an-a</u>
With Intensive extension	<u>bambisisa (hold tight)</u> <u>bamb[VRoot]-isis[IntensExt]-a[VT]</u> <u>bamb-isis-a</u>
With Passive extension	<u>banjwa (able to hold to)</u> <u>bamb[VRoot]-iw[PassExt]-a[VT]</u> <u>banj[VRoot]-w[PassExt]-a[VT]</u> <u>banj-w-a</u>
With Reduplicated stems extension	<u>bambabamba (hold many times)</u> <u>bamba[VStem]-bamba[VStem]</u> <u>bamba-bamba</u>

5.3.4.1.2 Conjugation

Verb negation in isiXhosa is expressed using a bound morpheme, e.g. **aka-hamb-ang-a** (he/she did not go). This is referred to as conjugation (Louw et al., 1984:132). The positive conjugation of the above example would be **u-hamb-ile** (he/she went). In verb conjugation the verb root is persistent and the word should be segmented as.

<u>uhambile (he/she left)</u>
<u>u[SC]-hamb[VRoot]-ile[VTPerf]</u>
<u>u[SC]-hamb[VRoot]-il[VPerf]-e[VT]</u>
<u>u-hamb-il-e</u>

<u>akahambanga (he/she left)</u>
<u>a[NegPre]-ka[SCNeg]--hamb[VRoot]-anga[NegCirc]</u>
<u>a[NegPre]-ka[SCNeg]--hamb[VRoot]-ang[NegCirc]-e[VT]</u>
<u>u-ka-ham-ang-a</u>

5.3.4.1.3 Forms

There are two forms of verbs in isiXhosa, i.e. the potential and temporal forms (Louw et al., 1984:133). Whilst the potential form indicates an ability, e.g. **ndi-nga-hamba** (I **can** go), the

temporal form indicates “when the action of the verb takes place”, e.g. *sa-ku-hamba* (**when** we go).

<i>ndingahamba</i> (I can go)
ndi[SC]-nga[Pot]-hamb[VRoot]-a[VT]
ndi-nga-hamb-a

<i>sakuhamba</i> (when we go)
sa[PronPref]-ku[SC]-hamb[VRoot]-a[VT]
sa-ku-hamb-a

Negation for both forms can be expressed as below, e.g. *a-ndi-nge-hamb-i* (I **cannot** go), *sa-ku-nga-hamb-i* (**when** we **didn't** go).

<i>ndingehambi</i> (when I do not go)
ndi[SC]-nga[Pot]-hamb[VRoot]-i[VT]
ndi[SC]-nge[Pot]-hamb[VRoot]-i[VT]
ndi-nge-hamb-i

<i>sakungahambi</i> (when we do not go)
sa[PronPref]-ku[SC]-nga[NegPre]-hamb[VRoot]-i[VT]
sa-ku-nga-hamb-i

5.3.4.1.4 Moods

There are five verb moods in isiXhosa. These express how the speaker views the reality expressed by the predicate (Louw et al., 1984:133). Although it is semantic, it is expressed in the morphology of the word. The infinite moods are the **infinitive** and **imperative** mood, whilst the **indicative**, **participial**, and **subjunctive** moods are finite moods.

Infinite moods do not have a subject concord and do not relate to a subject of a sentence e.g. the infinitive verb, *ukubona* [to see], does not indicate tense and can act as both a noun and a verb. Imperative verbs, on the other hand, do indicate tense but only the present tense, e.g. *bona* [see]. These can be segmented as follows

<i>ukubona</i> (to see)
uku[SC]-bona[VStem]
u[NPrePre]-ku[BPre]-bon[VRoot]-a[VT]
u-ku-bon-a

<i>bona</i> (see)
bona[VStem]
bon[VRoot]-a[VT]
bon-a

Finite moods use subject concords. The indicative mood makes a statement, e.g. *uyahamba* [he/she goes]. The participial mood stipulates the circumstances under which the predicate takes place, e.g. *uhamba ecela* [he/she goes begging]. The subjunctive mood displays a made-up reality, e.g. *ukhwazela ukuba avakale* (he/she shouts to be heard). The segmentation of these is shown below.

<i>uyahamba</i> (he/she goes)
u[SC]-ya[LongPres]-hamba[VStem]
u[SC]-ya[LongPres]-hamb[VRoot]-a[VT]
u-ya-hamb-a

<i>uhamba</i> (he/she goes)	<i>ecela</i> (begging)
u[SC]-hamba[VStem]	e[SC]-cela[VStem]
u[SC]-hamb[VRoot]-a[VT]	e[SC]-cel[VRoot]-a[VT]
u-ya-hamb-a	e-cel-a

<i>uhambela</i> (he/she walks)	<i>ukuba</i> (so that)	<i>abonakale</i> (he/she can be noticed)
u[SC]-hamb[VRoot]-el[ApplExt]-a[VT]	ukuba[Conj]	a[SCSubj]-bon[VRoot]-akal[NeutExt]-e[VTSubj]
u-hamb-el-a	ukuba	a-bon-akal-e

5.3.4.1.5 Aspect

Aspect qualifies tense indicating whether an action is complete or not, continuous or not or progressive or not (Louw et al., 1984:135). This translates into the **progressive** and **exclusive** aspects. *Usatya* (he/she is still eating) is an example of a progressive verb and *seletyile* (he/she has eaten) an example of an exclusive verb.

<i>usatya</i> (he/she is still eating)
u[SC]-sa[ProgPre]-tya[VStem]
u[SC]-sa[ProgPre]-ty[VRoot]-a[VT]
u-sa-ty-a

***seletyile* (he/she has eaten)**

se[SC] -le[AuxVStem]-ty[VRoot]-ile[VTPerf]

se-le-ty-ile

5.3.4.1.6 Tense

There are six verb tenses in isiXhosa and they are the **present**, **future**, **perfect**, the **remote past**, the **continuous past** and the **contingent** tense (Louw et al., 1984:134).

The positive conjugation of the verb ends in the terminative morpheme *–a*, e.g. *u-ya-bon-a* (s/he is seeing), except for the perfect tense which uses the suffix *–ile*, e.g. *u-bon-ile* (he saw). Tenses can also take a negative conjugation which terminates with *–i*. The morphemes *–ka-* and *–nga-* are also used in negation.

Most verbs have two forms per tense, i.e. the short form, e.g. *ndi-bona* (I see), and the long form, *ndi-ya-bona* (I see). To produce the long form in the present tense, one appends the morpheme *–ya-* to the subject concord.

The form of an isiXhosa verb in a particular tense is made of prefixes and suffixes that are defined by the conjugation, form, mood and aspect of the verb. The table below gives examples of isiXhosa verbs for different tenses.

Table 5-12: Examples of different isiXhosa tenses

Tense	Positive example	Negative example
Present tense	<i>ndi-bon-a</i> (I see), <i>Ndi-ya-bon-a</i> (I see)	<i>a-ndi-bon-i</i> (I don't see)
Future tense	<i>ndiza ku-bon-a</i> (I will see)	<i>andizi ku-bon-a</i> (I will not see)
Perfect tense	<i>ndi-bon-ile</i> (I saw)	<i>a-ndi-bon-ang-a</i> (I did not see)
Remote past tense	<i>wa-a-bona</i> (he/she saw)	<i>a-ka-bon-ang-a lakazange wa-bona</i> (he/she did not see),
Continuous past tense	<i>ba-be-bona</i> (they used to see)	<i>ba-be-nga-bon-i</i> (they did not see)
Contingent tense	<i>bendiza ku-bona</i> (I was going to see)	<i>Bendingazi ku-bona</i> (I was not going to see)

The above table shows a few of the isiXhosa verb tenses. One would have seen these forms earlier in this chapter, so their segmentation will not be redone. The table is not exhaustive of all the different combinations of isiXhosa verb tenses, moods, forms etc. However, it gives a good indication of the segments that constitute words across the different tenses.

In Table 5-12 it is seen that some verbs precede other verbs. These verbs are referred to as auxiliary verbs and should be segmented the same as other verbs.

5.3.4.1.7 Conclusions on Verbs

As demonstrated above, isiXhosa verbs are intricate but are consistent in their segmentation. The common morpheme in a verb paradigm is the verb root with further morphemes attached on both sides for mood, aspect, tense, form and conjugation.

5.3.4.2 Copulatives

Copulatives are non-verbal predicates and specify the nature of a substantive, e.g. *ngumthi* (it is a tree). They are derived from nouns, absolute pronouns, qualificatives, descriptives and conjunctions (Louw et al., 1984:216). They can also stand as a complete sentence (Mncube, n.d:59). Even though they can be produced from quantitatives as well, such copulatives appear as word phrases, instead of individual words (Louw et al., 1984:227). Such will not be expounded on here as a phrase can be segmented according to earlier specifications.

Copulatives are predicates and as such have positive and negative conjugation. They also have a personal and an impersonal form.

Copulatives derived from substantives are called identificative copulatives (Louw et al., 1984:216) and identify the substantive in terms of type, status etc.

This section will present only noun- and pronoun-derived copulatives because pronoun derived copulatives are used in the negation of other copulatives. Derivation of other copulatives follows a similar pattern.

5.3.4.2.1 Pronoun Derived Copulatives

Copulatives that are derived from absolute pronouns are formed by prefixing the copula to the absolute pronoun. The copula connects a subject with any predicate other than a verb (McLaren, 1948:44). Different noun class pronouns use different copulas as shown in Table 5-13.

The absolute pronoun stem **-na** is dropped.

To negate a copulative one prefixes the negation prefix *asi-* or *ayi*, depending on the dialect of isiXhosa spoken, to the positive copulative (Pahl, 1982:167).

Table 5-13: List of Copula (Louw et al., 1984:220)

Person/Class	Singular		Plural
1st Person	<i>ndi-</i>		<i>si-</i>
2nd person	<i>ngu-</i>		<i>ni-</i>
3rd person			
Class 1 and 1a	<i>ngu-</i>	Class 2 and 2a	<i>nga-</i>
3	<i>ngu-</i>	4	<i>yi-</i>
5	<i>li-</i>	6	<i>nga-</i>
7	<i>si-</i>	8	<i>zi-</i>
9	<i>yi-</i>	10	<i>zi-</i>
11	<i>lu-</i>		
14	<i>bu-</i>		
15 and 17	<i>ku-</i>		

An example of segmenting a pronoun derived copulative is shown below:

<i>nguwo</i> (it is it)	<i>asinguwo</i> (it is not it)
ngu[CopPre]-wo[PronStem]	a[NegPre]-si[BPre]-ngu[CopPre]-wo[PronStem]
ngu-wo	a-si-ngu-wo

A list of copulatives that are derived from pronouns are listed in Pahl (1982:167) and Louw et al (1984:220).

Personal copulative pronouns are derived by prefixing the copulative pronoun with the subject concord of the appropriate person Impersonal copulatives. The latter are derived by prefixing a subject concord (Table 5-3) to the copulative pronoun. The absolute pronoun stem *-na* is again dropped.

<i>unguwo</i> (s/he is it)	<i>akanguwo</i> (s/he is not it)
u[SC]-ngu[CopPre]-wo[PronStem]	a[NegPre]-ka[ExclNeg]-ngu[CopPre]-wo[PronStem]
u-ngu-wo	a-ka-ngu-wo

A list of impersonal copulatives that are derived from absolute pronouns is listed in Louw et al. (1984:222).

5.3.4.2.2 Noun Derived Copulatives

To form copulatives from nouns, one prefixes the noun stem with a compound prefix composed of the the Noun-Derived Copulative Prefix (Table 5-14), the copula (cf. Table 5-13 on page 112), and the noun prefix (cf. Table 5-2 on page 96) (Louw et al., 1984:217; Pahl, 1982:167).

Table 5-14: Noun Derived Copulative Prefixes (Louw et al., 1984:220)

Person	Singular		Plural
1st Person	<i>ndi-</i>		<i>si-</i>
2nd person	<i>ngu-</i>		<i>ni-</i>
3rd person			
Class 1	<i>ngu-</i>	Class 2	<i>nga-</i>
1a	<i>ngu-</i>	2a	<i>ngoo-</i>
3	<i>ngu-</i>	4	<i>yi-</i>
5	<i>li-</i>	6	<i>nga-</i>
7	<i>si-</i>	8	<i>zi-</i>
9	<i>yi-</i>	10	<i>zi-</i>
11	<i>lu-</i>	10	<i>zi-</i>
14	<i>bu-</i>		
15 and 17	<i>ku-</i>		

Personal copulatives derived from a noun are formed by prefixing the noun-derived copulative with the subject concord. To negate one prefixes the noun-derived copulative *asi-* e.g.:

***ndingumntu* (I am human)**

ndi[SC]-*ngu*[CopPre]-*m(u)*[BPre]-*nt*[NRoot]- *u*[NTerm]

ndi[SC]-*ngu*[CopPre]-*m*[BPre]-*nt*[NRoot]- *u*[NTerm]

ndi-ngu-m-nt-u

***ngumntu* (it is human)**

ngu[CopPre]-*m(u)*[BPre]-*nt*[NRoot]- *u*[NTerm]

ngu[CopPre]-*m*[BPre]-*nt*[NRoot]- *u*[NTerm]

ngu-m-nt-u

***andingomntu* (I am no human)**

a[NegPre]-*ndi*[SC]-*ngu*[CopPre]-*m(u)*[BPre]-*nt*[NRoot]- *u*[NTerm]

a[NegPre]-*ndi*[SC]-*ngo*[CopPre]-*m*[BPre]-*nt*[NRoot]- *u*[NTerm]

a-ndi-ngo-m-nt-u

***asingomntu* (it is no human)**

a[NegPre]-*si*[ExclNeg]-*ngu*[CopPre]-*m(u)*[BPre]-*nt*[NRoot]- *u*[NTerm]

a[NegPre]-*si*[ExclNeg]-*ngo*[CopPre]-*m*[BPre]-*nt*[NRoot]- *u*[NTerm]

a-si-ngo-m-nt-u

5.3.5 Descriptive

According to Louw et al. (1984:24), a descriptive is a word that describes a predicate or other descriptive in terms of manner, time and place (locatives).

Descriptives of manner are derived from adjective stems, relative stems, nouns and pronouns by prefixing them with the descriptive prefixes, *ka-*, *kaku-*, *na-*, *kuna-*, *nga-*, *njenga-*, and *nganga-*. (Louw et al., 1984:254). Descriptives of manner should be segmented by identifying the descriptive prefix [DesPre] e.g.

<i>kakuhle</i> (beautifully)
kaku[DesPre]-hle[AdjStem]
kaku[DesPre]-hl[AdjRoot]-e[AdjTerm]
kaku-hl-e

Descriptives of time are derived by prefixing the time noun stem with *e-*. These descriptives should also be segmented to show the descriptive prefix.

<i>emini</i> (during the day)
e[LocPre]-mini[NounStem]
e[LocPre]-min[NounRoot]-i[NTerm]
e-min-i

5.3.6 Adverbs

The isiXhosa adverb expresses a relation of place, time or manner of a verb or another adverb (Mncube, n.d:51).

IsiXhosa contains a few simple adverbs because of its use of nouns in the locative case, e.g. ***e-bu-suku*** (at night), ***ku-ba-ntu*** (to the people), ***e-mthi-ini*** (to/at the tree) or by preceding nouns with the preposition *nga-* e.g. *iya nga-sekunene* (go left). To express manner, adverbs are formed by prefixing *ka-* to an adjective stem, e.g. ***ka-de*** (for long), (McLaren, 1948:142). To describe time, one uses the prefixes *nge-*, *ngo-* and *ku-* to form adverbs, e.g. ***nge-Cawa*** (on Sunday), ***ngo-Mvulo*** (on Monday), ***ngo-mso*** (tomorrow), ***ku-sasa*** (in the morning), ***ku-qala*** (first).

<i>emthini</i> (at the tree)
e[AdvPre]-m[BPre]-thi[NStem]-ini[LocSuf]
e[AdvPre]-m[BPre]-th[NRoot]-in[LocSuf]-i[NTerm]
e-m-th-in-i

Another set of adverbs is a closed set called simple adverbs, e.g. *phandle* (outside) and *phesheya* (across the sea). These should be segmented with root and stem. Prefixal adverbs should be segmented to show the adverbial prefix the stem is bound to, e.g. *ngoMvulo* (on Monday) > *ngo-Mvulo*. An example is shown below

pshesheya (across the sea)

pshesheya[NStem]

pshesheya[NRoot]- a[NTerm]

pshesheya-a

5.3.7 Conjunctions

Conjunctions are words introducing a sentence or linking two clauses (Louw et al., 1984:262).

IsiXhosa has few primitive conjunctions. Most isiXhosa conjunctions are either verbal forms (e.g. *ukuze* < *ukuze* (then), *ndaye* < *ukuya* (and I), *ngokungathi* < *ukuthi* (like), nouns (e.g. *xa* < *ilixa* (when), *mhla* < *umhla* (the day when), and pronouns (e.g. *kaloku* (by the way) < *oku* (this), *okoko* < *oko* (that), especially the pronoun *oko* (McLaren, 1948:149; Pahl, 1982:205). Some conjunctions are phrases, e.g. *ngoko ke* (therefore) and *ke kaloku* (when).

Even though there seems to be productivity in conjunctions, that productivity is exhausted; conjunctions are a closed set, albeit a large set. To preserve similarity with other word categories, conjunctions will be segmented according to their source category or at least to a root and a terminal vowel.

5.3.8 Interjections

The last word category to consider are the interjections. Interjections express exclamation and are isolated. There is one class of original interjections, such as *ewe!* (yes), *hayi bo!* (no), and *heke!* (expressing approval). This list is provided by Pahl (1982:207) and could be considered almost as a closed list.

Another class of interjections is derived from other word categories, e.g. nouns, copulatives, conjunctions, adverbs and even short sentences.

For this study, original interjections should not be segmented as they are almost a closed list. Interjections that are derived from other word categories are difficult to identify without looking at the context. As such, the only solution is to segment them to the source category segmentation.

5.3.9 isiXhosa NLP Surface Segments in Summary

To summarise, IsiXhosa nouns influence the prefix structure of many word categories. The nouns themselves are segmented by identifying the different components of the prefix, the noun's root, any extension to the noun and a noun terminal vowel.

The other word categories are derived from prefixing a stem from another category with appropriate category specific concords. These concords are derived from the subject and object concords. The segmentation of such must show the original concords that the new concords are derived from. There are classes of words that cannot be categorised as definitely closed but have stabilised, like original interjections. Such should be segmented to only a root and terminal vowel. Words derived from other words must be segmented to show the segments responsible for the concordial agreement.

Verbs must be segmented to show the different segments responsible for concordial agreement, and all other grammatical variations and the stem must be segmented to a verbal root and a terminal vowel.

Therefore segmentation must show, in surface form, the concordial links to the subject and object as independent segments. For surface segmentation stems, even pronominal stems are segmented to show a root and a terminal vowel. Extension for grammatical type must be clear as well. To summarise all of the above, the simplified heuristics that ensure that all the segment boundaries can be identified is, therefore:

- The first vowel is always a prefix morpheme (pre-prefix or other);
- The last vowel is always a suffix morpheme (terminal vowel);
- Prefix morphemes are complete syllables, except for 'm' which has a silent vowel when followed by a consonant;
- Roots start with a consonant and end in a consonant; and
- Suffix morphemes are reverse syllables, so they start with a vowel and end in a consonant, except for the terminal vowel and for 'w' which has a silent preceding vowel when following a consonant.

The above heuristics are not linguistic but are meant to simplify isiXhosa surface segmentation

5.4 Dependencies among affixes and morphemes

One of the key aspects of probabilistic modelling is to map statistical dependencies between the entities being modelled. This section will briefly deal with that in relation to the affixes in isiXhosa.

5.4.1 Intra-affix dependencies

In isiXhosa, prefixes are influenced by high concordial agreement between words. Because our work is not inter-word but intra-word, this phenomenon will not be used. Importantly the strong sequence structure in both prefix and suffixes suggest intra-affix dependence. In addition affixal

(prefix and suffix) morphemes are closed classes. These two characteristics suggest a strong sequential probabilistic dependence among morphemes in an affix.

5.4.2 Affix-root dependencies

It is understood in linguistics that the category of a word attracts certain morphemes. This implies a dependence on some affix morphemes for a word category and an independence to other affix morphemes for the same word category, in a way, a distinct statistical characteristic that could be exploited or at least tested. One cannot be sure of this effect as some relatives and descriptives are derived from nouns, some nouns are derived from verbs and the roots tend to be maintained across the categories. This implies that some roots appear in many word categories, diluting this strong dependence or independence. Again another reason to test the affix-root dependency.

5.4.3 Circumfixing

Circumfixing could only be identified for negation in this chapter, however, the selective dependence between roots and affixes could mean a level of statistical circumfixing. I use the word “statistical circumfixing” to mean that there might be some selectivity between prefixes and suffixes, although it happens through the root, and it might be statistically useful for the study. Therefore, this is a reason to experiment.

5.4.4 Summary on possible morpheme dependency.

Considering the above, there is generally a strong dependence among the prefix morphemes in a prefix, and among the suffix morphemes in a suffix. This gives us a sequential affix morpheme dependence that we can base our work on.

The dependency between roots and affixes can be argued to exist, but the reuse of roots in many word categories means that this might be diluted and the affixes might be independent of each other and the root.

The dependence between prefixes and suffixes, namely circumfixing, seems very limited but because of the seeming selectivity between roots and affixes it might actually exist through the root and should be tested.

These specified possible dependencies mean that we should test a number of statistical model for isiXhosa, namely:

- A purely morpheme sequential model
- An Affix-root independent model

- An affix-root dependent model, and
- A circumfix-root dependence model.

The list of experiments above, provides a good combination of the dependencies that need to be evaluated to be able to choose an unsupervised surface segmentation model for isiXhosa. The work of converting these models into probabilistic graphical models is done in Chapter 6: *IsiXhosa Language model based segmentation* and the evaluation is done in Chapter 7: *Evaluation*.

5.5 Conclusions

In introducing this chapter, an explanation of what a segmentation in isiXhosa is was given. IsiXhosa words were then classified into words categories so that an appropriate segmentation can be determined.

Thereafter the appropriate surface segmentation for every word category and sub-category is established. This exhaustive approach allows for the identification of all the boundary locations. This work culminates in the establishment of a set of heuristics that covers the surface segmentation of isiXhosa. The surface segmentation does not concur with grammatical rules because it oversegments some stems, however this ensures that all the boundary locations are identified.

An analysis of the dependencies among morphemes in the languages provides a guide to a set of possible statistical models for the language that need to be evaluated.

6. ISIXHOSA LANGUAGE MODEL BASED SEGMENTATION

In this chapter, directed probabilistic graphical techniques are used to model the morphological segmentation of isiXhosa. These use language modelling to estimate the segmentation probability distributions. These distributions, constrained by isiXhosa heuristics established in Chapter 5, are then used to find the most probable isiXhosa word segmentation.

Probabilistic Graphical Modelling (Bishop, 2006; Airoldi, 2007) provides a simple way of modelling the generative behaviour of a stochastic system. It has been used in several morphological segmentation studies (Dreyer & Eisner, 2011; Chen et al., 2014; More & Tsarfaty, 2016; Costa-Jussà & Fonollosa, 2015). Most studies that use language modelling to implement the generative process use non-parametric models with Markov Chain Monte Carlo with Gibbs Sampling techniques (Geman & Geman, 1984) for inference because of the excessive size of the search space. In this study, the use of simplifying heuristics derived from isiXhosa's properties, to reduce the search space, will be tested.

Probabilistic graphical models are a diagrammatic representation of a parametric random process. These models allow for simple visualisation of the structure of the probabilistic model, giving a clear view of the properties of the model - including the conditional dependence properties (Bishop, 2006:360). PGMs come in two forms, directed graphs and undirected graphs. In this document directed graphs are used. These graphs use nodes to indicate the random variables and directed edges to indicate dependence. The direction arrow of an edge originates points at the dependent node, or the vertex of a group of random variables. Shaded nodes indicate observed variables, while unshaded nodes show latent variables that ought to be inferred by an implementation model.

Sect. 2.4.4 discusses Markov Random Fields and gives a summary of Bayesian networks. The majority of segmentation work that uses Bayesian networks has been on supervised text segmentation. In this work, Bayesian Networks are used in an unsupervised manner, and the design of these networks and how they are trained is detailed. The results are described in Sect. 7.2. Four network configurations are tested to determine which network best models isiXhosa words.

A language model is a model that assigns probabilities to sequences of words $p(w_{1..N})$ (Jurafsky & Martin, 2000:36). This is achieved by using the chain rule of probability and assuming that the conditional probabilities are a Markov Chain of order n . Sect. 2.6.9 gave some detail about language models.

In this work, four probabilistic graphical models for modelling isiXhosa word generation are proposed and compared. For training and inference, the models are guided with isiXhosa heuristics to reduce the search space. Because of sparsity, language models require smoothing to compensate for unseen tokens, but there are cases where this smoothing is ineffective. The state-of-the-art in language modelling smoothing is the modified Kneser-Ney (mKN) smoothing technique. In the evaluation in Chapter 7, results from mKN smoothed and non-smoothed models are compared.

6.1 IsiXhosa Probabilistic Models

The isiXhosa Heuristics Maximum Likelihood Segmenter (XHMLS) that is proposed is based on the directed graphical model of isiXhosa and isiXhosa heuristics.

The individual models required that different statistical data be kept. Where there is a sequence of morphemes, the modelling technique has an option of smoothing using the modified Kneser-Ney smoothing (Chen & Goodman, 1998). Information on the four models evaluated is presented below.

In the following models, the symbols are standardised such that, w is a word, s is a word segmentation, r is a word root, c is a circumfix, m is a morpheme, p is a prefix, s is a suffix, p_i is the i -th prefix morpheme, s_j is the j -th suffix morpheme, m_k is the k -th morpheme, I is the number of prefix morphemes, J is the number of suffix morphemes and K is the number of morphemes. Let a sequence $m_{1..k} = m_1, m_2, \dots, m_k$ represent a list of morphemes. To handle sequences and conditional probabilities effectively, an axiomatic convention for vanishing sequences and define a truncation operation is stated.

Axiom 6-1. Vanishing sequences convention. Consider sequences $\vec{x} = x_i, x_{i+1}, \dots, x_{i+n}$ and $\tilde{x} = x_{i-n}, \dots, x_{i-1}, x_i$ where $n, i \in \mathbb{Z}$. For convenience and as a convention in this document, both sequences $\vec{x}$ and $\tilde{x}$ reduce to x_i when $n = 0$, and vanish to an empty list, nothing, when $n < 0$. Furthermore, $p(y|\vec{x})$ and $p(y|\tilde{x})$ reduce to $p(y|x_1)$ when $n = 0$, and to $p(y)$ when $n < 0$.

Definition 6-1. The **truncation operator** $\langle \dots \rangle_{[n]}$ on a sequence $m_{1..k}$ is defined as a function that returns a vanishing sequence of its argument such that

$$\langle m_{1..k} \rangle_{[n]} = \begin{cases} \text{nothing,} & n \leq 0 \\ m_{1..k}, & n \geq k \\ m_{k-n+1}, \dots, m_{k-1}, m_k, & \text{otherwise} \end{cases} \quad (6-1)$$

This representation allows for a compact formulation of probabilistic mass functions.

In this section, four (4) models of varying complexity are presented. A modified graphical representation which differentiates between generation and statistical dependence is used. Generation is shown using dashed lines and statistical dependence using solid lines.

6.1.1 Morpheme Sequence model (MS)

The simplest model is the Morpheme Sequence (MS) model. The model proposes that an isiXhosa word is made up of a root and affixes. This model considers the word root to be one of the morphemes and models all the word morphemes to have a Markov sequential dependence of some order n .

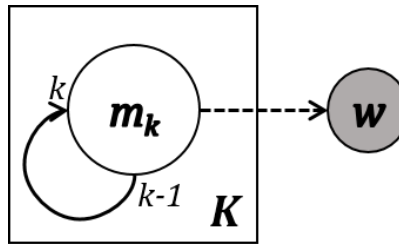


Figure 6-1: Probabilistic graphical model of the Morpheme Sequence model

In Figure 6-1 the plate notation of PGM is used. Here, a recursion plate form, something not seen in PGM, is introduced. The plate notation means that whatever is inside the block is repeated by the number of times specified on the right bottom corner of the block, in this case, K times and the m_k node is dependent on the m_{k-1} node. The probability of the word w when modelled by the segmentation $\mathcal{s}$, where the Markov order of the dependencies is n , is estimated to be

$$P(w|\mathcal{s}) = \prod_{k=1}^K P(m_k | \langle m_{1..k-1} \rangle_{[n]}). \quad (6-2)$$

The next model is also a simple model for isiXhosa.

6.1.2 Independent Affixes (IA)

The Independent Affixes (IA) model is one of the simplest models. It assumes that an isiXhosa word is made up of a root and affixes, that all three are independent of each other and that the affix morphemes have a Markov dependence within the affix. This model implements the affix dependencies proposed in Sect 5.4: *Dependencies among affixes*.

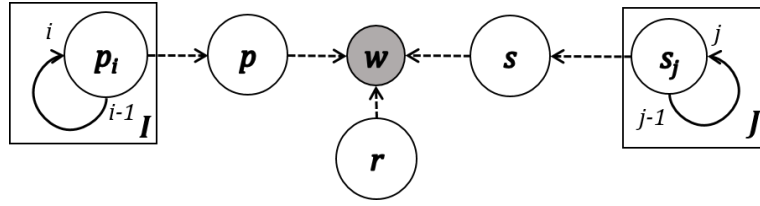


Figure 6-2: Probabilistic graphical model of the Independent Affixes model

Note that a word can have an empty prefix or an empty suffix, or both. The probability model distribution for the segmentation is formulated as follows:

$$P(w|\delta) = P(r) \cdot \prod_{i=1}^I P(p_i | \langle p_{1..i-1} \rangle_{[n]}) \cdot \prod_{j=1}^J P(s_j | \langle s_{1..j-1} \rangle_{[n]}). \quad (6-3)$$

It is important to note that the relationship between p and p_i is that of generation, that p is made up of I p_i 's and the suffix also has an analogous arrangement. The next model is called Affix-Links (AL) and introduces some complexity to the structure.

6.1.3 Affix-Links (AL)

The Affix-Links (AL) model proposes that a word is made up of a root and affixes and that the dependence between the root and the affixes is through the first prefix morpheme and the last suffix morpheme. If there are I prefix morphemes with the first morpheme being p_1 and there are J suffix morphemes with the last morpheme being s_J ; then the prefix attaches to the root through the first prefix morpheme, p_1 ; the suffix attaches to the root through the last suffix morpheme, s_J ; and the rest of the affix morphemes attach to these linking morphemes, and the root, in a Markov Sequence of order n . Figure 6-3 shows the Probabilistic Graphical model for Affix Links.

Note that the boundary of the plate model has been modified to only apply to the affix morpheme sequence dependence. In this model, variables for the generated prefix and suffix are not shown as they don't occur in the dependence structure.

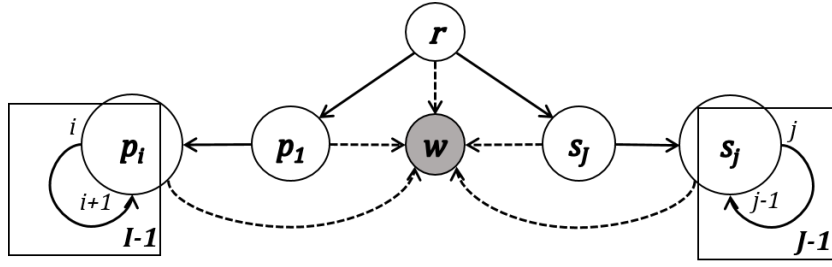


Figure 6-3: Probabilistic graphical model of the Affix-Links model

The probability formulation of the word under a segmentation is therefore

$$P(w|s) = P(r) \cdot \prod_{i=1}^I P(p_i | \langle r, p_{1..i-1} \rangle_{[n]}) \cdot \prod_{j=1}^J P(s_j | \langle r, s_{J..j+1} \rangle_{[n]}). \quad (6-4)$$

It is important to note that the Markov sequence dependency for the suffix is reversed.

6.1.4 Circumfix Link (CL)

The Circumfix Link model proposes that in generating an isiXhosa word a root is chosen and inflected with a circumfix, which is composed of a prefix and a suffix. In addition, the circumfix used is dependent of the root.

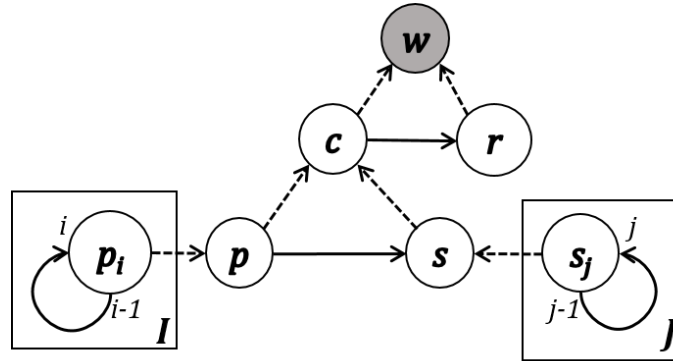


Figure 6-4: Probabilistic graphical model of the Circumfix-Link model

A circumfix can have an empty prefix, thereby presenting as a suffix only, and can have an empty suffix, thereby presenting as a prefix only. The assumption is also made that there is a Markov dependence of order n between the affix morphemes within an affix.

The model is shown in Figure 6-4. The probability formulation of a word segmentation using a circumfix link is shown in Equations (6-5)-(6-8).

$$P(w|\mathcal{s}) = P(r|c) \cdot P(s|p) \cdot P(p) \quad (6-5)$$

$$P(c) = \prod_{k=1, \dots, K}^{m_{1..K}=p_{1..I} \oplus s_{1..J}} P(m_k | \langle m_{1..k-1} \rangle_{[n]}), \quad (6-6)$$

$$P(p) = \prod_{i=1}^I P(p_i | \langle p_{1..i-1} \rangle_{[n]}), \quad (6-7)$$

$$P(s) = \prod_{j=1}^J P(s_j | \langle s_{1..j-1} \rangle_{[n]}). \quad (6-8)$$

Note that $\oplus$ denotes the concatenation of two lists.

6.2 Training and Inference

During training, each training word is split into all possible combinations of the heuristics as per Sect. 5.3.9: *isiXhosa NLP Surface Segments in Summary*. Each possible segmentation has a list of prefix morphemes, a list of suffix morphemes and a root. The root and the two lists are used in the language model. Depending on the model, they can be concatenated into a single list and n-gram statistics generated from that, as for the Morpheme Sequence model, or the model can use the individual lists as in the Independent Affixes model. For the Affix Links model, the prefix and suffix lists are augmented with the root, and n-grams generated from that. For the Circumfix Links model, the model is changed such that one list is generated such that the root is the last in the sequence.

A normal unsmoothed conditional distribution language model may be built from the frequency counts, or a modified Kneser-Ney smoothed model.

Inference chooses the segmentation that maximizes the probability of the word as

$$\mathcal{s}^* = \arg \max_{\mathcal{s} \in Segs(w)} P(w|\mathcal{s}), \quad (6-9)$$

where $Segs(w)$ is all the possible segmentations of w that the heuristics can generate.

6.3 Conclusions

In this chapter, four possible models that could capture the generative nature of isiXhosa are presented: the Morpheme Sequence, Independent Affixes, Affix-Links and Circumfix Link models.

The four language model implementations are explained and an overview of the inferences is given.

7. EVALUATION

This chapter details how the segmentation methods for isiXhosa, developed in this research, were experimentally evaluated and presents the results of these experiments.

There are three (3) sets of evaluations that were conducted for this study. Two of these directly measure the performance of the segmenters and the third determines the complexity-theoretic predictability of isiXhosa sequences. The first experiment evaluates the performance of the branching entropy based segmenter. The second experiment tests the predictability of isiXhosa words using information theoretic means. This evaluation reveals the reason for the performance of the branching entropy segmenter. The third evaluation test determines the performance of the heuristic guided language model segmenter.

This chapter provides an overview of the experimental setup - outlining the rigorous processes followed, the experimental procedures conducted and the detailed comparisons performed.

The full results of the three experiments are provided and evaluated by comparison with the Morfessor-Baseline in terms of accuracy and F1 Score.

7.1 Experimental Setup

Rigorous attention was given to the setup of the experiments to ensure that the validation was justified. Therefore, the description starts by detailing the sources of data used. Thereafter, how the data is used in development, validation and consequent evaluation is specified.

Sect 2.1 *Segmentation Metrics* described how the metrics measured for segmentation are the segmentation boundary identification accuracy and the F1 Score.

This research used three stages in the generation of a verified machine learning system. The first stage is the development and implementation of the system algorithms. The data used in this stage is meant to ensure that the segmenters are set up and work as intended. The next stage is that of validation. The validation stage is normally used for tuning the system's performance and choosing a configuration that gives the best performance. However, in this study, identifying the best configuration and the reason for that configuration is treated as part of the system development stage. In this case, the validation stage was modified to ensure that the system can handle the amount of training data required and to generate test results. The third stage is that of evaluation, which is the ultimate execution of the experiment for system verification.

To ensure that a machine learning system is not built with bias towards the data, the data used for development must be independent of that used for validation, which in turn must be independent of that used for evaluation. Therefore, there is a need for three separate datasets for each stage, i.e. a development dataset, a validation dataset and an evaluation dataset.

In addition to the above, each of the three datasets must be split further into a training dataset and a testing dataset, such that the development of the system does not contain performance biases. The training set is used to train the segmenter and the testing set is used to measure the performance of the segmenter.

7.1.1 Data Sources

In this work, two types of corpora are needed, a corpus of unsegmented words for training the segmenters and a corpus of word-segmentation pairs for testing. In this study two different corpora were used, one for training and the other for testing. The use of two independent corpora reduces the chances of overlap between the two, ensuring rigour in the study. The training corpus was compiled as part of this study, whilst the testing corpus was not. This makes the two corpora independent.

7.1.1.1 Training Corpus

Because this study uses unsupervised training, it needs to use a large training set of raw isiXhosa words. At the time of choosing an isiXhosa training corpus, there were not enough available, so one had to be assembled. To get to a large enough set we combined raw isiXhosa words from three sources to achieve a training corpus of 1.425 million tokens.

- the isiXhosa version of the South African Constitution (South African Parliament, 1996),
- raw isiXhosa text that was scraped from internet sources, and
- the IsiXhosa Genre Classification Corpus (Snyman et al., 2012).

We refer to this corpus in the rest of this document as the training corpus. This corpus has 157 095 unique words. The word with the highest frequency is *ukuba* and it is seen 14871 in the corpus, about 1% of the corpus.

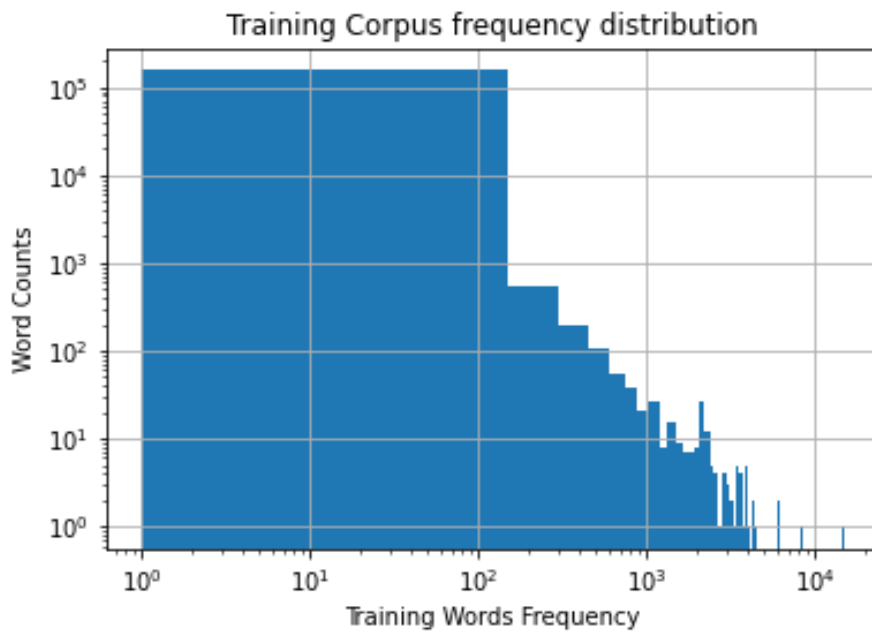


Figure 7-1: Training Corpus word frequency distribution

It is important to note that recently there are bigger raw isiXhosa text corpora available, that could be used, but these corpora were not available at the time of the development of this work.

7.1.1.2 Testing Corpus

For testing, one needs a segmentation annotated corpus, specifically one that consists of word-segmentation pairs, e.g. *siyabaleka* → *si-ya-balek-a*. At the time of choosing a testing corpus the largest segmentation corpus available was the NCHLT IsiXhosa Text Corpus (Eiselen & Puttkammer, 2014), so this was chosen as the base for the testing corpus. This corpus has 44 609 tokens including punctuations, numbers and sentence separation lines. The corpus is rated at an accuracy of 84.66%.

Some cleanup was required for this corpus. The corpus had the following types of undesirable entries. The list below shows the type of undesirable tokens and their quantities:

Table 7-1: Quantities of undesirable NCHLT corpus entries

Type	Quantity
Numbers, Punctuations, Sentence separation lines	8 073
Non-surface segmentations.	18 962
Unsegmented words. These were words that were not recognised by the rule-based segmenter therefore unsegmented.	4 133
Total	31 168

Excluding the above types of entries resulted in a testing corpus of 13 441 tokens. For a testing corpus of a scarce resource language, we consider 13 441 tokens a decent size. This set has 4 102 unique words. Below is the histogram of the frequency distribution of the test set words.

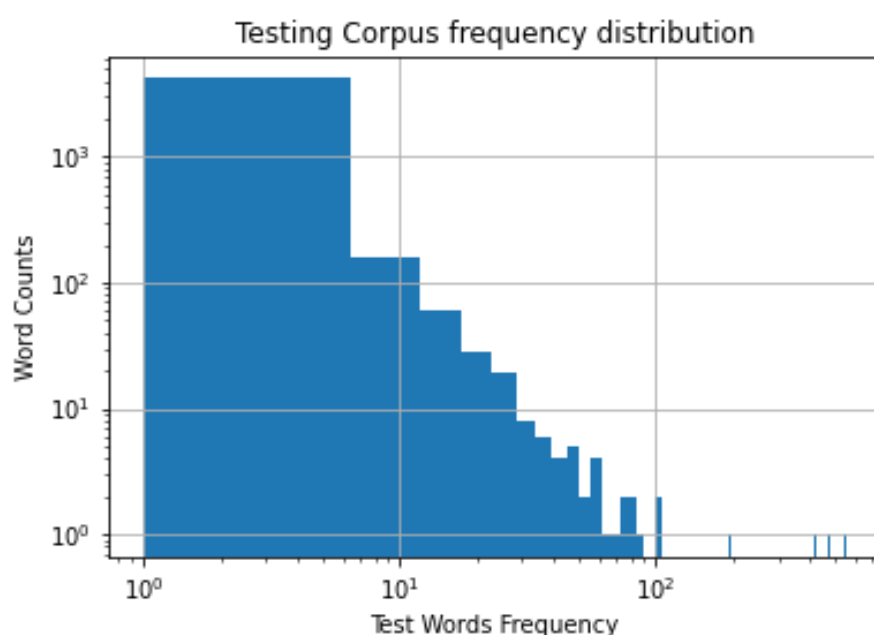


Figure 7-2: Testing Corpus word frequency distribution

The word with the highest frequency is *okanye*, and it is seen 548 times, 4.07% of the corpus.

There will always be an overlap of words between the training and testing corpora. Overall 3 483 of the unique testing words appear in the training corpus. This leaves 519 unique out of vocabulary words, 12.7% of the unique words in the testing corpus. We consider this a high out of vocabulary unique word percentage. This however accounts for 581 of the 13341 tokens, meaning that a great majority of these OoV words have a frequency count of 1. This is also a 4.35% token OoV rate which is comparable to other unsupervised segmentation studies.

7.1.1.3 Conclusion on the corpora

The choice of corpora for this study ensured that we cancel the training – test data bias as the two corpora were generated from two independent studies. That said, we still see a token in vocabulary rate of 95.7% and a unique word in vocabulary rate of 87.3%.

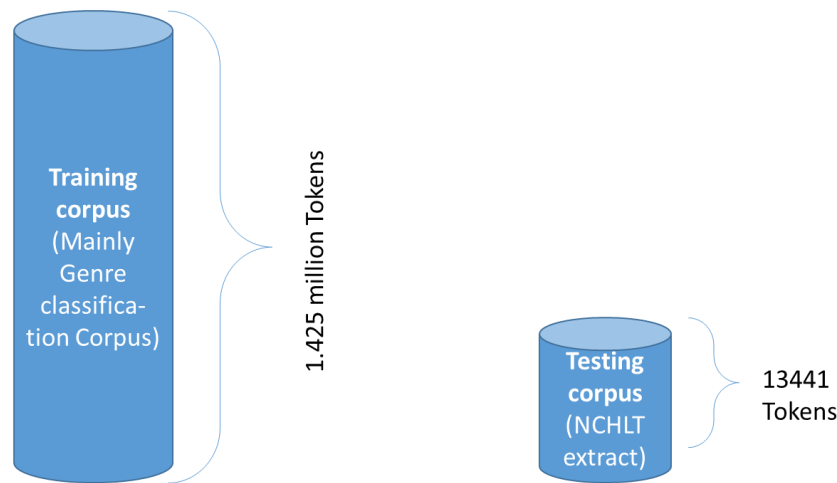


Figure 7-3: Graphical representation of Source corpora

The isiXhosa Genre Classification Corpus and the NCHLT IsiXhosa Text Corpus were accessed from the South African Centre for Digital Language Resources (SADiLaR) (<https://www.sadilar.org>), which provides a central source of data and resources for South African natural language processing work.

7.1.2 Data Splits

The first split to consider is the differentiation between development, validation and evaluation data set. For each of these, one needs a training corpus and a testing corpus.

After that, *N*-Fold validation, as it is applied in this study, is explained. Lastly, details of the data split done during evaluation are given.

7.1.2.1 Development, Validation and Evaluation Corpora

Ordinarily, the corpora are split into three pieces, i.e. development, validation and evaluation datasets, but because the training corpus and the testing corpus are completely separate (having

been created independently through different studies) corpora can be reused safely. The amounts of reuse are small compared to the set sizes and will have little impact on the overall results. This reuse is explained below.

For the development training corpus, the first 10 000 tokens from the 1.425 million token training corpus are used. That amounts to 0.69% of the full training corpus. For development testing, the first 100 tokens from the testing corpus of 13 441 tokens are used, being 0.744% of the testing corpus. This is 75 of the 4102 types, which is a type rate of 1.83%.

For validation, there is a need to simulate part of the evaluation environment. Therefore, the entire training corpus was used for training but testing was performed on only the first 1000 tokens, being 7.5% of the testing corpus. This amounts to 438 unique words of the 4102 words. This amount to reuse of 10.7% of the types. This allowed for the evaluation of the resource usage of the system without significant effect on the results of the evaluation. This is considered acceptable as the training and testing corpora come from two different sources. It is also important to remember that in this study, validation was not used to tune performance or the selection of the best performing configuration, it was only used to ensure that the training process can deal with the size of the training corpus. This design decision of using a testing corpora different from a training corpora allow us to maximise on the use of the corpora. For evaluation, the entire training and testing corpora were used.

Table 7-2: Test Corpora words overlap

Stage	Token count	Token Percent	Types count	Types Percent
Development	100	0.75%	75	1.83%
Validation	1000	7.44%	438	10.7%
Evaluation	13441	100%	4102	100%

For the experiment on the predictability of isiXhosa tests, only the training corpus was used. For development, the first 10 000 tokens from the training corpus were used. Since there is no testing required, validation and evaluation amount to the same thing, leading to only the evaluation run being done, on the entire training corpus.

7.1.2.2 N-Fold Validation for unsupervised systems

N-Fold validation is used in machine learning systems to ensure that the testing set is not the same as the training set. This tests the generalisation ability of the model. In unsupervised modelling, where one trains on large amounts of raw data and the testing corpus is completely

different from the training corpus, the case can be different. However, if testing should be done from the same corpus, then full N -fold validation should be used, except that the segmentations will not be used during training, only on the raw text. That said, for a valid statistical analysis, and to test for generalisation in cases where the training and testing corpora are different, it is still necessary to do several training runs using different training datasets. In this section this is done.

In both cases of N -Fold validation, the training corpus to be used for evaluation is split into N blocks, however, these blocks are used differently. In cases where one corpus is used for training and testing, the N blocks are all used for testing, but not at the same time. Training and testing is done N times. At each time, $N - 1$ of the blocks are used for training and the N -th block is used for testing. In this way, there will be N different training and testing sets.

In the case where the training corpus is different from the testing corpus, $N - 1$ of the N blocks are used at a time for training, and the entire testing data set can be used for testing. In this way, there are only N runs, not $N \times N$ as would have been the case if the testing set was split N times as well.

7.1.2.3 Evaluation sets

To evaluate how the segmenters performed under varying degrees of resource scarcity, the performance of the segmenters under different training data sizes and different n -gram lengths was evaluated. For training data sizes, training set sizes being powers of ten, starting from 100 (i.e. 100, 1000, ... 1 million, and finally a size of 1.5 million) were evaluated. For XBES, training used bigrams, trigrams, 4-grams, odd-numbered n -grams up to 19- and all possible n -grams, the infinite-gram. XHML training used bigrams, trigrams, 4-grams, 5-grams, the 7-gram and the infinite-gram. Validation testing showed that XBES did not have a change after 15-grams, motivating the decision to do up 19- and infinite-grams for coverage. For XHMLS, the performance was the same for 5-grams and above, so 7-grams and infinite-grams were added. The effect of modified Kneser-Ney smoothing was also tested. The training set was split using ten-fold validation, as explained above.

The testing corpus was not split in any way. This means that the models trained with different data sizes were evaluated against the entire testing corpus, even the model trained with 100 words.

7.1.3 Segmenters

This section gives some information on the segmentors designed and implemented as part of this study work. In addition to the benchmark segmenter, three segmentors were developed in this

study, i.e. the random segmenter, the branching entropy segmenter (XBES) and the Heuristic Maximum Likelihood Segmenter (XHMLS).

The random segmenter does not need training as it randomly decides if a location in a word is a boundary with a 50% probability. The other two segmenters are trained from a text file containing a word per line. All the segmenters developed in this study were programmed in the Python programming language (van Rossum, 2007) using Python version 3.7.9.

All the segmenters run as command line tools. The Random Segmenters' command line is;

```
> python.exe RandSeg.py <inputFileName> <outFileName>
```

where <inputFileName> if the fully qualified name of the test file and the <outFileName> is the fully qualified name of the segmented words

The command line usage for XBES is

```
> python.exe XBES.py <command> -input <inputFileName> -model  
    <modelFileName> -output <outFileName> -smooth <smoothing> -  
    algorithm <algorithm> -ngramsize <ngramsize>
```

and the one for XHMLS is

```
> python.exe XHMLS.py <command> -input <inputFileName> -model  
    <modelFileName> -output <outFileName> -smooth <smoothing> -  
    algorithm <algorithm> -ngramsize <ngramsize>
```

where: <inputFileName> and <outFileName> are as in the Random Segmenter. The <outFileName> is now required during training, only the <inputFileName> is needed. The rest of the arguments are

- <command> is either `train` for training or `segment` for segmenting.
- <model> is the fully qualified filename for the model file
- <smoothing> argument turns smoothing on (True) or off (False).
- <algorithm> is the abbreviation of the algorithm detailed in the chapter on branching entropy and heuristic guided probabilistic graphical models.
- <ngram> specifies the ngram size to use

7.1.4 Choice of benchmark unsupervised segmenter

To benchmark the performance of the isiXhosa segmenters presented, some baselines were created. This work required evaluation against an existing benchmark segmenter that is publicly available, and that also has shown robustness in segmenting highly agglutinative languages.

The minimum performance baseline was chosen to be a random segmenter that randomly, with 50% probability, decides if a location in a word is a segment or not.

Morfessor-Baseline (Creutz & Lagus, 2002) was chosen to be the comparative benchmark. This has been used as a benchmark extensively (van Niekerk, 2014; Kotzé & Wolff, 2015; de Pauw & de Schryver, 2008; Méndez-Cruz et al., 2016) and is freely available.

7.1.5 Experiments conducted

All the modes of the two segmenters XBES and XHMLS are evaluated, and the predictability of isiXhosa words using information theoretic predictive metrics is measured. XBES also requires choosing an operator to combine the left- and right-branching entropies, i.e. *minimum* or *sum*. The effect of smoothing on performance is also evaluated. Both segmenters are compared to Morfessor-baseline and a random segmenter.

Training was performed for three segmenters, i.e. XBES, XHMLS, and Morfessor-Baseline, using the training corpus, and tested using the testing corpus. The random segmenter does not require training. XBES, XHMLS and Morfessor-Baseline were trained with different training set sizes. Because Morfessor-Baseline does not support specifying n-gram size, only XBES and XHMLS were trained to different model n-gram lengths.

The segmentation using boundary identification accuracy and F1 Score, where, within a word, the morpheme boundary location is tagged with a 1 and everything else 0, was measured. Performance of the systems on out of vocabulary (OoV) words was also done. Statistical significance between results is measured using the Wilcoxon Signed-Rank test (Demšar, 2006). OoV words and statistical significance are explained in Sects. 2.1.2: *Performance on Out of Vocabulary Words* and 2.1.3: *Hypothesis Testing* respectively.

7.1.6 Conclusions on Experimental Design

In this section, the experimental design is explained. An experimental design that maximised evaluation on the available data was set up. To meet the rigour requirement, two separate corpora were used, one for training and one for testing.

The section shows that the data was split for development, validation and evaluation in a way that maximises the use of the data, again for reasons of rigour. The section also explains how N-Fold validation is implemented in this setup.

The command-line arguments for the three segmenters developed as part of this study are specified.

The benchmark that the segmenters were compared to is identified, and how these will be evaluated, as well as how the experiments were conducted is also detailed.

7.2 Results

In this section of the document, the results of the experiments conducted are presented. Three primary experiments were conducted.

The first experiment is the segmentation of isiXhosa using Branching Entropy and Bracing Joint Entropy techniques in the isiXhosa Branching Entropy Segmenter (XBES). XBES has twelve (12) modes, six (6) being Bracing Joint Entropy based, the other six (6) being Branching Entropy based, with various modes of configuration based on n-gram size smoothing and operator used.

The second experiment considers the predictability constraints of isiXhosa words, particularly validating the link between branching entropy and predictive complexity theory. This experiment establishes the n-gram length whose ensemble captures the predictability of isiXhosa.

The third experiment evaluates the segmentation of isiXhosa using heuristic guided language models in the isiXhosa Heuristics Maximum Likelihood Segmenter (XHMLS). This segmenter implements the four language models defined in Sect. 6.1: *IsiXhosa Probabilistic Models*.

For each of the segmenter evaluation experiments, the configuration with the best accuracy, and the one with the best F1 Score for each mode or model, is presented.

7.2.1 Benchmark performance

Starting with the performance of the benchmarks, the random segmenter presented an average accuracy from ten (10) runs of $50.1 \pm 0.16\%$. This implies that any segmenter whose accuracy is below this threshold would actively degrade segmentation. The benchmark ten-fold validation highest average accuracy from Morfessor-Baseline was measured at $77.2 \pm 0.10\%$ and achieved when trained with one million words. In terms of the performance when compared to the size of the training set size, see Figure 7-4, the accuracy of Morfessor-Baseline increases with training set size and flattens from 100 000 words onwards. In Table 7-3, the p-value listed next to a result

refers to the p-value between the results on the line where the p-value is written and the results on the line before. The average accuracy maximum of 77.2% is not statistically significant from the second-highest accuracy of 77.1% at 1.5 million words, as the p-value between the two is 0.492. The second highest accuracy is also not statistically significant from the third-highest accuracy, which is 76.9% at 100 000 words, as the p-value is 0.105. However, the third-highest Morfessor-Baseline accuracy is statistically lower than the highest accuracy as the p-value is 0.0195.

Table 7-3: Benchmark Overall Boundary Identification Results with p-values

Method	Highest % Accuracy (training Size)	Highest % F1 Score (training Size)
Random	50.1 $\pm$ 0.16	35.7 $\pm$ 0.16
Morfessor-Baseline	$p = 0.0195$ $77.2 \pm 0.10(1m)$ $77.1 \pm 0.20(1.5m), p = 0.492$ $76.9 \pm 0.28(100k), p = 0.105$	$p = 0.002$ $48.9 \pm 0.75 (10K)$ $48.7 \pm 0.64 (100K), p = 0.375$ $44.0 \pm 0.24 (100), p = 0.002$

In terms of the F1 Score, the random segmenter achieved an F1 Score of $35.7 \pm 0.16\%$ and Morfessor-Baseline achieved a maximum of $48.9 \pm 0.75\%$ when trained with 10 000 words. However, this is comparable to the Morfessor-Baseline's F1 Score of 48.7% when trained with 100 000 words as the p-value between the two is 0.375. For larger training set sizes the F1 Score degrades. Therefore, the best combined overall performance is of Morfessor-Baseline trained with 100 000 words as both the F1 Score and accuracy are high.

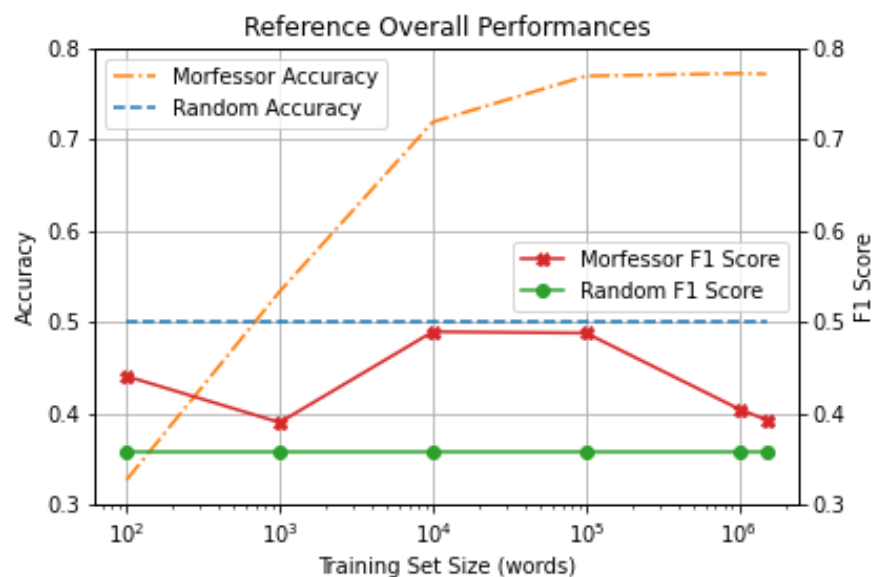


Figure 7-4: Benchmark overall performance by training set size

In terms of out of vocabulary (OoV) performance, Morfessor-Baseline's highest OoV accuracy was measured at 73.3 ± 0.26 %, performed with a training set size of 1 million words, and is the highest because it is statistically significant from the second highest and third highest accuracy scores.

Table 7-4: Benchmark OoV Boundary Identification Results with p-values

Method	Highest % Accuracy (training Size)	Highest % F1 Score (training Size)
Random	50.1 ± 0.16	35.7 ± 0.16
Morfessor-Baseline	$p = 0.002$ 73.3 ± 0.26 (1m) 72.7 ± 0.51(1.5m), $p = 0.012$ 70.6 ± 0.64(100k), $p = 0.002$	$p = 0.105$ 44.3 ± 0.42 (10K) 44.2 ± 0.24 (100), $p = 1$ 43.5 ± 1.14 (100K), $p = 0.084$ 39.3 ± 1.36 (1m), $p = 0.027$

As for out of vocabulary (OoV) F1 Scores, the top three scores are achieved at 10 000, 100 and 100 000 words with scores of $44.3 \pm 0.42\%$, $44.2 \pm 0.24\%$ and $43.5 \pm 1.14\%$ respectively, see Table 7-4 for the p-values and Figure 7-5 for the trendline. These three are equivalent as the pair-wise Wilcoxon p-scores among them is higher than 0.05. Everything else is statistically lower.

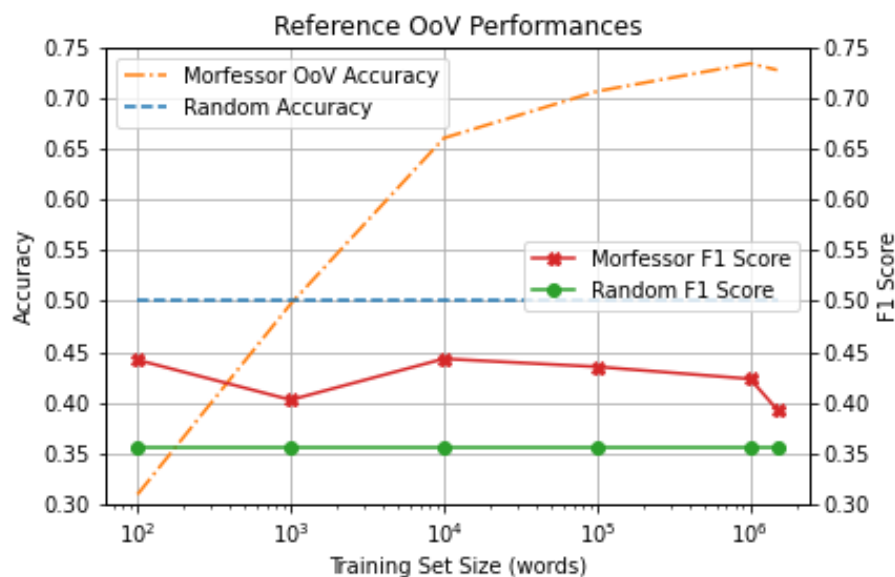


Figure 7-5: Benchmark OoV overall performance by training set size

For out of vocabulary words, there is no combined overall Morfessor-Baseline best performance, as the performance peaks are at different training set sizes. Having defined the benchmark, the next task is the results of the two segmenters developed in this study.

7.2.2 IsiXhosa Branching Entropy Segmenter (XBES) Results

Table 7-5 shows the best performing configurations of the different modes of the Branching Entropy (BE) and the Bracing Joint Entropy (BJE) compared to the random segmenter and Morfessor-Baseline. The best performances overall are marked with a ‘*’.

7.2.2.1 Overall performance of XBES

Starting with the Bracing Joint Entropy (BJE) scorer, the best accuracy was achieved by the Variation of Bracing Joint Entropy score (VBJE) at $75.0 \pm 0.31\%$. This was achieved with a training size of 1.5 million words using 15-grams and less. The best F1 Score for BJE, at $46.7 \pm 1.0\%$, was achieved by the mean normalised Variation of Bracing Joint Entropy score (N_{μ} VBJE) trained with 1.5 million words and a 10-gram model. The Wilcoxon p-value of with the next BJE highest F1 Score was 0.02841 which is less than 5%. The Bracing Joint Entropy models were not evaluated for smoothing because interpolation is not clear for bracing joint entropy models. Such work could be done in a follow-up study.

Table 7-5: XBES Overall Boundary Identification Results

Method	Highest Accuracy (Smoothing/Op/training Size/n-gram length)	Highest F1 Score (Smoothing/Op/training Size/n-gram length)
Random	50.1 ± 0.16	35.7 ± 0.16
XBES -BJE	44.3 ± 0.350 (100K/2)	45.6 ± 0.06 (10K/4-max)
XBES - N_{μ} BJE	71.1 ± 0.18 (1m/15-max)	45.6 ± 0.27 (1m/11)
XBES - N_z BJE	72.2 ± 0.19 (1m/12)	43.5 ± 1.06 (1.5m/9)
XBES -VBJE	75.0 ± 0.31 (1.5m/2-15)	41.9 ± 1.07 (1.5m/10)
XBES - N_{μ} VBJE	69.8 ± 0.67 (1.5m/15-max)	46.7 ± 1.0 (1.5m/10)
XBES - N_z VBJE	$70.2 \pm .28$ (1.5m/13)	$44.2 \pm .36$ (1.5m/9)
XBES-BE	71.6 ± 0.35 (No/Min/100K/4)	55.3 ± 0.12 (No/Sum/1m/7)
XBES-VBE	72.4 ± 0.25 (Yes/Min/1.5m/9)	58.0 ± 0.10 (No/Sum/1.5m/9)*
XBES - N_{μ} VBE	75.8 ± 0.60 (No/Sum/1m/11)	53.6 ± 0.35 (No/Sum/1.5m/13-max)
XBES - N_z VBE	77.4 ± 0.32 (No/Sum/1.5m/11)*	55.5 ± 1.18 (No/Sum/1.5m/9)
Morfessor-Baseline	77.2 ± 0.10 (1m)*	48.9 ± 0.75 (10K)

For the Branching Entropy (BE) scorers, accuracy peaked at $77.4 \pm 0.32\%$ achieved using the z-score normalised Variation of Branching Entropy (N_z VBE) model trained with 11-grams of a 1.5 million words training set. This accuracy is comparable to Morfessor-Baseline's peak accuracy of $77.2 \pm 0.10\%$, which was achieved at 1 million training word, as the Wilcoxon p-value between the two was measured at 0.07446. The F1 Score for the study was achieved by the un-normalised Variation of Branching Entropy (VBE) variant of Branching Entropy, at $58 \pm 0.10\%$, using a 9-

gram configuration trained on 1.5 million words. The pair-wise Wilcoxon p-values with the second-best F1 Score showed statistical significance at 0.0051. This F1 Score is superior to both Morfessor-Baseline and the BJE highest F1 Score. The results show that modified Knesser Ney smoothing does not provide any benefit to branching entropy overall accuracy and F1 Scores.

For Normalised Variation of Branching Entropy modes, the sum of the left- and right-branching measures performed better than the minimum of the two, implying that a smoothing effect is better, as the sum is a form of averaging the two branching directions. However, for the un-normalised metrics, the min operator gave better accuracy whilst the sum operator gave better F1 Scores.

7.2.2.2 XBES overall performance trends

The trend of the best performing scorer for branching entropy and bracing joint entropy over the training set size uncovers an unexpected result from the variation metric.

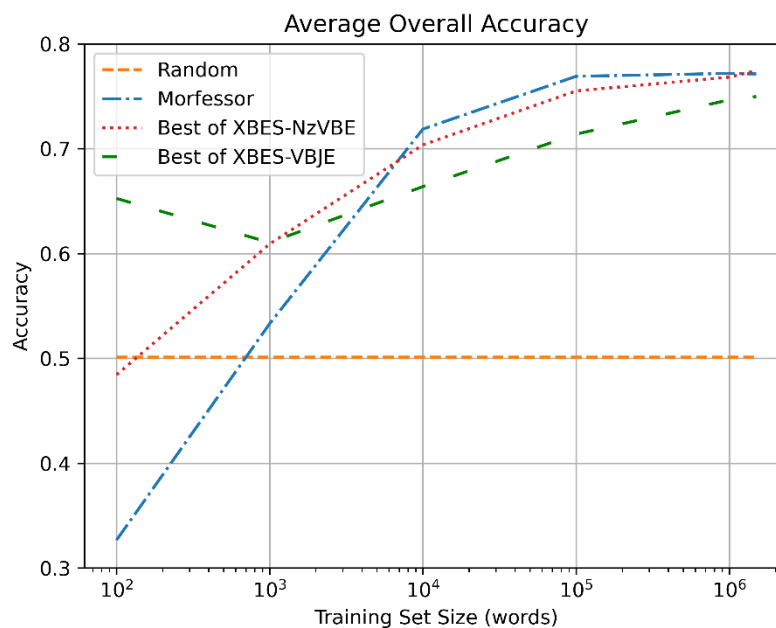


Figure 7-6: Average overall accuracy of the segmenters and the best BE and BJE modes by training set size

For the accuracy results shown in Figure 7-6, the bracing joint entropy gives impressive accuracy for the smallest training set with an accuracy of around 65% whilst the others are below 50%. The scorer is the un-normalised variation of bracing joint entropy (VBJE). These are very encouraging results because they mean that the proof that the variation metric captures predictive information has empirical support, in at least the bracing joint entropy metric when trained with small data.

Also, this makes the use of VBJE very attractive for use in endangered languages when the metric of choice is segmentation accuracy. For a training set size of 1000 words, BJE is comparable to BE.

As for training set sizes longer than 1000 words, Morfessor-Baseline outperformed the other scorers' accuracy except at 1.5 million words where the N_2 VBE is comparable. The trends suggest that more training data would result in Branching Entropy outperforming Morfessor-Baseline. However, the Bracing Joint Entropy shows a much steeper trajectory, which suggests that Bracing Joint Entropy's accuracy might exceed both Morfessor-Baseline and Branching Entropy, given enough data.

Figure 7-7 shows the F1 Score trends. Morfessor-Baseline and XBES's BJE (N_0 VBJE), show better performance than Branching Entropy's best for F1 Scores, un-normalised VBE, when trained on 100 words. However, from 1000 words onwards, VBE outperforms all the other metrics and suggest that it may improve with more data.

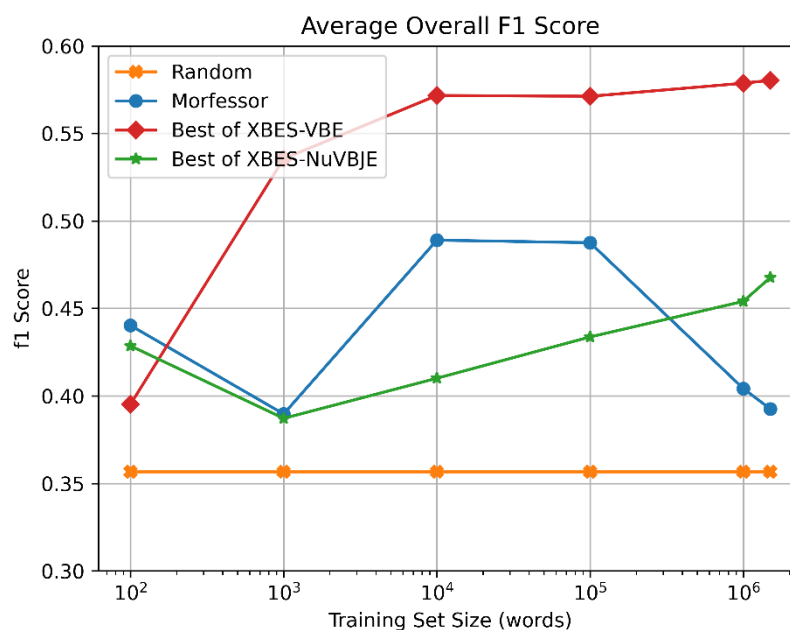


Figure 7-7: Average overall F1 Score of the Random Segmenter, Morfessor-Baseline, and the best BE and BJE modes by training set size

The F1 Score of Bracing Joint Entropy shows a steady performance increase with more data, which suggests that more data would continue to improve its F1 Score. Even though the trajectory suggests improvement, it is doubtful whether enough data would allow it to reach the Variation of

Branching Entropy (VBE). Again, this supports the understanding that Variation of Branching Entropy captures the predictive information in the text.

7.2.2.3 XBES Performance on OoV words

For out of vocabulary (OoV) words, the un-normalised Branching Entropy (BE) and z-score normalised Variation of Branching Entropy (N_zVBE) give the best accuracy and are comparable at 76.8 ± 0.45 and $76.7 \pm 0.47\%$ respectively. The p-value between them is 0.9527. These outperform Morfessor-Baseline's $73.3 \pm 0.26 \%$, and the p-value of Morfessor-Baseline relative to the two is 0.00195.

Table 7-6: XBES OoV Boundary Identification Results

Method	Highest OoV Accuracy (%) (Smoothing/Op/training Size/n-gram length)	Highest OoV F1 Score (%) (Smoothing/Op/training Size/n-gram length)
Random	50.1 $\pm$ 0.16	35.7 $\pm$ 0.16
XBES -BJE	48.6 $\pm$ 0.37 (100K/2)	47.3 $\pm$ 0.12 (10K/4-max)
XBES -N _u BJE	74.1 $\pm$ 0.26(1.5m/12)	50.1 $\pm$ 0.37(1.5m/9)
XBES -N _z BJE	74.7 $\pm$ 0.31 (1.5m/17)	47.9 $\pm$ 0.66(1.5m/10)
XBES -VBJE	73.6 $\pm$ 0.31(1.5m/15-max)	43.9 $\pm$ 0.63 (1.5m/15-max)
XBES -N _u VBJE	71.2 $\pm$ 0.49 (1.5m/15-max)	47.7 $\pm$ 0.96 (1.5m/15-max)
XBES -N _z VBJE	73.9 $\pm$ 0.34 (1.5m/15)	48.9 $\pm$ 0.54 (100K/9)
XBES-BE	76.8 $\pm$ 0.45 (Yes/Sum/1.5m/9)*	65.3 $\pm$ 0.12 (Yes/Sum/1.5m/9)*
XBES-VBE	74.7 $\pm$ 0.33 (No/Sum/1.5m/7)	65.0 $\pm$ 0.48 (No/Sum/1.5m/7)*
XBES-N _u VBE	74.4 $\pm$ 0.63 (No/Sum/1.5m/13-max)	51.8 $\pm$ 1.17 (No/Sum/1.5m/13-max)
XBES-N _z VBE	76.7 $\pm$ 0.47 (No/Sum/1.5m/13)*	54.3 $\pm$ 1.03 (No/Sum/1.5m/9)
Morfessor-Baseline	73.3 $\pm$ 0.26 (1m)	44.3 $\pm$ 0.42 (10K)

The best bracing joint entropy (BJE) also outperforms Morfessor-Baseline with OoV accuracy of $74.7 \pm 0.31\%$ when trained with 17-grams from 1.5 million words using the z-score normalised bracing joint entropy mode. The p-score is 0.00195. Importantly, the smoothed un-normalised branching entropy gives high OoV accuracy from very small training sets and is stable across the range of the training set sizes as visible in Figure 7-8.

For the out of vocabulary F1 Scores, un-normalised Branching Entropy (BE) and un-normalised Variation of Branching of Entropy (VBE) give the best performance and their performance is comparable at $65.3 \pm 0.12\%$ and $65.0 \pm 0.48\%$ respectively and the p-value between them is 0.08398. Of interest is the performance trend of the mode with the second-best F1 Score, the un-normalised Variation of Branching Entropy (VBE) mode which has a better OoV F1 Score than the other modes for the training set of 1000 to 1 million words when trained with unsmoothed 7-

grams. This shows that the variation metric, an information theoretic predictive measure, captures the word predictive information for smaller training set better than its counterpart.

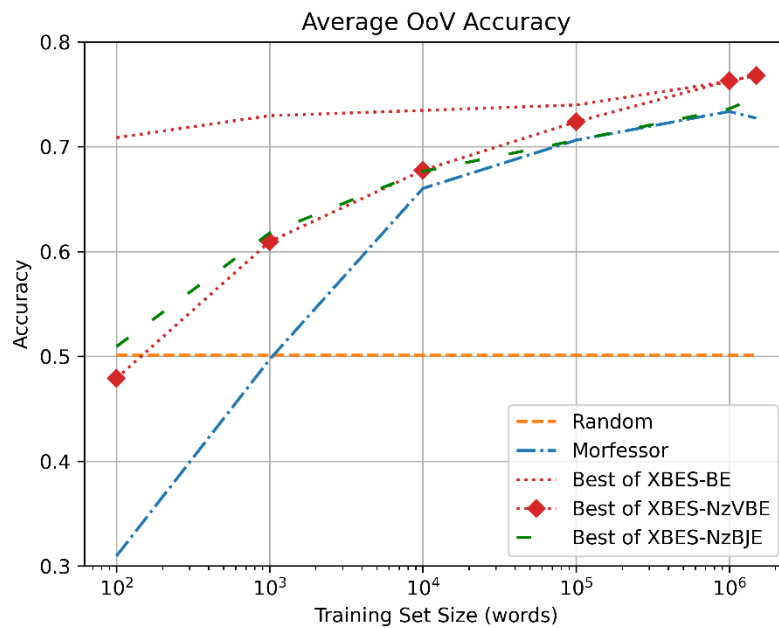


Figure 7-8: Average OoV Accuracy of the Random Segmenter, Morfessor-Baseline, and the bests BE and BJE modes by training set size

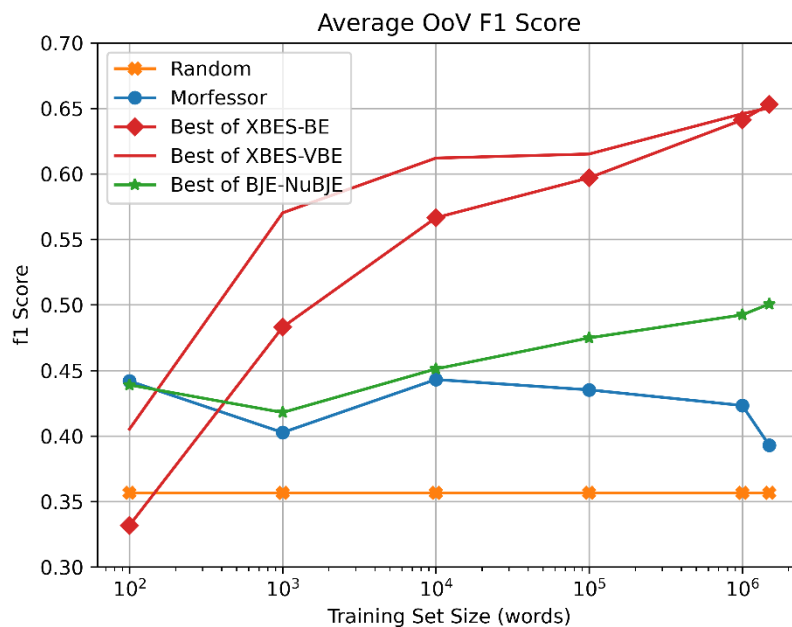


Figure 7-9: Average OoV F1 Score of the Random Segmenter, Morfessor-Baseline, and the bests BE and BJE modes by training set size

In out of vocabulary words, mKN smoothing improved the un-normalised branching entropy performance but did not aid the other modes.

7.2.2.4 Conclusions on XBES Performance

It is clear XBES is comparable to Morfessor-Baseline in accuracy but far superior to it when it comes to the F1 Score for surface segmentation of isiXhosa, depending on the configuration one uses.

It is also worth suggesting that XBES, using Variation of Bracing Joint Entropy (VBJE), should be considered the new benchmark for unsupervised accuracy of segmenting scarce resource languages because of its accuracy when trained with 100 words.

Out of vocabulary (OoV) accuracy is better achieved with a smoothed un-normalised Branching Entropy mode trained with 9-grams as this configuration gives high OoV accuracy for the entire training set range. This same configuration also gives good F1 Scores, but it is outperformed by the un-normalised Variation of Branching Entropy (VBE) when trained on unsmoothed 7-grams. This begs the question of whether there is a relationship between mKN smoothing and entropy finite differentiation, i.e. predictability gain (**Definition 3-19**).

7.2.3 Predictive information in isiXhosa

In this section, the results of measuring the predictability of isiXhosa words are presented. This is done using the information theoretic predictive metrics explained in Sect. 3.5: *Predictive Information Metrics*. These predictive metrics use information theoretic measures to measure the length of the ensembles that contain all the predictive information in a stochastic process. The figures below show the trend for the different predictive metrics for isiXhosa text. The ten-fold analysis made no difference as the trends were the same with a set corpus size. This makes sense because entropy is an averaging process.

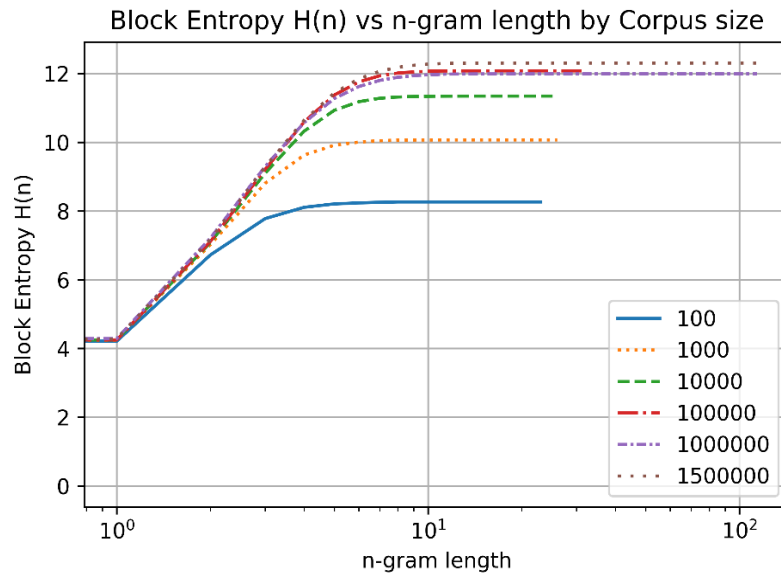


Figure 7-10: Trend in Block Entropy by n-gram length

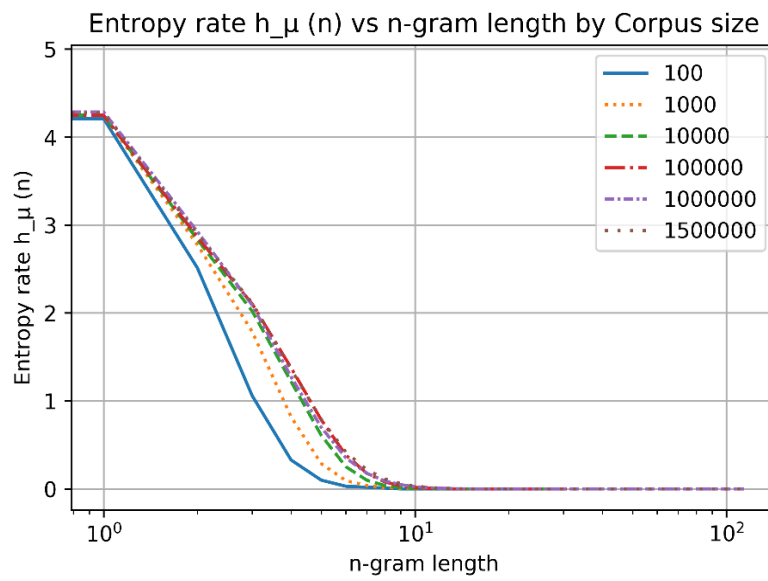


Figure 7-11: Trend in Entropy Rate trend by n-gram length

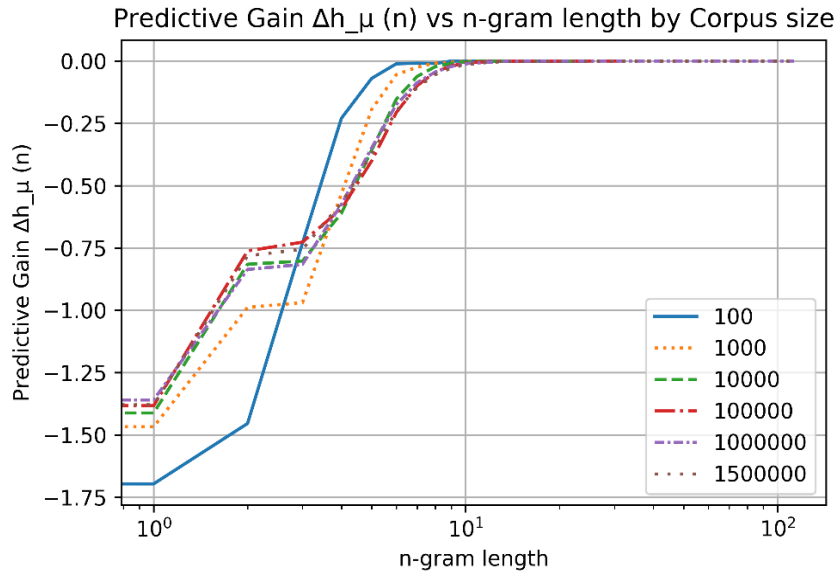


Figure 7-12: Trend in Predictive Gain by n-gram length

Figure 7-10 shows that the block entropy starts around 4.2 for the unigrams for all the analysis set sizes, and converges to some limits, whereafter longer n-grams do not provide more predictability. The n-gram length at convergence for the largest dataset is approximately 10. Figure 7-11 also confirms that the entropy rate vanishes at around the 10-grams for larger corpus sizes and much earlier for smaller sets, but that it should be at least longer than 6 irrespective of corpus size.

From Figure 7-12 one can see that predictive gain is only obtained up to the 10-gram. Of interest is also that trigrams add little predictive information for data sets larger than 100. This means that the initial predictability is on unigrams and bigrams, and then 4-grams to 10-grams.

This also implies that the bigrams provide the trigram information content. It would be interesting to see if this trend is reflected in other Nguni languages. These results reveal that isiXhosa words are periodic with its entropy rate vanishing at 10-grams for corpora of the size and characteristics used in this study.

These results, in some measure, correlate with **Theorem 4-1** and **Theorem 4-2** wherein it is stated that Branching Entropy (BE), h , captures the block entropy rate, h_μ , such that

$$h_\mu(n+1) = \sum_{x_{1..n} \in \mathcal{X}^n} p(x_{1..n}) h(x_{1..n}),$$

and variation of branching entropy (VBE), δh , captures predictive gain, Δh_μ , such that

$$\Delta h_{\mu}(n+1) = \sum_{x_{1..n} \in \mathcal{X}^n} p(x_{1..n}) \delta h(x_{1..n}).$$

According to the results in this section and the above theorems, branching entropy and the variation of branching entropy should capture the predictive information when

$$n+1 = 10 \Rightarrow n = 9. \quad (7-1)$$

If one considers the overall XBES results, VBE provides the best results when trained with 9-grams. The same 9-grams appear as the best configuration for out of vocabulary (OoV) words when segmenting with the mKN smoothed branching entropy (BE) mode. This gives the impression that smoothing assists the classic branching entropy to extract the predictability from the text. This means that, at the least, the aspect of predictability needs further investigation.

7.2.4 IsiXhosa Heuristics Maximum Likelihood Segmenter (XHMLS) performance

This section considers the benchmarks for isiXhosa unsupervised surface segmentation to be Morfessor-Baseline and XBES, and the best results from XBES are included. The benchmark accuracy is therefore 77.4% from XBES, which is comparable to Morfessor-Baseline's 77.2%.,

For the F1 Score, XBES is the benchmark, but Morfessor-Baseline's F1 Score is presented as well.

7.2.4.1 XHMLS Overall performance results

The overall results, including the best performance per XHMLS model, are shown in Table 7-7. The results are shown with configuration information (i.e. smoothed or not, the training set size and the language model n-gram level that produced the results).

Table 7-7. Best XHMLS Overall Boundary Identification Results per model

Method	Highest % Accuracy (Smoothing/ Training Size/n-length length)	Highest % F1 Score (Smoothing/ Training set Size/n-gram length)
Random	50.1 ± 0.16	35.7 ± 0.16
Morfessor-Baseline	77.2 ± 0.10 (1m)	48.9 ± 0.75 (10K)
Best XBES	77.4 ± 0.32 (No/Sum/1.5m/11)	58.0 ± 0.10 (No/Sum/1.5m/9)
XHMLS-MS	74.7 ± 0.24 (Yes/1m/4)	59.5 ± 0.30 (Yes/1.5m/4)
XHMLS-IA	75.2 ± 0.18 (No/100K/2)	59.4 ± 0.20 (No/10K/3)
XHMLS-AL	73.7 ± 0.01 (Yes/1.5m/2)	59.0 ± 0.04(Yes/1.5m/3)
XHMLS-CL	74.0 ± 0.01(Yes/10K/4)	49.6 ± 0.04(Yes/10K/4)

From Table 7-7, the best 10-fold average accuracy for XHMLS, 75.2 ± 0.18 %, is achieved by the Independent Affixes model (XHMLS-IA) of XHMLS at a training set size of one hundred thousand (100 000) words using an unsmoothed bigram language model. This accuracy is, however, inferior to Morfessor-baseline's accuracy of 77.2% and XBES's 77.4% as the Wilcoxon Signed Rank test (Demšar, 2006) p-value against the two was measured at 0.00195.

However, the accuracy of the XHMLS-IA model shows high accuracy even with a small training set (100 words), and is stable across the training set sizes, as shown in the trend lines of Figure 7-13.

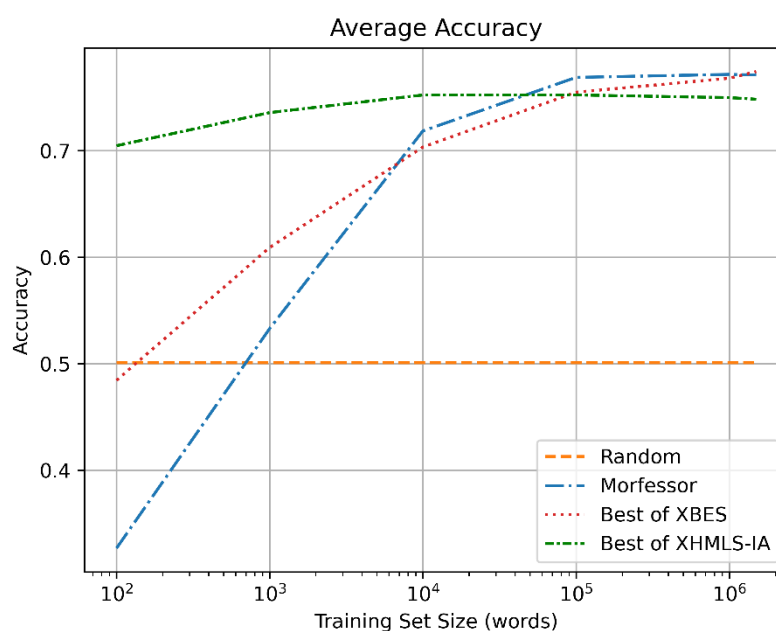


Figure 7-13: Average overall accuracy of the best modes for the segmenters by training set size

As for the F1 Score, all the XHMLS modes show segmentation F1 Scores superior to Morfessor-Baseline at some training set size. The XHMLS mode whose best is the lowest is the Circumfix Link model (XHMLS-CL) of XHMLS whose best F1 Score is 49.6 ± 0.04 % at a training set size of ten thousand words (10 000) using a smoothed 4-gram language model. Even this outperforms Morfessor-Baseline as the p-value between the two is 0.0039.

Regarding the best XHMLS F1 Score, there is no significant difference between the best of the Morpheme Segments (XHMLS-MS) and the Independent Affixes (XHMLS-IA) models as the p-value between the two is 0.4921. The best F1 Scores for the two are respectively 59.5 ± 0.30 % and 59.4 ± 0.20 % respectively with the former having been trained with mKN smoothed 4-grams

from a 1.5 million word corpus and the latter with unsmoothed trigrams from ten thousand words. These two outperform every other mode and have p-values of 0.00195 and less.

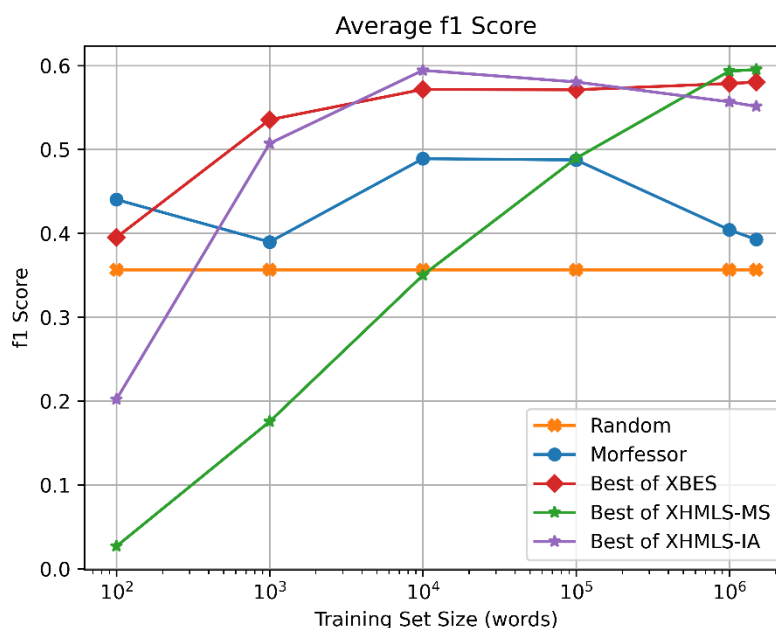


Figure 7-14: Average overall F1 Scores of the best modes for the segmenters by training set size

Of interest are the trend lines of the two modes. The XHMLS-IA reaches its best F1 Score with a smaller training set size than XHMLS-MS, but it is still lower than XBES for small training sets, however, the overall performance of XHMLS-IA degrades for training sets above 10 000 words in both accuracy and F1 Score. This might mean that the XHMLS-IA mode overfits, but only OoV performance can verify that.

As is evident from Table 7-7 the effect of modified Kneser-Ney smoothing improves performance for all the XHMLS models except the Independent Affixes model.

7.2.4.2 XHMLS out of vocabulary results

For the out of vocabulary words, a different picture emerges where the Independent Affixes (XHMLS-IA) model provides the best OoV accuracy when trained with unsmoothed trigrams from 1.5 million words. This OoV accuracy is statistically better than all the work done in this study as the pairwise p-value with the others is 0.00195. The best OoV accuracy is around 81.8% and the F1 Score is 68.7% as shown in Table 7-8. The trend of the XHMLS-IA OoV Accuracy is higher than the other segmenters across the training set sizes, as seen in Figure 7-15.

Table 7-8: Best XHMLS OoV Boundary Identification Results per model

Method	Highest % Accuracy (Smoothing/ Training Size/n-length length)	Highest % F1 Score (Smoothing/ Training set Size/n-gram length)
Random	50.1 $\pm$ 0.16	35.7 $\pm$ 0.16
Morfessor-Baseline	73.3 $\pm$ 0.26 (1m)	44.3 $\pm$ 0.42 (10K)
Best XBES	76.8 $\pm$ 0.07 (Yes/Sum/1.5m/9)	65.3 $\pm$ 0.12 (Yes/Sum/1.5m/9)
XHMLS-MS	73.2 $\pm$ 0.24(Yes/1m/4)	16.8 $\pm$ 0.30 (Yes/1.5m/5-max)
XHMLS-IA	81.8 $\pm$ 0.18 (No/1.5m/3)	68.7 $\pm$ 0.10 (No/100K/3)
XHMLS-AL	80.1 $\pm$ 0.30 (Yes/1.5m/3)	64.8 $\pm$ 0.05 (Yes/1.5m/3)
XHMLS-CL	77.1 $\pm$ 0.05 (Yes/100K/3-max)	53.3 $\pm$ 0.07(Yes/10K/3-max)

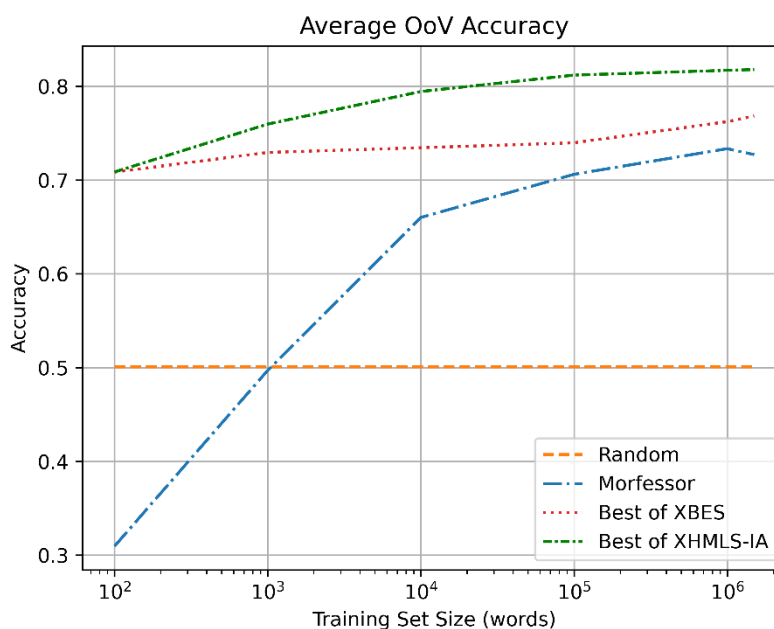


Figure 7-15: Average OoV Accuracy of the best modes for the segmenters by training set size

The best OoV F1 Score is also attributed to XHMLS-IA model trained with unsmoothed tri-grams from a corpus of one hundred thousand (100 000) words. From the trend line in Figure 7-16, one can see that the XHMLS-IA model provides superior generalisation for the F1 Score from as small as one thousand words.

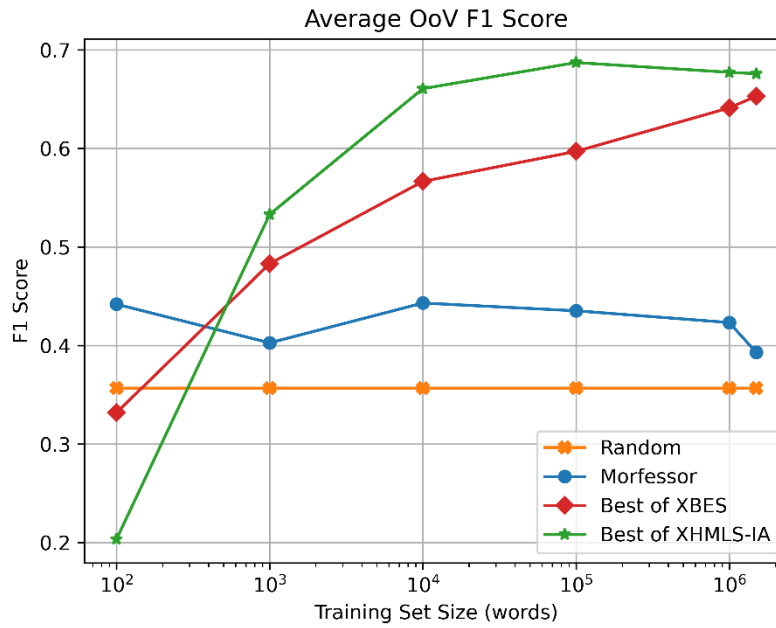


Figure 7-16: Average OoV F1 Scores of the best modes for the segmenters by training set size

7.2.4.3 Conclusions of XHMLS results

In summary, XBES and Morfessor-Baseline continue to maintain their benchmark status on unsupervised segmentation of isiXhosa if the metric is overall accuracy. A different picture appears when the metric of choice is the F1 Score. In this case, the XHMLS is the benchmark as two of its models outperform both XBES and Morfessor Baseline.

An even more interesting picture appears when looking at performance on out of vocabulary (OoV) words, the measure of generalisation. In both accuracy and F1 Score, the Independent Affixes model of XHMLS (XHMLS-IA) dominates without the need for language model smoothing for all the training set sizes and has a superior F1 Score for all sizes but the smallest training set.

Since the Independent Affixes mode being the best mode for XHMLS, this confirms the summary of the dependencies within the language specified in Sect. 5.4: *Dependencies among affixes*. In that section, it was identified that morpheme dependence is mostly within the affixes, that there is little circumfixing and that root, prefixes and roots have very little dependence because of their major reuse in the language across word categories.

7.2.5 Conclusions on Experiments Results

The experiments conducted resulted in the best morpheme boundary identification accuracy being those of XBES and Morfessor-Baseline and the F1 Score being that of XHMLS. Model smoothing on XBES is useful only for the standard branching entropy technique and provides no improvement on the other. The same applies to XHMLS, where smoothing improves performance on all the models except the Independent Affixes model, which is dominant in OoV performance.

The predictiveness of isiXhosa words is investigated and confirmed using information theoretic and complexity metrics. The experiments reveal that isiXhosa is periodic with its entropy rate vanishing at 10-grams for corpora of the size and with the characteristics used in this research study. This study therefore shows that it is possible to predict the maximum n-gram size that captures the predictive information of isiXhosa text.

7.3 Summary

This chapter specified how the experiments were conducted and recorded all the results achieved.

On the experimental setup, the data sources; the data splits that ensure rigour; and the maximum use of the data have been presented. In addition, the benchmarks in unsupervised segmentation were identified and described with detailed explanations on the experimental procedure.

Results of the evaluations, including out of vocabulary results, are provided. It is found that XBES and Morfessor-Baseline provided the best accuracy results and XHMLS provided the best F1 Score result. In addition, it is found that XHMLS provided the best out of vocabulary results when compared to XBES and Morfessor-Baseline for both accuracy and F1 Score. Lastly, the results show that XHMLS has high accuracy and F1 Score performance from very small training sets and furthermore one of its modes provided stable performance across the training set sizes.

8. CONCLUSIONS AND FUTURE WORK

In this chapter, a summary of the work presented in this document is provided, the findings are presented, direction on future work that could be conducted from this study is suggested and concluding remarks are given.

The major finding of this research is that an unsupervised, morphological surface segmenter is achievable for isiXhosa, which achieves better results than a comparable, accepted benchmark.

8.1 Summary of the work

This thesis investigated the use of variants of branching entropy and heuristics guided language modelling of isiXhosa words in the unsupervised segmentation of isiXhosa. This includes the theory, application and evaluation of these techniques.

The work commenced by motivating for word segmentation in the natural language processing (NLP) of morphologically complex languages. The motivation extended to the use of unsupervised machine learning in the word segmentation of agglutinating and resource-scarce languages, a technique that avoids the use of expensive and time consuming resources, like linguistics experts.

For sample depth, the study used the language isiXhosa, which is in the largest mother tongue subgroup in South Africa, namely Nguni, with the aim of advancing the use of unsupervised machine learning in the surface segmentation of Nguni languages, with isiXhosa as a language of application.

To set the scene in the use of technology, an overview of the vast work done in the unsupervised segmentation of words is provided. The work is extensive, covers many decades and a vast array of technologies. The latest developments have seen the use of deep architectures in artificial neural networks. Surprisingly, work on deep neural networks seems to need segmentation of words as input to train word embeddings that are then used for other higher-order NLP tasks like translation, making yet another case for word segmentation.

For this study, the literature study revealed that the use of variants of branching entropy and character-level language modelling were two suitable techniques for unsupervised segmentation of isiXhosa. The study also identifies the morpheme boundary identification accuracy and F1 Scores as the most appropriate evaluation metrics for this study. The study also reveals that the benchmark of choice for segmentation studies is Morfessor-Baseline.

In the first technique of branching entropy, preliminaries on information theory and complexity and predictability measures are given. These guide the work in proving that the different branching entropy variants capture different aspects of predictive information. A combined bi-directional character-level entropy language model, with an option for modified Kneser-Ney smoothing, is also presented. This model combines the branching entropy character-level language for left-branching and right-branching models. A new variant to branching entropy techniques that does not require directional modelling is introduced, the so-called bracing joint entropy (BJE) model.

In the second technique using word language modelling work, probabilistic graphical modelling (PGM) is used to model the morphology of isiXhosa words. To get to a set of models that would capitalise on the structure of the language, the surface morphological structure of the language is studied. This study revealed two things, a set of heuristics that could be used to reduce the search space of a probabilistic model. In addition, the work lead us to a set of possible statistical dependencies between the morphemes in a language. These possible dependencies lead us to a combination of four possible models of differing complexity for the language. To capture the generative and statistical dependence model in a graphical model, modifications were done to the PGM notation and also to the use of the plate system to model conditional probability recurrence and to differentiate between generative links and probabilistic conditional links.

To verify this work, three experiments were conducted. The first experiment used branching and bracing joint entropy in the segmentation of isiXhosa. The second experiment, which confirmed branching entropy, established the n-gram length that encapsulates the predictability of isiXhosa words. The last experiment segmented isiXhosa using heuristic guided probabilistic language models.

8.2 Contributions

This work makes six (6) contributions to the work on automated text segmentation.

The first contribution is the combined, bi-directional, branching entropy language model (BELM). The BELM combines two directional branching entropy models into one and supports the modified Kneser-Ney smoothing algorithm. This improved the performance of the basic branching entropy model but did not provide improvement to the other modes of XBES.

The second contribution is the Bracing Joint Entropy (BJE) for isiXhosa. BJE does away with directional branching entropy altogether by combining the two into a joint entropy. This mode of “branching” entropy performed better than Morfessor and other branching entropy modes when trained with one hundred words and was comparable when trained with one thousand words.

The third contribution is the use of Branching Entropy in the segmentation of isiXhosa, opening the way for this technique to be used in other Nguni languages. This work matches the current state-of-the-art in accuracy.

The fourth contribution is proof that the different branching entropy variants capture different aspects of predictive information. Specifically, it is proved that branching entropy captures entropy rate and variation of branching entropy captures predictive gain.

The fifth contribution is the determination that ensembles of character n-grams of length ten (10) capture the predictive information in isiXhosa words. This motivates for branching entropy techniques to use 9-grams.

Lastly, the work contributes the use of morphology heuristics as a guide to probabilistic graphical modelling of the segmentation as a substitute to non-parametric language modelling techniques. This is proven in isiXhosa with the use of isiXhosa heuristics and achieve state-of-the-art results in unsupervised F1 Scores. This result could be generalised to other Nguni languages.

8.3 Main Findings

This section evaluates how much of the intended research work was achieved and to what extent.

As its main aim, this study set out to:

- Advance the use of unsupervised machine learning in the morphological segmentation of Nguni languages, with isiXhosa as a language of application.

The research questions that the study set out to answer were:

- Question 1: Are there techniques that can readily advance the use of unsupervised machine learning in the segmentation of Nguni languages, evaluated using isiXhosa?
- Question 2: Can the unsupervised segmentation of isiXhosa words, using these techniques, be improved relative to a benchmark?
- Question 3: What configuration of these techniques provides the best performance, and why?

The research objectives that were specified, are:

- Objective 1: Identify a benchmark in unsupervised morphological segmentation and evaluate it on isiXhosa,

- Objective 2: Research and investigate unsupervised text segmentation techniques that could readily be used in isiXhosa,
- Objective 3: Identify, extend, implement and evaluate unsupervised machine learning techniques that can readily advance the morphological segmentation of isiXhosa.

The activity of establishing whether the intended research aims, questions and objectives were resolved, or not, will be done from the bottom up. This means the research objectives are addressed first. The results of the research objectives lead to outcomes on whether the questions were answered, or not. The answers to the research questions, once combined, will give an indication whether the aim of the study was achieved.

8.3.1 Addressing Research Objectives

This section details work done to address the research objectives.

8.3.1.1 Findings on Objectives 1 and 2

Objective 1 was to identify a benchmark in unsupervised surface segmentation and evaluating it using isiXhosa.

Objective 2 was to research and investigate unsupervised text segmentation techniques that could readily be used in isiXhosa.

Both objectives were met by:

- A literature study of unsupervised text segmentation techniques, published worldwide, that is documented in Chapter 2: *Literature Review*.
- The literature study culminating in the identification of Morfessor-Baseline as the benchmark for the experiments to be conducted in this thesis, in Sect. 2.7: *Techniques and Benchmark of choice for isiXhosa segmentation*.

8.3.1.2 Findings on Objective 3

Objective 3 was to identify, research, extend, implement and evaluate unsupervised machine learning techniques that could readily advance the morphological segmentation of isiXhosa.

This objective was met as follows:

- The literature study, Chapter 2: *Literature Review* on unsupervised text segmentation techniques culminated in the identification of branching entropy and language model based segmentation as the techniques that could be used in the unsupervised

segmentation of isiXhosa, in Sect. 2.7: *Techniques and Benchmark of choice for isiXhosa segmentation*.

- The two techniques identified were researched, extended and implemented as detailed in chapters three (3) to six (6).
 - Chapter three (3) researched the underpinning metrics to branching entropy including predictive information metrics.
 - Chapter four (4) gave specifics of branching entropy, extended the work with a new metric, bracing joint entropy (BJE) and proved that the different variants of branching entropy capture predictive information.
 - Chapter five (5) researched and formulated a set of isiXhosa guiding heuristics that could be used to guide a probabilistic language model and proposes four (4) models for the surface segmentation of isiXhosa.
 - Chapter six (6) presented four (4) generative language models for the unsupervised surface segmentation of isiXhosa.
- The two techniques identified were evaluated and the results presented in chapter 7: *Evaluation*.

8.3.2 Answering the Research Questions

The answer to Question 1, the existence of techniques that can readily advance the use of unsupervised machine learning in the morphological segmentation of Nguni languages, especially isiXhosa, is provided by the identification of the two surface segmentation techniques in chapter 2.

The answer to Question 2, whether the unsupervised segmentation of isiXhosa words could be improved using these techniques relative to a benchmark, can be found in chapter 7: *Evaluation*. In this chapter, the techniques implemented outperform the benchmark, Morfessor-Baseline, on the F1 Score. One of the techniques, branching entropy is comparable to the baseline. The overall performance is considered an improvement relative to the baseline.

The answer to Question 3, on the configuration of these techniques that provided the best performance, and the reasons thereof, is also found in chapter 7: *Evaluation*. In the branching entropy techniques, the Variation of Branching Entropy (VBE) technique provided the best performance because it captures predictive 9-grams. This is confirmed by the results that VBE capture all the predictive gain when using 9-grams as specified in equation (7-1). The heuristic guided system's best performing model, Independent Affixes, is also presented in this chapter and confirms that morphemic dependence is strong within an affix and weak among affixes as specified in Sect. 5.4: *Dependencies among affixes*. In addition, the tri-gram reflects the fact that

there are three dependency factors in the morphology of the language that dominate the structure, i.e. the subject concord, the object concord and the root. The other dependencies could be considered weak.

8.3.3 Achievement of the Aim

From the resolution of all of the research questions above, it is clear that the main research aim, has been met. Two systems that match the benchmark on accuracy, and outperform the benchmark on the F1 Score, have been provided.

8.4 Future work

In the future, more information-theoretic and statistical metrics could be evaluated on isiXhosa.

Regarding smoothing, there is a possibility of using the Good-Turing or Witten-Bell (Chen & Goodman, 1998) smoothing in branching entropy segmentation as they have shown promising results in Arabic (Darwish, 2002).

In addition, the techniques used in this thesis can be used for unsupervised segmentation of other Nguni languages and other, more resource-scarce languages.

There is also scope to see what guidance one can get from predictive information metrics for natural language processing, and what techniques would maximise the use of this information.

As identified in Sect. 4.2.4.1: *Inference in BE and VBE* the branching entropy measurements are not comparable close to the edges of a word. Another study could be an investigation into the effect of normalising the class branching entropy as well.

There is further scope to also establish heuristics and appropriate models for other South African languages.

Finally, XBES or XHMLS could be a pre-processing step to improve machine translation and web embeddings for South African languages.

8.5 Conclusions

In this study, unsupervised word segmentation techniques have been used to extend the field of unsupervised text segmentation.

It has been shown that branching entropy captures the entropy rate of a language as a time-series, and that variation of branching entropy captures the predictive gain.

It has also been shown that the use of generalised word morphology heuristics, when coupled with a character level language model, can provide state-of-the-art unsupervised word segmentation in isiXhosa.

BIBLIOGRAPHY

- Abdelali, A., Darwish, K., Durrani, N., and Mubarak, H., 2016. Farasa: A Fast and Furious Segmenter for Arabic. In *Proceedings of NAACL-HLT*. San Diego, California, USA: Association for Computational Linguistics, 11–16. Available at: <http://alt.qcri.org/tools/farasa/> [Accessed August 28, 2018].
- Aggarwal, C.C., 2018. *Neural Networks and Deep Learning*, Springer.
- Airoldi, E.M., 2007. Getting started in probabilistic graphical models. *PLoS Computational Biology*, 3(12), pp.2421–2425.
- Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., and Mohri, M., 2007. OpenFst: A General and Efficient Weighted Finite-State Transducer Library. In *Proceedings of the Twelfth International Conference on Implementation and Application of Automata, (CIAA 2007), Lecture Notes in Computer Science*. Prague, Czech Republic: Springer, 11–23. Available at: <http://www.openfst.org/twiki/pub/FST/FstBackground/ciaa.pdf> [Accessed June 4, 2018].
- Ando, R.K. and Lee, L., 2000. Mostly-Unsupervised Statistical Segmentation of Japanese: Applications to Kanji. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*.
- Andrew, G., 2006. A Hybrid Markov/Semi-Markov Conditional Random Field for Sequence Segmentation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006)*. Sydney, Australia: Association for Computational Linguistics, 465–472. Available at: <http://www.anthology.aclweb.org/W/W06/W06-1655.pdf> [Accessed October 29, 2017].
- Baker, J.K., 1975. The DRAGON System-An Overview. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 23(1), pp.24–29.
- Baum, L.E. and Petrie, T., 1966. Statistical inference for probabilistic functions for finite state Markov Chains. *The Annals of Mathematical Statistics*, 37(6), pp.1554–1563.
- Beesley, K., Buckwalter, T., and Newton, S., 1989. Two-level finite-state analysis of Arabic morphology. *Proceedings of the Seminar on Bilingual Computing in Arabic and English*, pp.6–7.
- Beesley, K.R., 1996. Arabic Finite-State Morphological Analysis and Generation. In *COLING '96 Proceedings of the 16th conference on Computational linguistics - Volume 1*. Copenhagen,

- Denmark: Association for Computational Linguistics, 89–94. Available at: http://delivery.acm.org/10.1145/1000000/992647/p89-beesley.pdf?ip=196.15.154.226&id=992647&acc=OPEN&key=4D4702B0C3E38B35.4D4702B0C3E38B35.4D4702B0C3E38B35.6D218144511F3437&__acm__=1526378281_c91f46632df2fc43c99f8d4f70d4a5c8 [Accessed May 15, 2018].
- Belkin, M. and Goldsmith, J., 2002. Using eigenvectors of the bigram graph to infer morpheme identity. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning*. Philadelphia, USA: Association for Computational Linguistics, 41–47.
- Bender, E.M., 2013. *Linguistic Fundamentals for Natural Language Processing: 100 Essentials from Morphology and Syntax*, Morgan & Claypool.
- Bengio, Y. and Senécal, J.-S., 2003. Quick Training of Probabilistic Neural Nets by Importance Sampling. In *AISTATS*. 1–9. Available at: http://www.iro.umontreal.ca/~lisa/pointeurs/senecal_aistats2003.pdf [Accessed September 17, 2018].
- Berger, A., Della Pietra, V.J., and Della Pietra, S.A., 1996. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1), pp.39–71.
- Bergmanis, T. and Goldwater, S., 2017. From Segmentation to Analyses: A Probabilistic Model for Unsupervised Morphology Induction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Valencia, Spain: Association for Computational Linguistics, 337–346. Available at: <http://www.aclweb.org/anthology/E17-1032> [Accessed August 31, 2018].
- Bialek, W., Nemenman, I., and Tishby, N., 2001a. Complexity through nonextensivity. *Physica A*, 302, pp.89–99. Available at: www.elsevier.com/locate/physa [Accessed July 21, 2020].
- Bialek, W., Nemenman, I., and Tishby, N., 2001b. Predictability, Complexity, and Learning. *Neural Computation*, 13, pp.2409–2463.
- Bishop, C.M., 2006. *Pattern Recognition and Machine Learning*, Springer.
- Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M.K., 1987. Occam’s razor. *Information processing letters*, 24(6), pp.377–380.
- Boffetta, G., Cencini, M., Falcioni, M., and Vulpiani, A., 2002. Predictability: a way to characterize complexity. *Physics Reports*, 356(6), pp.367–474.

- Booij, G., 2012. *The grammar of words: an introduction to linguistic morphology* Third Edit., Oxford University Press.
- van den Bosch, A. and Daelemans, W., 1999. Memory-based Morphological Analysis. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*. Maryland, USA: Association for Computational Linguistics, 285–292.
- Bosch, S., 2020. Computational morphology systems for Zulu – a comparison. *Nordic Journal of African Studies*, 29(3), pp.1–28.
- Bosch, S., Pretorius, L., and Fleisch, A., 2008. Experimental Bootstrapping of Morphological Analysers for Nguni Languages. *Nordic Journal of African Studies*, 17(2), pp.66–88.
- Botha, J.A. and Blunsom, P., 2013. Adaptor Grammars for Learning Non-Concatenative Morphology. In *Proceedings of the Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, 345–356. Available at: <http://www.aclweb.org/anthology/D/D13/D13-1034.pdf> [Accessed March 15, 2017].
- Boyce, W.B., 1844. *A Grammar of the Kaffir Language* Second Edi., London: Wesleyan Missionary Society.
- Brits, K., Pretorius, R., and van Huyssteen, G.B., 2005. Automatic lemmatization in Setswana: towards a prototype. *South African Journal of African Languages*, 1, pp.37–48.
- Brooks, S.P., 1998. Markov Chain Monte Carlo Method and Its Application. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 47(1), pp.69–100. Available at: <http://www.jstor.org/stable/2988428> [Accessed February 15, 2018].
- Buckwalter, T., 2002. Buckwalter Arabic Morphological Analyzer Version 1.0. *LDC Catalog No.: LDC2002L49*. Available at: <https://catalog.ldc.upenn.edu/LDC2002L49>.
- Buckwalter, T., 2004. Buckwalter Arabic Morphological Analyzer Version 2.0. *LDC Catalog No.: LDC2004L02*. Available at: <https://catalog.ldc.upenn.edu/docs/LDC2004L02/>.
- Çagrı Çöltekin, 2010. A Freely Available Morphological Analyzer for Turkish. In *LREC 2010*.
- Callison-Burch, C., Osborne, M., and Koehn, P., 2006. Re-evaluating the Role of BLEU in Machine Translation Research. In *1th Conference of the European Chapter of the Association for Computational Linguistics*. 249–256.
- Can, B. and Manandhar, S., 2014. Methods and Algorithms for Unsupervised Learning of

- Morphology. In *International Conference on Intelligent Text Processing and Computational Linguistics*. 177–205. Available at: <https://www.researchgate.net/publication/262450380> [Accessed August 27, 2018].
- Cao, Z., Li, S., and Ji, H., 2014. Joint Learning of Chinese Words, Terms and Keywords. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp.1774–1778.
- Chahuneau, V., Schlinger, E., Smith, N.A., and Dyer, C., 2013. Translating into Morphologically Rich Languages with Synthetic Phrases. In *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1677–1687. Available at: <http://repository.cmu.edu/lti> [Accessed March 24, 2017].
- Chavula, C. and Suleman, H., 2017. Morphological Cluster Induction of Bantu Words Using a Weighted Similarity Measure. In *Proceedings of SAICSIT '17*. Thaba Nchu, South Africa, 9. Available at: <http://pubs.cs.uct.ac.za/archive/00001225/01/morphological-cluster-induction-camera.pdf> [Accessed May 2, 2018].
- Chen, M., Chang, B., and Pei, W., 2014. A Joint Model for Unsupervised Chinese Word Segmentation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 854–863. Available at: <https://www.aclweb.org/anthology/D14-1092> [Accessed August 23, 2019].
- Chen, S., Xu, Y., and Chang, H., 2011. A Simple and Effective Unsupervised Word Segmentation Approach. *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, pp.866–871.
- Chen, S.F. and Goodman, J., 1998. *An Empirical Study of Smoothing Techniques for Language Modeling*, Cambridge, Massachusetts. Available at: <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-10-98.pdf> [Accessed October 16, 2018].
- Chiang, D., 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting of the ACL*. Ann Arbor: Association for Computational Linguistics, 263–270. Available at: http://delivery.acm.org/10.1145/1220000/1219873/p263-chiang.pdf?ip=41.114.39.135&id=1219873&acc=OPEN&key=4D4702B0C3E38B35.4D4702B0C3E38B35.4D4702B0C3E38B35.6D218144511F3437&CFID=748877195&CFTOKEN=84485195&__acm__=1491732224_dc913646e8e993e5f0888baaa [Accessed April 9,

2017].

- Chueh, C.-H., Wang, H.-M., and Chien, J.-T., 2006. A Maximum Entropy Approach for Semantic Language Modeling. *Computational Linguistics and Chinese Language Processing*, 11(1), pp.37–56. Available at: <http://aclweb.org/anthology/O06-2003> [Accessed January 30, 2019].
- Collobert, R., 2011. Deep Learning for Efficient Discriminative Parsing. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*. Fort Lauderdale, FL, USA, 224–232.
- Collobert, R., Weston, J., Karlen, M., Kavukcuoglu, K., Kuksa, P., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P., 2011. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12, pp.2493–2537.
- Collobert, R. and Weston, J., 2008. A unified architecture for natural language processing. In *Proceedings of the 25th international conference on Machine learning - ICML '08*. Helsinki, Finland, 160–167. Available at: <http://portal.acm.org/citation.cfm?id=1390177%5Cnhttp://portal.acm.org/citation.cfm?doid=1390156.1390177>.
- Comrie, B., Haspelmath, M., and Balthasar, B., 2008. *The Leipzig Glossing Rules Conventions for interlinear morpheme-by-morpheme glosses*,
- Cortes, C. and Vapnik, V., 1995. Support-Vector Networks. *Machine Learning*, 20, pp.273–297. Available at: <https://link.springer.com/content/pdf/10.1007/BF00994018.pdf> [Accessed May 22, 2018].
- Costa-Jussà, M.R. and Fonollosa, J.A.R., 2015. Latest trends in hybrid machine translation and its applications. *Computer Speech and Language*, 32(1), pp.3–10.
- Cotterell, R., Kirov, C., Sylak-Glassman, J., Yarowsky, D., Eisner, J., and Hulden, M., 2016. The SIGMORPHON 2016 Shared Task-Morphological Reinflection. In *Proceedings of the 14th Annual SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Berlin, Germany: Association for Computational Linguistics, 10–22. Available at: <http://ryancotterell>. [Accessed April 5, 2020].
- Cotterell, R. and Duh, K., 2017. Low-Resource Named Entity Recognition with Cross-Lingual, Character-Level Neural Conditional Random Fields. In *Proceedings of the 8th International Joint Conference on Natural Language Processing*. Taipei, Taiwan: AFNLP, 91–96.

- Cotterell, R., Vieira, T., and Schütze, H., 2016. A joint model of orthography and morphological segmentation. In *Proceedings of NAACL-HLT 2016*. San Diego, California, USA: Association for Computational Linguistics, 664–669. Available at: <http://www.aclweb.org/anthology/N16-1080> [Accessed September 3, 2018].
- Cover, T.M. and Thomas, J.A., 1991. *Elements of Information Theory*, New York: John Wiley & Sons.
- Creutz, M. and Lagus, K., 2005. Inducing the morphological lexicon of a natural language from unannotated text. In *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR'05)*. Espoo, Finland, 106–113.
- Creutz, M. and Lagus, K., 2002. Unsupervised Discovery of Morphemes. In *Proceedings of the 6th Workshop of the ACL Special Interest Group in Computational Phonology*. Philadelphia, USA, 21–30.
- Creutz, M. and Lagus, K., 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(1), p.Article 3. Available at: <http://eprints.pascal-network.org/archive/00002394/>.
- Crutchfield, J.P. and Feldman, D.P., 2003. Regularities unseen, randomness observed: Levels of entropy convergence. *Chaos*, 13(1), pp.25–54.
- Culotta, A. and McCallum, A., 2004. Confidence Estimation for Information Extraction. In *Proceedings of HLT-NAACL 2004: Short Papers*. Boston, Massachusetts, USA: Association for Computational Linguistics, 109–112. Available at: <http://www.aclweb.org/anthology/N04-4028> [Accessed February 27, 2018].
- Curtis, C.M., 2014. A finite-state morphological analyzer for a Lakota precision grammar. In *Proc. LREC. 2014*. 541–544.
- Darwish, K., 2002. Building a Shallow Arabic Morphological Analyzer in One Day. In *Proceedings of the ACL-02 workshop on Computational approaches to Semitic languages*. Association for Computational Linguistics, 1–8. Available at: <http://www.aclweb.org/anthology/W02-0506> [Accessed March 14, 2018].
- David Yarowsky and Richard Wicentowski, 2000. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. 207–216.

- Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., and Harshman, R., 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science* Sep, 41(6), pp.391–407.
- Déjean, H., 1998. Morphemes as Necessary Concept for Structures Discovery from Untagged Corpora. In D.M.W. Powers, ed. *NeMLaP3/CoNLL98 Workshop on Paradigms and Grounding in Natural Language Learning*. Adelaide: ACL, 295–299.
- Demšar, J., 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. *The Journal of Machine Learning Research*, 7, pp.1–30.
- Diab, M., Hacıoglu, K., and Jurafsky, D., 2004. Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. In *Proceedings of HLT-NAACL 2004: Short papers*. Association for Computational Linguistics, 149–152. Available at: <https://aclanthology.info/pdf/N/N04/N04-4038.pdf> [Accessed May 21, 2018].
- Dreyer, M. and Eisner, J., 2011. Discovering Morphological Paradigms from Plain Text Using a Dirichlet Process Mixture Model. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK: Association for Computational Linguistics, 616–627. Available at: <https://aclanthology.info/pdf/D/D11/D11-1057.pdf> [Accessed September 7, 2018].
- Dreyer, M. and Eisner, J., 2009. Graphical Models over Multiple Strings. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore: ACL and AFNLP, 101–110. Available at: <https://www.aclweb.org/anthology/D09-1011> [Accessed September 7, 2018].
- Durrett, G. and Klein, D., 2015. Neural CRF Parsing. *arXiv preprint rXiv:1507.03641v1*. Available at: <http://nlp.cs.berkeley.edu> [Accessed April 8, 2020].
- Džeroski, S. and Erjavec, T., 1998. Inductive Learning of Multilingual Morphology. *Elektrotehniški Vestnik/Electrotechnical Review*, 65(5), pp.296–302.
- Eiselen, R. and Puttkammer, M.J., 2014. Developing Text Resources for Ten South African Languages. In N. Calzolari et al., eds. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), 3698–3703.
- El-Kishky, A., Xu, F., Zhang, A., Macke, S., and Han, J., 2018. Entropy-Based Subword Mining with an Application to Word Embeddings. In *Proceedings of the Second Workshop on*

- Subword/Character Level Models*. New Orleans, Louisiana, USA: Association for Computational Linguistics, 12–21.
- El-Kishky, A., Xu, F., Zhang, A., and Han, J., 2019. Parsimonious Morpheme Segmentation with an Application to Enriching Word Embeddings. *Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019*, (August), pp.64–73.
- Erdmann, A., Khalifa, S., Oudah, M., Habash, N., and Bouamor, H., 2019. A Little Linguistics Goes a Long Way: Unsupervised Segmentation with Limited Language Specific Guidance. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Florence, Italy: Association for Computational Linguistics, 113–124.
- Erjavec, T. and Džeroski, S., 2004. Machine learning of morphosyntactic structure: Lemmatizing unknown Slovene words. *Applied Artificial Intelligence*, 18, pp.17–41.
- Fano, R.M., 1961. *Transmission of information*, MIT Press.
- Feldman, D.P. and Crutchfield, J.P., 2004. Synchronizing to Periodicity: The Transient Information and Synchronization Time of Periodic Sequences. *Advances in Complex Systems*, 7(3–4), pp.329–355.
- Feng, H., Chen, K., Deng, X., and Zheng, W., 2004. Accessor Variety Criteria for Chinese Word Extraction. *Computational Linguistics*, 30(1), pp.75–93. Available at: <https://www.mitpressjournals.org/doi/pdfplus/10.1162/089120104773633394> [Accessed August 26, 2018].
- Feng, H., Chen, K., Kit, C., and Deng, X., 2004. Unsupervised Segmentation of Chinese Corpus Using Accessor Variety. In *International Conference on Natural Language Processing*. Springer, Berlin, Heidelberg, 694–703. Available at: <https://pdfs.semanticscholar.org/6361/00aa5d12e96c13a82d626224721ef82410f7.pdf> [Accessed February 19, 2018].
- Ferguson, T.S., 1973. A Bayesian Analysis of Some nonparametric problems. *The Annals of Statistics*, 1(2), pp.209–230. Available at: http://projecteuclid.org/download/pdf_1/euclid.aos/1176342360 [Accessed March 22, 2017].
- Fernández, N., Maldonado, C., and Gershenson, C., 2014. Information Measures of Complexity, Emergence, Self-organization, Homeostasis, and Autopoiesis. In M. Prokopenko, ed. *Guided Self-Organization: Inception, Emergence, Complexity and Computation 9*. Berlin Heidelberg: Springer-Verlag, 19–51. Available at:

<https://www.researchgate.net/publication/236124438> [Accessed June 15, 2020].

Fine, S., Singer, Y., and Tishby, N., 1998. The Hierarchical Hidden Markov Model: Analysis and Applications. *Machine Learning*, 32(1), pp.41–62.

Gahbiche-Braham, S., Ene Bonneau-Maynard, H., Lavergne, T., and Yvon, F., 2012. Joint Segmentation and POS Tagging for Arabic Using a CRF-based Classifier. *Lrec*, (May 2012), pp.2107–2113. Available at: http://www.lrec-conf.org/proceedings/lrec2012/pdf/855_Paper.pdf.

Gao, J., Wu, A., Li, M., Huang, C.-N., Li, H., Xia, X., and Qin, H., 2004. Adaptive Chinese Word Segmentation. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics., 462. Available at: <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/acws.acl04.pdf> [Accessed March 7, 2018].

Gaussier, E., 1999. Unsupervised learning of derivational morphology from inflectional lexicons. In *Proceedings of ACL'99 Workshop: Unsupervised Learning in Natural Language Processing*. 24–30.

Geman, S. and Geman, D., 1984. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 6(6). Available at: <https://pdfs.semanticscholar.org/62c3/4c8a8d8b82a9c466c35cda5e4837c17d9ccb.pdf> [Accessed February 15, 2018].

Geurts, P., Ernst, D., and Wehenkel, L., 2006. Extremely randomized trees. *Mach Learning*, 63(1), pp.3–42. Available at: <https://link.springer.com/content/pdf/10.1007/s10994-006-6226-1.pdf> [Accessed September 17, 2018].

Goldsmith, J., 2000. Linguistica: An Automatic Morphological Analyzer. *Papers from the 36th Meeting of the Chicago Linguistics Society*, 1.

Goldsmith, J., 2001. Unsupervised Learning of the Morphology of a Natural Language. *Computational Linguistics*, 27(2), pp.153–198. Available at: <http://www.mitpressjournals.org/doi/abs/10.1162/089120101750300490>.

Goldwater, S., Griffiths, T.L., and Johnson, M., 2009. A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112, pp.21–54. Available at: <https://cocosci.berkeley.edu/tom/papers/wordseg3.pdf> [Accessed February 13, 2018].

- Goldwater, S., Griffiths, T.L., and Johnson, M., 2006. Contextual Dependencies in Unsupervised Word Segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Sydney, Australia: Association for Computational Linguistics, 673–680. Available at: http://www.research.ed.ac.uk/portal/files/7693581/P06_1085.pdf [Accessed March 22, 2017].
- Goldwater, S., Griffiths, T.L., and Johnson, M., 2005. Interpolating Between Types and Tokens by Estimating Power-Law Generators. In *Advances in neural information processing systems*. 459–466.
- Golénia, B., Spiegler, S., and Flach, P., 2009. UNGRADE: UNSupervised GRAph DEcomposition. In *Working Notes for the CLEF 2009 Workshop, CLEF 2009*. Corfu, Greece. Available at: http://www.clef-campaign.org/2009/working_notes/morpho-papers/golenia-paperCLEF2009.pdf.
- Golénia, B., Spiegler, S., and Flach, P.A., 2010. Unsupervised morpheme discovery with UNGRADE. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer, Berlin, Heidelberg, 633–640. Available at: <http://www.cs.bris.ac.uk/Publications/Papers/2001221.pdf> [Accessed August 2, 2017].
- Gong, J., Chen, X., Gui, T., and Qiu, X., 2019. Switch-LSTMs for Multi-Criteria Chinese Word Segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence, 6457–6464.
- Grassberger, P., 1986. Toward a Quantitative Theory of Self-Generated Complexity. *International Journal of Theoretical Physics*, 25(9), pp.907–938.
- Graves, A. and Schmidhuber, J., 2005. Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. *Neural Networks*, 18(5–6), pp.602–610. Available at: <http://wwwknoll.informatik.tu-muenchen.de/pub/Main/Publications/Graves2005b.pdf> [Accessed August 7, 2018].
- Grönroos, S.-A., Virpioja, S., Smit, P., and Kurimo, M., 2014. Morfessor FlatCat: An HMM-Based Method for Unsupervised and Semi-Supervised Learning of Morphology. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics (2014)*. 1177–1185.
- Habash, N., 2007. Arabic Morphological Representations for Machine Translation. In A. Soudi, A.

- van den Bosch, & G. Neumann, eds. *Arabic Computational Morphology: Knowledge-based and Empirical Methods, Text, Speech, and Language Technology*. Dordrech: Springer, 263–285. Available at: http://www.cs.cmu.edu/afs/cs.cmu.edu/project/cmt-40/Nice/Transfer/Arabic/chapter14_Nizar_DRAFT.pdf [Accessed May 22, 2018].
- Habash, N. and Rambow, O., 2005. Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Morristown, NJ, USA: Association for Computational Linguistics, 573–580. Available at: <http://www1.cs.columbia.edu/~rambow/papers/habash-rambow-2005a.pdf> [Accessed July 31, 2017].
- Hammarström, H. and Borin, L., 2011. Unsupervised learning of morphology. *Computational Linguistics*, 37(2), pp.309–350.
- Harris, Z.S., 1955. From Phoneme to Morpheme. *Language*, 31(2), pp.190–222. Available at: <https://www-jstor-org.nwulib.nwu.ac.za/stable/pdf/411036.pdf> [Accessed August 26, 2018].
- Harris, Z.S., 1942. Morpheme Alternants in Linguistic Analysis. *Language*, 18(3), pp.169–180. Available at: <http://anthropology.uwo.ca/faculty/creider/9640a/Harris-MorphemeAlternants.pdf> [Accessed May 26, 2018].
- Hastie, T., Tibshirani, R., and Friedman, J., 2009. *The elements of statistical learning* 2nd ed., Springer.
- Hatori, J., Matsuzaki, T., Miyao, Y., and Tsujii, J., 2012. Incremental Joint Approach to Word Segmentation, POS Tagging, and Dependency Parsing in Chinese. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. Jeju, Republic of Korea: Association for Computational Linguistics, 1045–1053. Available at: <http://www.aclweb.org/anthology/P12-1110> [Accessed April 5, 2018].
- Heafield, K., 2011. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the 6th Workshop on Statistical Machine Translation*. Edinburgh, Scotland, UK: Association for Computational Linguistics, 187–197. Available at: <http://sites.google.com/site/murmurhash/> [Accessed October 7, 2018].
- Heafield, K., Pouzyrevsky, I., Clark, J.H., and Koehn, P., 2013. Scalable Modified Kneser-Ney Language Model Estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Sofia, Bulgaria: Association for Computational Linguistics, 690–696.

- Heymann, J., Walter, O., Haeb-Umbach, R., and Raj, B., 2013. Unsupervised word segmentation from noisy input. In *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 458–463. Available at: <https://pdfs.semanticscholar.org/eacc/53f1266eac335a0042b2abe2d955f3a8d8ac.pdf> [Accessed January 20, 2018].
- Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. *Neural Computation*, 9(8), pp.1735–1780. Available at: http://web.eecs.utk.edu/~itamar/courses/ECE-692/Bobby_paper1.pdf [Accessed August 9, 2017].
- Huang, J.H. and Powers, D., 2003. Chinese Word Segmentation Based on Contextual Entropy. In *Proceedings of the 17th Pacific Asia Conference on Language, Information and Computation*. 152–158.
- Hurskainen, A., 2004. Swahili Language Manager: A Storehouse for Developing Multiple Computational Applications. *Nordic Journal of African Studies*, 13(3), pp.363–397. Available at: www.csc.fi [Accessed June 19, 2020].
- Ince, R.A.A., 2017. Measuring multivariate redundant information with pointwise common change in surprisal. *Entropy*, 19(7), p.318.
- Itti, L. and Baldi, P., 2009. Bayesian surprise attracts human attention. *Vision Research*, 49(10), pp.1295–1306.
- Jaynes, E.T., 1957. Information Theory and Statistical Mechanics. *Physical Review*, 106(4), pp.620–630.
- Jelinek, F., 1976. Speech Recognition by Statistical Methods. *PROCEEDINGS OF THE IEEE*, 64(4), pp.532–556. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.473.9761&rep=rep1&type=pdf> [Accessed May 24, 2018].
- Jensen, J.L.W.V., 1906. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1), pp.175–193.
- Jin, Z. and Tanaka-Ishii, K., 2006. Unsupervised Segmentation of Chinese Text by Use of Branching Entropy. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*. Sydney, Australia: Association for Computational Linguistics, 428–435. Available at: <http://www.aclweb.org/anthology/P06-2056> [Accessed February 19, 2018].

- Johnson, M., 2008a. Unsupervised word segmentation for Sesotho using Adaptor Grammars. In *Proceedings of the Tenth Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*. Columbus, Ohio, USA: Association for Computational Linguistics, 20–27.
- Johnson, M., 2008b. Using adaptor grammars to identify synergies in the unsupervised acquisition of linguistic structure. *ACL-08: HLT*, pp.398–406. Available at: <https://www.aclweb.org/anthology/P/P08/P08-1046.pdf> [Accessed November 30, 2017].
- Johnson, M., Griffiths, T.L., and Goldwater, S., 2007. Adaptor Grammars: A Framework for Specifying Compositional Nonparametric Bayesian Models. *Advances in neural information processing systems*, 19, p.641. Available at: <https://papers.nips.cc/paper/3101-adaptor-grammars-a-framework-for-specifying-compositional-nonparametric-bayesian-models.pdf> [Accessed March 22, 2017].
- Jung, S.-Y., Park, Y.C., Choi, K.-S., and Kim, Y., 1996. Markov random field based English Part-Of-Speech tagging system. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 236–242.
- Jurafsky, D. and Martin, J., 2017. *Speech and language processing* 3rd ed. dr., Available at: https://web.stanford.edu/~jurafsky/slp3/edbook_oct162019.pdf.
- Jurafsky, D. and Martin, J.H., 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.*, Upper Saddle River, New Jersey: Pearson Prentice Hall.
- Kakade, S., Teh, Y.W., and Roweis, S.T., 2002. An alternate objective function for Markovian fields. *Proceedings of the Nineteenth International Conference on Machine Learning*, pp.275–282. Available at: <http://www.cs.berkeley.edu/~ywtteh/research/newcost/icml2002.pdf> [Accessed March 11, 2020].
- Kann, K., Cotterell, R., and Schütze, H., 2016. Neural Morphological Analysis: Encoding-Decoding Canonical Segments. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas, USA: Association for Computational Linguistics, 961–967.
- Kann, K. and Schütze, H., 2016. Single-Model Encoder-Decoder with Explicit Morphological Representation for Reinflection. In *Proceedings of ACL*.

- Karttunen, L., 1993. *Finite-state lexicon compiler. Technical Report ISTL-NLTT-1993-04-02*, Palo Alto, CA.
- Kempe, A., 1999. Experiments in Unsupervised Entropy-Based Corpus Segmentation. In *EACL 1999: CoNLL-99 Computational Natural Language Learning*. 7–13.
- Khelifa, M.O.M., Belkasmi, M., Abdellah, Y., and Elhadj, Y.O., 2017. An Accurate HSMM-based System for Arabic Phoneme Recognition. In *017 Ninth International Conference on Advanced Computational Intelligence (ICACI)*. Doha, Qatar: IEEE, 211–216. Available at: <https://www.researchgate.net/publication/313384708> [Accessed March 5, 2020].
- Kim, Y., Jernite, Y., Sontag, D., and Rush, A.M., 2016. Character-Aware Neural Language Models. In *AAAI*. Available at: <http://arxiv.org/abs/1508.06615>.
- Kit, C., 1998. A Goodness Measure for Phrase Learning via Compression with the MDL Principle. In Ivana Kruijff-Korbyova, ed. *Proceedings of the Third ESSLLI Student Session*. 175–187. Available at: <https://pdfs.semanticscholar.org/120d/b0372be64b0c2a52ff836932d98937582674.pdf> [Accessed February 26, 2018].
- Kit, C. and Wilks, Y., 1999. Unsupervised Learning of Word Boundary with Description Length Gain. In *EACL 1999: CoNLL-99 Computational Natural Language Learning*. Available at: <http://www.aclweb.org/anthology/W99-0701> [Accessed February 19, 2018].
- Kohonen, O., Virpioja, S., and Lagus, K., 2010. Semi-supervised learning of concatenative morphology. In *Proceedings of the 11th Meeting of the ACL-SIGMORPHON, ACL 2010*. Uppsala, Sweden: Association for Computational Linguistics, 78–86.
- Kosch, I.M., 2006. *Topics in Morphology in the African Language Context*, Pretoria: Unisa Press.
- Koskenniemi, K., 1984. A general computational model for word-form recognition and production. In *Proceedings of the 10th international conference on Computational linguistics* -. 178–181.
- Kotzé, G. and Wolff, F., 2015. Syllabification and parameter optimisation in Zulu to English machine translation. *South African Computer Journal*, 57(December), pp.1–23.
- Krantz, J., Dulin, M., and De Palma, P., 2019. Language-Agnostic Syllabification with Neural Sequence Labeling. *arXiv preprint rXiv:1909.13362v1*.
- Kullback, S. and Leibler, R.A., 1951. On Information and Sufficiency. *The annals of mathematical statistics*, 22(1), pp.79–86. Available at:

https://projecteuclid.org/download/pdf_1/euclid.aoms/1177729694 [Accessed September 19, 2018].

- Lafferty, J., McCallum, A., and Pereira, F.C.N., 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the 18th International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc, 282–289.
- Lee, K.A., Horiuchi, R., and Hong, C.S., 2003. PLATFORM-INDEPENDENT AUTOMATED MACHINE TRANSLATION SYSTEM.
- Lee, Y.-S., Papineni, K., Roukos, S., Emam, O., and Hassan, H., 2003. Language Model Based Arabic Word Segmentation. In *Proceedings of the 41st Annual Meeting of the ACL*. Sapporo, Japan: Association for Computational Linguistics, 399–406.
- Lee, Y.K., Haghighi, A., and Barzilay, R., 2011. Modeling syntactic context improves morphological segmentation. In *CoNLL 2011 - Fifteenth Conference on Computational Natural Language Learning, Proceedings of the Conference*. Portland, Oregon, USA: Association for Computational Linguistics, 1–9. Available at: <http://groups.csail.mit.edu/rbg/code/morphsyn/>. [Accessed September 7, 2018].
- Levenshtein, V.I., 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics-Doklady*, 10(8), pp.707–710.
- Lignos, C., Chan, E., Marcus, M.P., and Yang, C., 2009. A rule-based unsupervised morphology learning framework. *CEUR Workshop Proceedings*, 1175(January 2009).
- Lignos, C., 2010. Learning from Unseen Data. In Mikko Kurimo, Sami Virpioja, & Ville T. Turunen, eds. *Proceedings of the Morpho Challenge 2010 Workshop*. Aalto University School of Science and Technology, 35–38. Available at: https://www.researchgate.net/profile/Constantine_Lignos/publication/266465284_Learning_from_Unseen_Data/links/54b66b530cf2bd04be320b0f.pdf [Accessed March 22, 2017].
- Lin, C.-C., Ammar, W., Dyer, C., and Levin, L., 2015. Unsupervised POS Induction with Word Embeddings. In *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL (2015)*. 1311–1316. Available at: <https://arxiv.org/pdf/1503.06760.pdf> [Accessed April 7, 2017].
- Ling, W., Luís, T., Marujo, L., Astudillo, R.F., Amir, S., Dyer, C., Black, A.W., and Trancoso, I., 2015. Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural*

- Language Processing* (2015). Association for Computational Linguistics, 1520–1530.
- Liu, Y., Che, W., Qin, B., and Liu, T., 2020. Exploring Segment Representations for Neural Semi-Markov Conditional Random Fields. *IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, 28, p.2020. Available at: <https://www.ieee.org/publications/rights/index.html> [Accessed April 8, 2020].
- Loubser, M. and Puttkammer, M.J., 2020. Viability of Neural Networks for Core Technologies for Resource-Scarce Languages. *Information*, 11(41), p.17.
- Louw, J.A., Finlayson, R., and Satyo, S.C., 1984. *Xhosa Guide 3 for XHA100-F*, Pretoria: University of South Africa.
- Mackay, D.J.C., 1995. A Hierarchical Dirichlet Language Model. *Natural Language Engineering*, 1(3), pp.289–308. Available at: papers3://publication/uuid/606D40A3-375D-4B7B-8D56-FD1C91699D7B.
- Magistry, P., 2012. Segmentation non supervisée : le cas du mandarin. In *Actes de la conférence conjointe JEP-TALN-RECITAL*. Grenoble: ATALA & AFCEP, 1–13. Available at: <http://www.aclweb.org/anthology/F12-3001> [Accessed September 6, 2018].
- Magistry, P., 2013. *Unsupervised Word Segmentation and Wordhood Assessment: The case for Mandarin Chinese*. Université Paris Diderot.
- Magistry, P. and Sagot, B., 2013. Can MDL Improve Unsupervised Chinese Word Segmentation?. In *Sixth International Joint Conference on Natural Language Processing: Sighan workshop*. Nagoya, Japan, 2. Available at: <https://hal.inria.fr/hal-00876389> [Accessed September 5, 2018].
- Magistry, P. and Sagot, B., 2012. Unsupervised Word Segmentation: the case for Mandarin Chinese. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. Jeju, Republic of Korea: Association for Computational Linguistics, 383–387. Available at: <http://www.aclweb.org/anthology/P12-2075> [Accessed September 5, 2018].
- Manning, C.D., Raghavan, P., and Schütze, H., 2009. *An Introduction to Information Retrieval* Online edi., Cambridge, England: Cambridge University Press.
- Manning, C.D. and Schütze, H., 1999. *Foundations of Statistical Natural Language Processing*, MIT Press.
- McCallum, A., Freitag, D., and Pereira, F., 2000. Maximum Entropy Markov Models for

- Information Extraction and Segmentation. In *ICML*. 591–598. Available at: <http://l2r.cs.uiuc.edu/~danr/Teaching/CS546-12/Papers/McCallum-MEMM-01.pdf> [Accessed July 27, 2017].
- McCowan, I., Moore, D., and Dines, J., 2005. *On the Use of Information Retrieval Measures for Speech Recognition Evaluation*, Martigny, Swoi. Available at: <http://www.idiap.ch/ftp/reports/2004/rr04-73.pdf> [Accessed July 28, 2017].
- McLaren, J., 1948. *A Xhosa Grammar* Third. G. . Welsh, ed., London; Cape Town: Longmans, Green.
- Meinhof, C., 1932. *Introduction to the Phonology of the Bantu languages.*, Berlin: Dietrich Reimer/Erns Vohsen.
- Méndez-Cruz, C.-F., Torres-Moreno, J.-M., Medina-Urrea, A., and Sierra, G., 2012. Extrinsic Evaluation on Automatic Summarization Tasks: Testing Affixality Measurements for Statistical Word Stemming. In *Mexican International Conference on Artificial Intelligence*. Springer Berlin Heidelberg, 46–57.
- Méndez-Cruz, C.-F., Medina-Urrea, A., and Sierra, G., 2016. *Unsupervised morphological segmentation based on affixality measurements*,
- Mikolov, T., Corrado, G., Chen, K., and Dean, J., 2013. Efficient Estimation of Word Representations in Vector Space. *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, pp.1–12. Available at: <http://arxiv.org/pdf/1301.3781v3.pdf>.
- Mikolov, T., Yih, W., and Zweig, G., 2013. Linguistic regularities in continuous space word representations. In *Proceedings of NAACL-HLT*. Atlanta, Georgia, USA: Association for Computational Linguistics, 746–751. Available at: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Linguistic+Regularities+in+Continuous+Space+Word+Representations#0%5Cnhttps://www.aclweb.org/anthology/N13/N13-1090.pdf>.
- Mini, B.M. and Tshabe, S.L.,Eds., 2003. *The Greater Dictionary of isiXhosa: Volume 2 (K to P)*, Alice: University of Fort Hare.
- Minkov, E., Toutanova, K., and Suzuki, H., 2007. Generating Complex Morphology for Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, 128–135.

- Mitchell, T.M., 1997. *Machine Learning*, WCB/McGraw-Hill.
- Mncube, F.S.M., 1955. *Xhosa manual*, Johannesburg: Juta & Company, Limited.
- Mochihashi, D., Yamada, T., and Ueda, N., 2009. Bayesian Unsupervised Word Segmentation with Nested Pitman-Yor Language Modeling. In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*. Suntec, Singapore: ACL and AFNLP, 100–108. Available at: <http://anthology.aclweb.org/P/P09/P09-1012.pdf> [Accessed November 13, 2017].
- More, A. and Tsarfaty, R., 2016. Data-Driven Morphological Analysis and Disambiguation for Morphologically Rich Languages and Universal Dependencies. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan, 337–348. Available at: <http://www.aclweb.org/anthology/C16-1033> [Accessed September 17, 2018].
- Morris, A.C., Maier, V., and Green, P., 2004. From WER and RIL to MER and WIL: Improved evaluation measures for connected speech recognition. In *8th International Conference on Spoken Language Processing, ICSLP 2004*. 2765–2768. Available at: https://d5da38cc-a-62cb3a1a-s-sites.googlegroups.com/site/andrewcameronmorris/Home/publications/icslp_wip.pdf?attachauth=ANoY7cobi6_zfOUiM9WP1Fx52ts37N_2CriE23Xd9o3h5isDr8uuPD-30Dh2SmbdgEaM0eilypWPo0TXUuS2J9a5sPVdo-W0OJzLD-HYlrBnhW37qFWNGjJYJrRqvA5uDOO2bf [Accessed July 28, 2017].
- Muggleton, S., 2019. Stochastic Logic Programs. In *Introduction to Statistical Relational Learning*. 254–264.
- Muggleton, S. and de Raedt, L., 1994. Inductive Logic Programming: Theory and methods. *The Journal of Logic Programming*, 19–20(SUPPL. 1), pp.629–679.
- Muller, T. and Schutze, H., 2015. Robust Morphological Tagging with Word Representations. In *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*. Denver, Colorado: Association for Computational Linguistics, 526–536. Available at: http://www.aclweb.org/website/old_anthology/N/N15/N15-1055.pdf [Accessed April 7, 2017].
- Murphy, Kevin Patrick, 2002. *Dynamic Bayesian Networks: Representation, Inference and Learning*. Phd-Thesis. University of California, Berkley. Available at: <http://www.cs.ubc.ca/~murphyk/Thesis/thesis.pdf>.

- Murphy, Kevin P, 2002. *Hidden semi-Markov models (HSMMs)*, Available at: www.ai.mit.edu/~murphyk [Accessed April 16, 2018].
- Murphy, K.P., 2012. *Machine Learning: A probabilistic perspective*, Cambridge, MA,USA: MIT Press.
- Mzamo, L., 2015. *Evaluation of the performance of a machine learning lemmatiser for isiXhosa*. M.Eng Thesis. North-West University.
- Mzamo, L., Helberg, A., and Bosch, S., 2015. Introducing XGL-a lexicalised probabilistic graphical lemmatiser for isiXhosa. In *Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech) 2015*. Port Elizabeth, South Africa: IEEE, 142–147.
- Narasimhan, K., Barzilay, R., and Jaakkola, T., 2015. An Unsupervised Method for Uncovering Morphological Chains. *Transactions of the Association for Computational Linguistics*, 3, pp.157–167. Available at: <https://github.com/> [Accessed August 31, 2018].
- Neubig, G., Mimura, M., Mori, S., and Kawahara, T., 2012. Bayesian Learning of a Language Model from Continuous Speech. *IEICE TRANSACTIONS on Information and Systems*, 95(2), pp.614–625.
- Neubig, G., Mimura, M., Mori, S., and Kawahara, T., 2010. Learning a Language Model from Continuous Speech. In *Interspeech 2010*. Makuhari, Chiba, Japan, 1053–1056. Available at: <http://ai2-s2-pdfs.s3.amazonaws.com/931a/103258c96a1230dc5c7e38a1cd0b095b9d62.pdf> [Accessed November 13, 2017].
- Neuvel, S. and Fulop, S.A., 2002. Unsupervised Learning of Morphology Without Morphemes. In *Morphological and Phonological Learning: Proceedings of the 6th Workshop of the ACL Special Interest Group in Computational Phonology (SIGPHON)*. Philadelphia, USA: Association for Computational Linguistics, 31–40.
- van Niekerk, D.R., 2014. Exploring unsupervised word segmentation for machine translation in the South African context. In M. J. Puttkammer & R. Eiselen, eds. *Proceedings of the 2014 PRASA, RobMech and AfLaT International Joint Symposium*. Cape Town, 202–206.
- Nogwina, M., 2016. *Development of a Stemmer for the IsiXhosa Language*. University of Fort Hare: MSc. Dissertation. Available at: http://libdspace.ufh.ac.za/bitstream/handle/20.500.11837/221/M_Sc_%28Computer

- Science%29 - NOGWINA%2C M.pdf?sequence=1&isAllowed=y [Accessed September 18, 2017].
- Pahl, H.W., 1982. *IsiXhosa* D. M. Ntusi & S. M. Burns-Ncamashe, eds., King Williams Town: Educum Publishers.
- Pahl, H.W., Pienaar, A.M., and Ndungane, T.A., Eds., 1989. *The Greater Dictionary of Xhosa: Volume 3 (Q to Z)*, Alice: University of Fort Hare.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J., 2002. BLEU: a Method for Automatic Evaluation of Machine Translation A. H. Adams, G. W. Austin, & M. Taylor, eds. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, pages(July), pp.311–318. Available at: <http://dl.acm.org/citation.cfm?id=1073083.1073135>.
- de Pauw, G. and de Schryver, G., 2008. Improving the Computational Morphological Analysis of a Swahili Corpus for Lexicographic Purposes. *Lexikos*, 18(July), pp.303–318.
- De Pauw, G. and Wagacha, P.W., 2007. Bootstrapping Morphological Analysis of Gĩkũyũ Using Unsupervised Maximum Entropy Learning. In *INTERSPEECH 2007*. Antwerp, Belgium, 1517–1520. Available at: <https://pdfs.semanticscholar.org/e7d8/d953bbcafd70b92f9d68023e143beb76123f.pdf> [Accessed September 18, 2017].
- Peng, F., Feng, F., and McCallum, A., 2004. Chinese Segmentation and New Word Detection using Conditional Random Fields. In *Proceedings of the 20th international conference on Computational Linguistics*. 562. Available at: <http://www.aclweb.org/anthology/C04-1081> [Accessed February 26, 2018].
- Peng, N. and Dredze, M., 2016. Multi-task Multi-domain Representation Learning for Sequence Tagging. *arXivpreprint arXiv:1608.02689v1*. Available at: <https://www.researchgate.net/publication/306186061> [Accessed April 8, 2020].
- Pickhardt, R., Gottron, T., Körner, M., Staab, S., Wagner, P.G., and Speicher, T., 2014. A Generalized Language Model as the Combination of Skipped n-grams and Modified Kneser-Ney Smoothing. *arXiv:1404.3377*, (Apr 13). Available at: <https://arxiv.org/pdf/1404.3377.pdf> [Accessed November 14, 2018].
- Della Pietra, S., Della Pietra, V., and Lafferty, J., 1995. *Inducing features of random fields*, Pittsburgh. Available at: <https://arxiv.org/pdf/cmp-lg/9506014.pdf> [Accessed January 30, 2019].

- Pitman, J. and Yor, M., 1997. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25(2), pp.855–900. Available at: http://projecteuclid.org/download/pdf_1/euclid.aop/1024404422 [Accessed March 17, 2017].
- Poon, H., Cherry, C., and Toutanova, K., 2009. Unsupervised Morphological Segmentation with Log-Linear Models. In *NAACL '09 Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Boulder, Colorado, 209–217.
- Pretorius, L. and Bosch, S., 2018. ZulMorph: Finite state morphological analyser for Zulu. Available at: <https://portal.sadilar.org/FiniteState/demo/zulmorph/>.
- Pretorius, L. and Bosch, S.E., 2005. Finite-State Computational Morphology: An Analyzer Prototype For Zulu. *Machine Translation*, 18(3), pp.195–216.
- Prokopenko, M., Boschetti, F., and Ryan, A.J., 2009. An information-theoretic primer on complexity, self-organization, and emergence. *Complexity*, 15(1), pp.11–28. Available at: <http://doi.wiley.com/10.1002/cplx.20249> [Accessed June 15, 2020].
- Pylkkönen, J. and Kurimo, M., 2004. Duration modeling techniques for continuous speech recognition. In Eighth International Conference on Spoken Language Processing. Available at: <https://www.researchgate.net/publication/221489959> [Accessed March 5, 2020].
- Quasthoff, U., Bosch, S., and Goldhahn, D., 2014. Morphological analysis for less-resourced languages: Maximum Affix Overlap applied to Zulu. In *CCURL 2014: Collaboration and Computing for Under-Resourced Languages in the Linked Open Data Era*. 52. Available at: <http://rma.nwu.ac.za/> [Accessed December 8, 2020].
- Quasthoff, U., Goldhahn, D., and Bosch, S., 2016. Morphology Learning for Zulu. In C. Soria et al., eds. *Proceedings of the LREC 2016 Workshop "CCURL 2016 – Towards an Alliance for Digital Language Diversity"*. 89–95.
- Rabiner, L.R., 1989. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), pp.257–286.
- Rasinger, S.M., 2013. *Quantitative Research in Linguistics: An Introduction* 2nd Editio., London: Bloomsbury Academic.
- Ratnaparkhi, A., 1996. A Maximum Entropy Model for Part-Of-Speech Tagging. In *Proceedings of the conference on empirical methods in natural language processing*. 133–142. Available

- at: <http://www.anthology.aclweb.org/W/W96/W96-0213.pdf> [Accessed July 28, 2017].
- Ratnaparkhi, A., 1997. A Simple Introduction to Maximum Entropy Models for Natural Language Processing. *IRCS Technical Reports Series: I RCS-97-08.*, (May), p.81. Available at: http://www-rohan.sdsu.edu/~gawron/mt_plus/mt/course_core/lectures/max_ent_gentle.ps.
- Ratnaparkhi, A., 1998. *Maximum entropy models for natural language ambiguity resolution*. University of Pennsylvania.
- Renuka and Terdal, S., 2017. Markov Random Field Region Based Text Detection and Segmentation by Stroke Width Transform. –*International Journal for Innovative Research in Science & Technology*, 4(2), p.195.
- van Rijsbergen, C.J., 1979. *Information Retrieval* Second Edi., London: Butterworths.
- Rissanen, J., 1978. Modelling by the shortest data description. *Automatica*, 14, pp.465–471.
- de Roeck, A.N. and Al-Fares, W., 2000. A Morphologically Sensitive Clustering Algorithm for Identifying Arabic Roots. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. 199–206. Available at: <http://www.aclweb.org/anthology/P00-1026> [Accessed September 6, 2017].
- Rosas, F., Mediano, P.A.M., Ugarte, M., and Jensen, H.J., 2018. An Information-Theoretic Approach to Self-Organisation: Emergence of Complex Interdependencies in Coupled Dynamical Systems. *Entropy*, 20(10), p.793.
- Rosenberg, A. and Hirschberg, J., 2007. V-measure: A conditional entropy-based external cluster evaluation measure. *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language (EMNLP-CoNLL'07)*, 1(June), pp.410–420. Available at: <http://acl.ldc.upenn.edu/D/D07/D07-1043.pdf>.
- van Rossum, G., 2007. Python Programming Language. *USENIX Annual Technical Conference*, 41.
- Roth, R., Rambow, O., Habash, N., Diab, M., and Rudin, C., 2008. Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In *ACL-08: HLT - 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*. Columbus, Ohio, USA, 117–120.
- Rumelhart, D.E., Hinton, G.E., and Williams, R.J., 1986. Learning representations by back-propagating errors. *Nature*, 323(6088), pp.533–536.

- Ruokolainen, T., Kohonen, O., Virpioja, S., and Kurimo, M., 2013. Supervised Morphological Segmentation in a Low-Resource Learning Setting using Conditional Random Fields. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. Sofia, Bulgaria: Association for Computational Linguistics, 29–37.
- Russell, S. and Norvig, P., 2014. *Artificial Intelligence: A Modern Approach* 3rd ed., London: Pearson.
- Sami Virpioja, Ville T. Turunen, Sebastian Spiegler, Oskar Kohonen, and Mikko Kurimo, 2011. Empirical Comparison of Evaluation Methods for Unsupervised Learning of Morphology. *TAL*, 52(2), pp.45–90. Available at: <https://www.researchgate.net/publication/229076997> [Accessed July 17, 2021].
- Sarikaya, R., Afify, M., and Gao, Y., 2007. Joint Morphological-Lexical Language Modeling (JMLLM) for Arabic. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. IEEE, IV–181. Available at: https://419bb89b-a-62cb3a1a-s-sites.googlegroups.com/site/ruhisarikaya01/publications/icassp2007_2.pdf?attachauth=ANoY7co-U14XXzyFpflWp6PnNaj7bBldwC9pOj2uopr-0lBq4W2pOvL_WkyvrTmhVaWgwX8f8Uep7LoEkvLYWPBh5eXm9J3R2aRc669UcHZso5SaFPOfM_bxyZE_rOuVcBP-ydigX6Stu [Accessed March 17, 2018].
- Sarikaya, R. and Deng, Y., 2007. Joint Morphological-Lexical Language Modeling for Machine Translation. In *Proceedings of NAACL HLT 2007*. Rochester, NY: Association for Computational Linguistics, 145–148. Available at: <http://www.aclweb.org/anthology/N07-2037> [Accessed March 17, 2018].
- Schmid, H., 2005. A programming language for finite state transducers. In *FSMNLP*. Helsinki, Finland, 308–309.
- Schone, P. and Jurafsky, D., 2000. Knowledge-Free Induction of Morphology Using Latent Semantic Analysis. In *Proceedings of CoNLL-2000 and LLL-2000*. Lisbon, Portugal, 67–72.
- Schreiber, T., 2000. Measuring information transfer. *Physical Review Letters*, 85(2), pp.461–464.
- Scutari, M., 2018. Dirichlet Bayesian network scores and the maximum relative entropy principle. *Behaviormetrika*, 45(2), pp.337–362. Available at: <https://doi.org/10.1007/s41237-018-0048-x>.
- Shalonova, K. and Golenia, B., 2010. Weakly Supervised Morphology Learning for Agglutinating Languages Using Small Training Sets. In *Proceedings of the 23rd International Conference*

- on *Computational Linguistics (Coling 2010)*. Beijing, China, 976–983.
- Shalounova, K., Golénia, B., and Flach, P., 2009. Towards Learning Morphology for Under-Resourced Fusional and Agglutinating Languages. *IEEE Transactions on Audio, Speech and Language Processing*, 17(5), pp.956–965.
- Shannon, C.E., 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3), pp.379–423. Available at: <http://dl.acm.org/citation.cfm?id=584093> [Accessed July 19, 2014].
- Silfverberg, M. and Hulden, M., 2017. Automatic Morpheme Segmentation and Labeling in Universal Dependencies resources. In *Proceedings of the NoDaLiDa 2017 Workshop on Universal Dependencies (UDW 2017)*. Gothenburg, Sweden, 140–145. Available at: <http://www.aclweb.org/anthology/W17-0418> [Accessed August 28, 2018].
- da Silva, J.F., Dias, G., Guillore, S., and Lopes, J.G.P., 1999. Using LocalMaxs Algorithm for the Extraction of Contiguous and Non-contiguous Multiword Lexical Units. In *Portuguese Conference on Artificial Intelligence*. Springer, Berlin, Heidelberg, 113–132. Available at: <https://www.researchgate.net/publication/220773699> [Accessed August 31, 2018].
- Sirts, K. and Goldwater, S., 2013. Minimally-Supervised Morphological Segmentation using Adaptor Grammars. *Transactions of the Association for Computational Linguistics*, 1, pp.255–266. Available at: <http://www.aclweb.org/anthology/Q/Q13/Q13-1021.pdf>.
- Smith, N.A. and Eisner, J., 2005. Contrastive Estimation: Training Log-Linear Models on Unlabeled Data. In *Proceedings of the 43rd Annual Meeting of the ACL*. Ann Arbor, Michigan, USA, 354–362. Available at: <http://www.anthology.aclweb.org/P/P05/P05-1044.pdf> [Accessed October 29, 2017].
- Smith, N.A., Smith, D.A., and Tromble, R.W., 2005. Context-Based Morphological Disambiguation with Random Fields. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*. Vancouver, Canada, 475–482. Available at: <http://acl-arc.comp.nus.edu.sg/archives/acl-arc-090501d3/data/pdf/anthology-PDF/H/H05/H05-1060.pdf> [Accessed August 7, 2017].
- Snyman, D., Van Huyssteen, G.B., and Daelemans, W., 2012. Cross-Lingual Genre Classification for Closely Related Languages. In *Twenty-Third Annual Symposium of the Pattern Recognition Association of South Africa*. 132–137. Available at: <http://www.let.rug.nl/~vannoord/Lassy/deliverable1-1.pdf> [Accessed October 25, 2018].

- South African Parliament, 1996. *UMgaqo-siseko weRiphabliki yoMzantsi-Afrika ka-1996*, Available at: <http://www.justice.gov.za/legislation/constitution/SACConstitution-web-xho.pdf> [Accessed October 25, 2018].
- Spiegler, S., Gołenias, B., Shalanova, K., Flach, P., and Tucker, R., 2008. Learning the morphology of zulu with different degrees of supervision. In *2008 IEEE Workshop on Spoken Language Technology, SLT 2008 - Proceedings*. 9–12.
- Spiegler, S. and Monson, C., 2010. EMMA: A Novel Evaluation Metric for Morphological Analysis. *Proceedings of the 23rd International Conference on Computational Linguistics. Association for Computational Linguistics*, pp.1029–1037. Available at: <http://www.cs.bris.ac.uk/> [Accessed August 4, 2017].
- Spiegler, S., van der Spuy, A., and Flach, P.A., 2010a. *Additional material for the Ukwabelana Zulu corpus. CSTR-10-003*,
- Spiegler, S., van der Spuy, A., and Flach, P.A., 2010b. Ukwabelana - An open-source morphological Zulu corpus. *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, (August), pp.1020–1028.
- Stanfill, C. and Waltz, D., 1986. Towards memory-based reasoning. *Communicaiton of the ACM*, 29(12), pp.1213–1228.
- Statistics South Africa, 2016. *Community survey 2016 in Brief*, Available at: <https://www.statssa.gov.za/publications/03-01-06/03-01-062016.pdf>.
- Stolcke, A., 1995. An Efficient Probabilistic Context-Free Parsing Algorithm that Computes Prefix Probabilities. *Computational Linguistics*, 21(2), pp.165–201.
- Stolcke, A., 2002. Srilm — an Extensible Language Modeling Toolkit. *Interspeech*, 2(Denver, Colorado), pp.901–904.
- Stroppa, N. and Yvon, F., 2005. An Analogical Learner for Morphological Analysis. In *Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL)*. Ann Arbor: Association for Computational Linguistics, 120–127.
- Sun, M., Shen, D., and Tsou, B.K., 1998. Chinese Word Segmentation without Using Lexicon and Hand-crafted Training Data. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2*. Association for Computational Linguistics, 1265–1271. Available at:

- <https://aclanthology.info/pdf/C/C98/C98-2201.pdf> [Accessed February 19, 2018].
- Teahan, W.J., Wen, Y., Mcnab, R., and Witten, I.H., 2000. A Compression-based Algorithm for Chinese Word Segmentation. *Computational Linguistics*, 26(3), pp.375–393. Available at: <https://www.mitpressjournals.org/doi/pdfplus/10.1162/089120100561746> [Accessed February 27, 2018].
- Teh, Y.W., 2006a. *A Bayesian Interpretation of Interpolated Kneser-Ney*, Singapore.
- Teh, Y.W., 2006b. A Hierarchical Bayesian Language Model based on Pitman-Yor Processes. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*. Sydney, Australia: Association for Computational Linguistics, 985–992. Available at: <http://discovery.ucl.ac.uk/150521/> [Accessed March 17, 2017].
- Teh, Y.W., Jordan, M.I., Beal, M.J., and Blei, D.M., 2005. Hierarchical Dirichlet Processes *. *Journal of the American Statistical Association.*, 101, pp.1566–1581. Available at: <http://www.gatsby.ucl.ac.uk/~ywteh/research/npbayes/jasa2006.pdf> [Accessed November 10, 2017].
- Theron, P. and Cloete, I., 1997. Automatic acquisition of two-level morphological rules. In *Proceedings of the fifth conference on Applied Natural Language Processing*. Washington, DC: Morgan Kaufmann Publishers, San Francisco, CA, 103–110.
- Tian, S., Lu, S., Su, B., and Lim Tan, C., 2015. Robust Text Segmentation using Graph Cut. In *13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 331–335. Available at: <http://vision.middlebury.edu/MRF/> [Accessed March 10, 2020].
- Tshabe, S.L. and Shoba, F., Eds., 2006. *The Greater Dictionary of isiXhosa: Volume 1 (A to J)*, Alice: University of Fort Hare.
- Uchiumi, K., Tsukahara, H., and Mochihashi, D., 2015. Inducing Word and Part-of-Speech with Pitman-Yor Hidden Semi-Markov Models. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. Beijing, China: Association for Computational Linguistics, 1774–1782. Available at: <http://www.aclweb.org/anthology/P15-1171> [Accessed January 20, 2018].
- Vapnik, V.N., 1999. An Overview of Statistical Learning Theory. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 10(5). Available at: <http://citeseer.ist.psu.edu/viewdoc/download;jsessionid=223B2B87090CDFEB19C83B67C>

26248F7?doi=10.1.1.332.356&rep=rep1&type=pdf [Accessed May 30, 2018].

- Virpioja, S., Väyrynen, J.J., Creutz, M., and Sadeniemi, M., 2007. Morphology-Aware Statistical Machine Translation Based on Morphs Induced in an Unsupervised Manner. In *Proceedings of the Machine Translation Summit XI*. 491–498.
- Viterbi, A.J., 1967. Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Transactions on Information Theory*, 13(2), pp.260–269. Available at: http://www.nt.tuwien.ac.at/fileadmin/courses/389075/Error_Bounds_for_Convolutional_Codes_and_an_Asymptotically_Optimum_Decoding_Algorithm.pdf.
- Wang, C., Wang, Y., Huang, P.-S., Mohamed, A., Zhou, D., and Deng, L., 2017. Sequence Modeling via Segmentations. In *Proceedings of the 34 th International Conference on Machine Learning*. Sydney, Australia. Available at: <https://arxiv.org/pdf/1702.07463.pdf> [Accessed April 9, 2018].
- Wang, H., Zhu, J., Tang, S., and Fan, X., 2011. A New Unsupervised Approach to Word Segmentation. *Computational Linguistics*, 37(3), pp.421–454. Available at: https://www.mitpressjournals.org/doi/pdf/10.1162/COLI_a_00058 [Accessed August 26, 2018].
- Wang, L., Cao, Z., Xia, Y., and de Melo, G., 2016. Morphological Segmentation with Window LSTM Neural Networks. *Proceedings of the 30th {AAAI} Conference on Artificial Intelligence (AAAI 2016)*, pp.2842–2848.
- Wang, M. and Manning, C.D., 2013. Effect of Non-linear Deep Architecture in Sequence Labeling. In *International Joint Conference on Natural Language Processing*. Nagoya, Japan, 1285–1291.
- Wang, X., Cai, D., Li, L., Xu, G., Zhao, H., and Si, L., 2019. Unsupervised Learning Helps Supervised Neural Word Segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence, 7200–7207.
- Welman, C., Kruger, F., and Mitchell, B., 2005. *Research Methodology* 3rd Editio., Cape Town: Oxford University Press, Southern Africa.
- Wilcoxon, F., 1945. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6), pp.80–83.

- Williams, P.L., 2011. *Information dynamics: Its theory and application to embodied cognitive systems*. Indiana University. Available at: <http://search.proquest.com.proxyiub.uits.iu.edu/pqdt/docview/1494133299/abstract/C112104AA1964384PQ/2?accountid=11620>.
- Williams, P.L. and Beer, R.D., 2010. Nonnegative Decomposition of Multivariate Information. *arXiv:1004.2515*.
- Wu, Y. and Zhao, H., 2018. Finding Better Subword Segmentation for Neural Machine Translation. *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pp.53–64. Available at: <https://github.com/Lindsay125/gbpe>. [Accessed April 13, 2020].
- Yamron, J.P., Carp, I., Gillick, L., Lowe, S., Mulbregt, P. Van, and Street, N., 1998. A hidden Markov model approach to text segmentation and event tracking. *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, 1(January 1992), pp.333–336.
- Yang, J., Zhang, Y., and Dong, F., 2017. Neural Word Segmentation with Rich Pretraining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Vancouver, Canada: Association for Computational Linguistics, 839–849. Available at: <https://doi.org/10.18653/v1/P17-1078> [Accessed April 7, 2020].
- Ye, Y., Wu, Q., Li, Y., Chow, K.P., Hui, L.C.K., and Yiu, S.M., 2013. Unknown Chinese word extraction based on variety of overlapping strings. *Information Processing and Management*, 49(2), pp.497–512. Available at: <http://dx.doi.org/10.1016/j.ipm.2012.09.004> [Accessed August 27, 2018].
- Yu, S.Z., 2010. Hidden semi-Markov models. *Artificial Intelligence*, 174, pp.215–243. Available at: www.elsevier.com/locate/artint [Accessed April 16, 2018].
- Zeldes, A., 2019. A Characterwise Windowed Approach to Hebrew Morphological Segmentation. In *arXiv preprint arXiv:1808.07214*. 101–110. Available at: <https://arxiv.org/pdf/1808.07214.pdf> [Accessed August 28, 2018].
- Zhang, H.-P., Qun Liu, Xue-Qi Cheng, Hao Zhang, and Hong-Kui Yu, 2003. Chinese Lexical Analysis Using Hierarchical Hidden Markov Model. In *Proceedings of the second SIGHAN workshop on Chinese language processing-Volume 17*. Association for Computational Linguistics., 63–70. Available at: <https://aclanthology.info/pdf/W/W03/W03-1709.pdf> [Accessed February 27, 2018].

- Zhang, M., Yu, N., and Fu, G., 2018. A Simple and Effective Neural Model for Joint Word Segmentation and POS Tagging. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 26(9), pp.1528–1538.
- Zhang, M., Zhang, Y., and Fu, G., 2018. Transition-based neural word segmentation using word-level features. *Journal of Artificial Intelligence Research*, 63, pp.923–953.
- Zhang, X., Wei, C., and Zhao, H., 2017. Tracing a Loose Wordhood for Chinese Input Method Engine. *arXivar prePrint Xiv:1712.04158v1* .
- Zhang, Y., Lei, T., Barzilay, R., and Jaakkola, T., 2014. Greed Is Good If Randomized : New Inference for Dependency Parsing. *Conference on Empirical Methods on Natural Language Processing*, (October).
- Zhang, Y., Li, C., Barzilay, R., and Darwish, K., 2015. Randomized Greedy Inference for Joint Segmentation, POS Tagging and Dependency Parsing. In *Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*. Denver, Colorado, 42–52. Available at: http://www.aclweb.org/website/old_anthology/N/N15/N15-1005.pdf [Accessed April 7, 2017].
- Zhao, H., Huang, C.-N., and Li, M., 2006. An Improved Chinese Word Segmentation System with Conditional Random Fields. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*. Sydney, Australia: Association for Computational Linguistics, 162–165. Available at: <http://www.aclweb.org/anthology/W06-0127> [Accessed February 26, 2018].
- Zheng, X., Chen, H., and Xu, T., 2013. Deep Learning for Chinese Word Segmentation and POS Tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, 647–657.
- Zhikov, V., Takamura, H., and Okumura, M., 2010. An Efficient Algorithm for Unsupervised Word Segmentation with Branching Entropy and MDL. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Massachusetts, USA: Association for Computational Linguistics, 832–842. Available at: <http://www.aclweb.org/anthology/D10-1081> [Accessed August 26, 2018].
- Zhikov, V., Takamura, H., and Okumura, M., 2013. An Efficient Algorithm for Unsupervised Word Segmentation with Branching Entropy and MDL. *Information and Media Technologies*, (reprinted from: *Transactions of the Japanese Society for Artificial Intelligence*, 2013, 28(3), pp. 347-360), 8(2), pp.514–527. Available at: <http://www.lr.pi.titech.ac>. [Accessed February

5, 2018].

Zhou, H., Yu, Z., Zhang, Y., Huang, S., Dai, X., and Chen, J., 2017. Word-Context Character Embeddings for Chinese Word Segmentation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, 760–766. Available at: <http://zhangmeishan.github.io/> [Accessed April 7, 2020].

Zoph, B., Vaswani, A., May, J., and Knight, K., 2016. Simple, Fast Noise-Contrastive Estimation for Large RNN Vocabularies. In *Proceedings of NAACL-HLT 2016*. San Diego, California: Association for Computational Linguistics, 1217–1222. Available at: <http://www.aclweb.org/anthology/N16-1145> [Accessed March 14, 2018].

Zwicky, F., 1966. Entdecken, erfinden, forschen: im morphologischen Weltbild. *Muenchen: Droemer*.

ANNEXURES A: GLOSSES USED

Gloss	Description
AC	Adjective concord
AdjRoot* ³	Adjective root
AdjStem*	Adjective stem
AdjTerm*	Adjective terminal vowel
AdvPre	Adverb prefix
AdvRoot*	Adverb root
AdvStem*	Adverb stem
AdvTerm*	Adverb terminal vowel
ApplExt	Applied extension
AugSuf	Augmentative suffix
AuxVStem	Auxiliary verb stem
BPre	Basic prefix
CausExt	Causative extension
CopPre	Copulative prefix
Dem*	Demonstrative prefix
DesPre*	Descriptive prefix
DiffPref*	Differentiative prefix
DiffRoot*	Differentiative root
DiffStem*	Differentiative stem
DiffTerm*	Differentiative terminal vowel
DimSuf*	Diminutive suffix
Drift*	The drift consonant in the production of a demonstrat
EnumStem*	Enumerative stem

³ Glosses with ‘*’ are identified specifically for this study. They may not be grammatical

EnumRoot*	Enumerative root
EnumTerm*	Enumerative termin vowel
ExclNeg	Exclusive negative
IntensExt	Intensive extension
LocSuf	Locative suffix
LongPres	Long present tense
NegCirc*	Negative circumfix suffix
NegPre	Negative prefix
NeutExt	Neuter extension
NPrePre	Noun preprefix
NRoot*	Noun Root
NStem	Noun stem
NTerm*	Noun Terminal vowel
PassExt	Passive extension
PC	Possessive concord
PosPre	Possessive prefix
PosRoot*	Possessive root
PosStem*	Possessive stem
Posterm*	Possessive terminal vowel
Pot	Potential
ProgPre	Progressive prefix
PronPref	Pronoun prefix
PronRoot*	Pronoun root
PronStem	Pronoun stem
PronTerm*	Pronoun terminal vowel
QC	Quantitative concord
QuantPref*	Quantitative prefix

RecipExt	Reciprocal extension
RelPref*	Relative prefix
RelStem	Relative stem
RelTerm*	Relative terminal vowel
SC	Subject concord
SupPref*	Superlative prefix
VPerfExt	Verb extension perfect
VRoot	Verb root
VStem	Verb stem
VT	Verb terminative
VTPerf	Verb terminative perfect