

UNIVERSITY OF THE WESTERN CAPE

**Short Message Service Normalization for  
Communication with a Health  
Information System**

by

Ademola Olusola Adesina

a thesis submitted in fulfilment of the  
requirements for the degree of  
Doctor of Philosophy  
in  
Computer Science

Faculty of Science  
Department of Computer Science

November 2013

# Declaration of Authorship

I, Ademola Olusola Adesina, declare that this thesis *Short Message Service Normalization for Communication with a Health Information System* is my own work, that it has not been submitted for any degree or assessment at any other University, and that all the sources I have used or quoted have been indicated and acknowledged by means of complete reference

Signed:

---

ADEMOLA OLUSOLA ADESINA

Date:

---

UNIVERSITY OF THE WESTERN CAPE

# *Abstract*

Faculty of Science  
Department of Computer Science

Doctor of Philosophy

by

Ademola Olusola Adesina

Supervisor: Prof. Henry O. Nyongesa

Short Message Service (SMS) is one of the most popularly used services for communication between mobile phone users. In recent times it has also been proposed as a means for information access. However, there are several challenges to be overcome in order to process an SMS, especially when it is used as a query in an information retrieval system. SMS users often tend deliberately to use compacted and grammatically incorrect writing that makes the message difficult to process with conventional information retrieval systems. To overcome this, a pre-processing step known as normalization is required. In this thesis an investigation of SMS normalization algorithms is carried out. To this end, studies have been conducted into the design of algorithms for translating and normalizing SMS text. Character-based, unsupervised and rule-based techniques are presented. An investigation was also undertaken into the design and development of a system for information access via SMS. A specific system was designed to access information related to a Frequently Asked Questions (FAQ) database in healthcare, using a case study. This study secures SMS communication, especially for healthcare information systems. The proposed technique is to encipher the messages using the secure shell (SSH) protocol.

**Keywords:** Short message service (SMS), SMS normalization, SMS translation, Spelling error, Spelling correction, SMS security, Frequently asked questions (FAQ), HIV/AIDS, Mobile health (mHealth), Information retrieval (IR)

# *Acknowledgment*

First and foremost, my reserved thanks go to THE ALMIGHTY GOD, the beginning and the end. He has given me life, hope and courage to complete the programme. My utmost thanks go to my supervisor Professor Henry O. Nyongesa. Without his support and guidance, I would have been lost in the wilderness of research. Yes, my academic background had prepared me, in a little way, for the challenges and realities of the PhD. But with his encouragements I have become victorious. I feel honoured and privileged to have worked with him.

I sincerely appreciate the foundation you laid in our life as a father, Pa Joseph Okunade Adesina. Even though you are no more, I will always remember your lessons of aiming towards excellence in life through hard-work and prayer. Father, the training and motivation I have inherited from you improve my attitude of perseverance, humility, determination, courage, commitment, and contentment. These attributes will take me far!

I appreciate every member of staff of the Department of Computer Science, headed by Professor Isabella Venter. Professors Tiko Iyamu and Bill Tucker, your words of encouragement rekindle the spirit of hope in me, that I will finish this programme. My special thanks go to Mr. Reg Dodds. I was surprised on the occasion when you called me your friend. I appreciate every effort you put into making the work suitable for assessment. Ms Rene Abbott, Fatima Jacobs and Messrs Daniel Leenderts, Andries Kruger, Yasser Buchana and Thomson Khosa, I appreciate you all. I cannot but mention the support I received through the University work study, PG Research Committee and PET programmes; all these mean so much to me.

I am also indebted to the UWC library staff (especially, the Faculty of Science librarians) who always assisted me in finding related literature to my area of study.

I appreciate my colleagues in the lab and my research group, both past and present— Messrs KK Agbele, AP Abidoye and Dr NA Azeez. For the constructive criticism of my work, suggestions, and general to and fro of intellectual discussion. I must with great appreciation mention my statistician, Siaka Lounge, who tirelessly assisted me with the statistical aspects of my work. I would also like to thank Caroline Tagg and Liu for making their SMS corpora available.

I want to thank everyone God has used—spiritually, physically, financially— people with whom I shared my emotions in the darkest moments of this epic adventure; when I felt tempted to give up, they gave me strength to continue. I feel honoured and privileged to say thank you to my wife for her enduring support, endless love, prayers and care

---

over for the past five years; she is a wonderful partner. I appreciate the special gifts of God for my family—Israel, Darasimi, and Honour—God bless you all. To my big daughter—Comfort John—I say thank you.

Thank you to all my fathers in the Lord that have stood by me in their prayers—Prophets Arisekola, Bello, Ezekiel, Nathaniel, Ogunrinde, and Tovide; the members of congregations of The Redeemed Christian Church of God (RCCG); the Household of God Parish and Harvest Centre; the RCCG Pastors—Alegbe, Balogun, Fatoba, Oke, Olayinka, Olowu, Omotosho, Osas, and Wole & Ronke Adesina; Deaconesses Fatoba and Akinyeye, Pastor & Mrs Olugbade, Pastor & Mrs Akinleye, Brother and Sister Oyenuga. I thank you all. You will never run dry of the anointing of God.

I especially would like to thank my sponsor, Lagos State University (LASU), for the opportunity to further my studies. To my special friend and colleague, Mrs. GM Saibu, and of the LASU colleagues pursuing their PhDs at UWC—MJ Alegbe, OO Tovide, Cole-Showers, AP Abidoye, OR Rufai, Paul Sewanu, Azeez Olaiya, HA Bamikole—we shall all eat the labour of our hands. My special thanks also go to Mr & Mrs Rotimi Adeniyi, Ms Omoyeni AO, Deacon & Mrs Adewumi, Brothers Emmanuel Amel, Gbenga Ogidan, Kenneth Kehinde, and Dr. Wale Ajuwon. Thank you.

I appreciate you my mother, Mama Maria Adesina, for your love and prayer, and also to my step mother, Mrs Dupe Adesina, I say thank you. To all my siblings—Mrs CM Afolabi, Messrs Tolulope, George and Wole Adesina, and Mrs Femi Ajekigbe—also to my half-sisters—Mrs Toyin Ejitade, Miss Yemisi Adesina, and Miss Seyi Adesina, I show my gratitude.

My special thanks go to my in-laws, Pa SO Fawole, Mrs Owolabi, Opeyemi Fawole and Tunde Fawole, Mrs Popoola and Mrs Ijaduola.

I also remember my good neighbours—Alhaja Olanrewaju (landlady), Mr & Mrs Kehinde Olawunmi, Mr & Mrs Lekan Rahman, Mr & Mrs Ajayi, Mrs Taiwo, and Mr & Mrs Kehinde. I appreciate every role you played in my family while i was away. God bless you all.

To all my childhood friends, I am grateful for your friendship—Deacon Yinka Fasasi, Niyi Adeyeye, Kazeem Bello, Femi Oyeniran, Bola Ajayi, Curtis Olayinka, Dr. MA Alabi, Wale Egunjobi, Azeez Taiwo, Mrs. EA Adeyemi, Mrs. Bose Oguntade, Mrs. Imoh Eyoh, Dr. Akintayo Adebayo, SG Adedapo, TJ Odule, JS Oladimeji, George Airhekhola, Wole Ilori, Ibrahim Ayankola and Rev. Olusola Ladipo.

Messrs MJ Alegebe, Yomi Onanuga, JO Ogunjinmi and Tola Odumade you are all special to me.

My academic mentors—Professors AC Odebode, NA Olasupo, MAO Bankole, and AO Oluwade; I appreciate you all.

## *List of Publications*

**Adesina, A. O.**, Agbele, K. K., Abidoye, A. P., and Nyongesa, H. O. SMS-based healthcare FAQ information retrieval system. *Information Retrieval (Springer)* (under review)

**Adesina, A. O.**, Agbele, K. K., Abidoye, A. P., and Nyongesa, H. O. Text messaging and retrieval techniques for mobile health information systems. *Journal of Information Science (SAGE Publication)* (under review)

**Adesina, A. O.**, and Nyongesa, H. O., A Mobile-Health Information Access System in *Proceedings of the Southern Africa Telecommunication Networks and Applications Conference (SATNAC)*, Spier Wine Estate, Stellenbosch 1–4 September 2013, Cape Town, South Africa pp. 191–197

**Adesina, A. O.**, and Nyongesa, H. O., Access to mobile health information using a short message service (SMS) in *Proceedings of the World Wide Applications, (ZAWWW 2013)*, 10–13 September 2013, Cape Town, South Africa

**Adesina, A. O.**, Agbele, K. K., Abidoye, A. P., and Azeez, N.A., Evaluating SMS Parsing Using Automated Testing Software. *Afr J. of Comp & ICTs*. Vol 5, No. 4. pp 53-62, 2012

**Adesina, A. O.**, Agbele, K. K., Februarie, R., Abidoye, A. P., and Nyongesa, H. O. Ensuring the security and privacy of information in mobile health-care communication systems. *South African Journal of Science*, vol. 107, 2011

**Adesina, A. O.** and Nyongesa, H. O., SMS Parsing for Information Accessing: A Medical Domain Application in *Proceedings of the Annual Postgraduate Research Open Day of the Faculty of Natural Sciences*, New Life Science Building, Oct 24 & 25, 2011, University of the Western Cape, Cape Town, South Africa

**Adesina, A. O.**, Agbele, K. K., Azeez, N. A., and Abidoye, A. P., A Query-Based SMS Translation in Information Access System. *International Journal of Soft Computing*, vol. 1(5), pp. 13–18, 2011

Abidoye, A. P., Azeez, N. A., **Adesina, A. O.**, Agbele, K. K., and Nyongesa, H. O. Using wearable sensors for remote healthcare monitoring system, *Journal of Sensor Technology*, 1(2), pp. 22–28, 2011

**Adesina, A. O.**, Nyongesa, H. O., and Agbele, K. K., Digital watermarking: A state-of-the-art review in *Proceedings of the IST-Africa*, 19–21 May 2010 Durban, South Africa, pp. 1–8

**Adesina, A. O.**, Agbele, K. K., and Nyongesa, H. O., Text Messaging: a tool in e-Health Services, in *Proceedings of the Southern Africa Telecommunication Networks and Applications Conference (SATNAC)*, Spier Wine Estate, Stellenbosch 5–8 September 2010, Cape Town, South Africa

Agbele, K., Nyongesa H., and **Adesina, A.**, ICT and information security perspectives in e-health systems, *J Mobile Commun*, vol. 1, pp. 17–22, 2010

Agbele, K. K., **Adesina, A. O.**, Nyongesa, H. O., and Febba, R., A Novel Approach Integrating Ranking Functions Discovery, Optimization and Inference to Improve Retrieval Performance, *International Journal of Soft Computing*, vol.5, pp.155–163, 2010



# Contents

<b>Declaration of Authorship</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Acknowledgment</b>	<b>iii</b>
<b>List of Publications</b>	<b>vi</b>
<b>List of Figures</b>	<b>xii</b>
<b>List of Tables</b>	<b>xiv</b>
<b>Abbreviations</b>	<b>xvi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Motivation and background . . . . .	1
1.2.1 The normalization of SMS text preserves the original language . . . . .	2
1.2.2 Normalized SMS text is important for natural language processing . . . . .	3
1.2.3 The cost-effectiveness of SMS has promoted other technologies to be developed on this platform . . . . .	3
1.2.4 The high penetration of SMS, especially among the youth, promotes information dissemination . . . . .	3
1.3 Problem definition . . . . .	4
1.4 Research question . . . . .	4
1.5 Research aims and objectives . . . . .	4
1.6 Research methodology . . . . .	4
1.7 Scope of study . . . . .	5
1.8 Contribution to knowledge . . . . .	5
1.9 Thesis outline . . . . .	6
<b>2 Review of Previous Research</b>	<b>8</b>
2.1 Introduction . . . . .	8
2.2 SMS classification . . . . .	8

2.3	SMS normalization techniques (supervised, semi-supervised and unsupervised learning approaches) . . . . .	10
2.4	SMS normalization: a review . . . . .	12
2.4.1	Noisy channel model . . . . .	13
2.4.2	Phrase-based normalization . . . . .	14
2.4.3	Character-level MT . . . . .	15
2.4.4	Character-string-based . . . . .	16
2.4.5	Letter transformation model . . . . .	16
2.4.6	Three-module architecture . . . . .	18
2.4.7	Statistical and rule-based modelling . . . . .	19
2.4.8	Spelling error and its normalization/correction . . . . .	19
2.5	Text entry errors: a review . . . . .	21
2.6	Similarity measurements: a review . . . . .	23
2.6.1	Dice's coefficient . . . . .	24
2.6.2	Longest Common Subsequence Ratio (LCSR) . . . . .	25
2.6.3	Word Error Rate (WER) or edit distance . . . . .	26
2.6.4	BLEU and human judgement . . . . .	27
2.7	The Least Character Distance (LCD) calculation . . . . .	29
2.8	Vowel selection through rule-based approach . . . . .	31
2.8.1	Order of vowel precedence . . . . .	34
2.9	SMS lexical normalization scope . . . . .	35
2.10	SMS normalization and mobile information access . . . . .	37
2.11	SMS-based FAQ information retrieval mechanism: a review . . . . .	39
2.12	Keyword extraction: a review . . . . .	44
2.13	SMS security . . . . .	46
2.13.1	Secure Shell (SSH) Protocol . . . . .	48
2.14	Chapter summary . . . . .	51
<b>3</b>	<b>Research Design and Methodology</b> . . . . .	<b>52</b>
3.1	Introduction . . . . .	52
3.2	Research design and approach . . . . .	53
3.3	Test data collection . . . . .	59
3.4	Description of the data structure and methodology used in SMS normalization . . . . .	61
3.5	The developed SMS normalization technique—SCORE algorithm . . . . .	63
3.5.1	Further description of the SCORE algorithm . . . . .	66
3.6	Experimentation methods for SMS normalization . . . . .	68
3.6.1	Experiment 1—Vowel stripping . . . . .	69
3.6.2	Experiment 2—Clipping positions . . . . .	70
3.6.3	Experiment 3—Frequency or probability model . . . . .	70
3.6.4	Experiment 4—Evaluation of two set corpora with SCORE algorithm . . . . .	71
3.6.5	Experiment 5—Annotator translations . . . . .	71
3.6.6	Experiment 6—Cross validation . . . . .	73
3.7	Description of data structure and methodology used in information access using SMS in a FAQ system . . . . .	75
3.7.1	Architecture, procedure and extraction process in <i>SMSql</i> . . . . .	76

3.7.2	Flow diagram of <i>SMSql</i> . . . . .	77
3.7.3	Applications of <i>n</i> -grams in SMS-based information retrieval system . . . . .	78
3.7.4	Scoring and ranking techniques . . . . .	79
3.7.4.1	<i>Tf-idf</i> measurements . . . . .	79
3.7.4.2	Vector space model . . . . .	80
3.7.4.3	Cosine similarity measurements . . . . .	81
3.8	SMS-based FAQ analytical methods . . . . .	81
3.8.1	Description of the FAQ database system . . . . .	81
3.8.2	Stop-word lists . . . . .	82
3.8.3	Identifying the query codes from query-answer pairs . . . . .	84
3.9	Experimental methodology on FAQ information access using SMS . . . . .	86
3.10	Algorithms for information retrieval experiments . . . . .	88
3.10.1	Application of scoring functions to the query selection using the three algorithms . . . . .	91
3.11	Statistical analysis . . . . .	92
3.12	Chapter summary . . . . .	93
<b>4</b>	<b>Results</b> . . . . .	<b>95</b>
4.1	Introduction . . . . .	95
4.2	Experimental results for SMS normalization . . . . .	95
4.2.1	Results obtained in Experiment 1—Vowel stripping . . . . .	96
4.2.2	Results obtained in Experiment 2—Clipping positions . . . . .	97
4.2.3	Results obtained in Experiment 3—Frequency or probability model . . . . .	97
4.2.4	Results obtained in Experiment 4—Evaluation of two set corpora using SCORE algorithm . . . . .	98
4.2.5	Results obtained in Experiment 5—Annotator translations . . . . .	99
4.2.5.1	Experiment 5a: Result obtained from English → SMS . . . . .	99
4.2.5.2	Experiment 5b: Result obtained from SMS → English . . . . .	100
4.2.5.3	Experiment 5c: Annotator with/without prior knowledge . . . . .	100
4.2.6	Results obtained in Experiment 6—Cross validations . . . . .	101
4.2.6.1	Experimental results . . . . .	103
4.2.6.2	Statistical analysis . . . . .	105
4.2.6.3	Significance test . . . . .	105
4.2.6.4	T-tests . . . . .	106
4.2.6.5	Correlations . . . . .	108
4.3	Experimental results on information access using SMS . . . . .	108
4.3.1	Results of <i>tf-idf</i> algorithm on information access using SMS . . . . .	109
4.3.2	Results of the <i>naive</i> algorithm on information access using SMS . . . . .	110
4.3.3	Results of <i>SMSql</i> algorithm on information access using SMS . . . . .	110
4.4	Performance evaluation . . . . .	111
4.5	Statistical analysis . . . . .	114
4.6	Chapter summary . . . . .	115
<b>5</b>	<b>Discussion and Conclusion</b> . . . . .	<b>117</b>
5.1	Overview . . . . .	117
5.2	Summary of research contribution . . . . .	119
5.3	Chapters recap . . . . .	119

---

5.4 Further work . . . . .	121
<b>Bibliography</b>	<b>123</b>
<b>Appendix A</b>	
Data set for FAQ information retrieval system	144
<b>Appendix B</b>	
PHP code for SCORE and SMSql algorithms	148
<b>Appendix C</b>	
Data structure of other modules	152
<b>Appendix D</b>	
Results of annotators	154
<b>Appendix E</b>	
HIV/AIDS websites	157

# List of Figures

2.1	Schematic representation of LCSR . . . . .	26
2.2	Automated FAQ information retrieval system . . . . .	39
2.3	Symmetric encryption diagram [101] . . . . .	47
2.4	Asymmetric encryption diagram [101] . . . . .	48
3.1	Four elements of the research process [64] . . . . .	53
3.2	Application database connection . . . . .	54
3.3	SMS normalization architecture showing various modules . . . . .	55
3.4	Information retrieval process . . . . .	56
3.5	Web-based SMS normalization and information retrieval flow diagram . . . . .	57
3.6	Frequently_used_smswords table . . . . .	62
3.7	SMS normalization flowchart . . . . .	63
3.8	System architecture of an SMS-query and reformulation process . . . . .	76
3.9	Flowchart of SMS question locator ( <i>SMSql</i> ) . . . . .	77
3.10	Punctuation/prepositions table . . . . .	83
4.1	Normalization performances on 100 data sets using different clipping position . . . . .	97
4.2	Relative frequency analysis of the 10 queries used for the experimentation . . . . .	98
4.3	Annotators' forward and backward selection . . . . .	100
4.4	Average precision of all the annotators . . . . .	105
4.5	Average precision of all the queries . . . . .	105
4.6	Boxplots . . . . .	107
4.7	Scattered plot of SCORE vs BLEU . . . . .	108
4.8	Average precision of the three algorithms . . . . .	112
4.9	Average recall of the three algorithms . . . . .	112
4.10	Comparison of the execution time of the three algorithms . . . . .	114
C.1	Acronyms/abbreviation . . . . .	152
C.2	Punctuation/preposition . . . . .	152
C.3	Homophone table . . . . .	153
C.4	English and medical words . . . . .	153
D.1	Average precision for query in Bin 1: BLEU and SCORE . . . . .	154
D.2	Average precision for query in Bin 2: BLEU and SCORE . . . . .	154
D.3	Average precision for query in Bin 3: BLEU and SCORE . . . . .	154
D.4	Average precision for query in Bin 4: BLEU and SCORE . . . . .	155
D.5	Average precision for query in Bin 5: BLEU and SCORE . . . . .	155
D.6	Average precision for query in Bin 6: BLEU and SCORE . . . . .	155

---

D.7 Average precision for query in Bin 7: BLEU and SCORE . . . . .	155
D.8 Average precision for query in Bin 8: BLEU and SCORE . . . . .	156
D.9 Average precision for query in Bin 9: BLEU and SCORE . . . . .	156
D.10 Average precision for query in Bin 10: BLEU and SCORE . . . . .	156

# List of Tables

2.1	Transformation in RID operations . . . . .	27
2.2	Top 10 most common substitution, deletion and insertion . . . . .	28
2.3	Examples of Least Character Distance and Percentage Error Rate . . . . .	30
2.4	Order of vowel precedence . . . . .	33
2.5	Comparison between symmetric encryption systems (stream algorithms) and asymmetric encryption systems (block algorithms) . . . . .	49
3.1	Liu and Caroline corpora . . . . .	60
3.2	SMS normalization database design . . . . .	61
3.3	Summary of the SMSs in each bin . . . . .	73
3.4	Relevance scores . . . . .	74
3.5	MySQL description of the FAQ database table . . . . .	82
3.6	Keywords extraction from FAQ data files . . . . .	84
3.7	SMS codes, query and keyword extraction . . . . .	85
3.8	Assigning token_id to the keyword . . . . .	85
3.9	(n x m) term-document matrix corresponding to the FAQ sentences . . . . .	86
3.10	Relevance judgment value . . . . .	88
3.11	Scoring function . . . . .	91
3.12	Questions and scores . . . . .	92
4.1	Results using vowel stripping method . . . . .	96
4.2	The results of normalized SMS from the Tagg (2009) corpus after appli- cation of the SCORE algorithm . . . . .	98
4.3	The results of normalized SMS from the Liu (2010) corpus after applica- tion of SCORE algorithm . . . . .	99
4.4	Annotator results obtained from English $\rightarrow$ SMS . . . . .	100
4.5	Annotator results obtained from SMS $\rightarrow$ English . . . . .	100
4.6	Results for the annotators . . . . .	101
4.7	Precision results for BLEU method . . . . .	102
4.8	Precision results for SCORE method . . . . .	103
4.9	Average precision of BLEU and SCORE algorithms for each query sen- tence conducted for the experiment . . . . .	104
4.10	Average precision of BLEU and SCORE algorithms for the annotators . . . . .	104
4.11	Summary of the SMS in each bin at the end of the normalization . . . . .	106
4.12	Test of normality . . . . .	107
4.13	Results of paired samples t-test . . . . .	108
4.14	Results of tf-idf algorithm . . . . .	109
4.15	Results of naive algorithm . . . . .	110

---

4.16	Results of SMSql algorithm . . . . .	111
4.17	Time computation for the retrieval process of the SMS queries . . . . .	113
4.18	Precision: descriptive analysis . . . . .	114
4.19	Multivariate test for precision . . . . .	115
4.20	Timing: descriptive analysis . . . . .	115
4.21	Multivariate test for timing . . . . .	115
5.1	Summary of research contribution . . . . .	120



# Abbreviations

<b>AIDS</b>	<b>A</b> cquired <b>I</b> mmune <b>D</b> eficiency <b>S</b> yndrome
<b>BLEU</b>	<b>B</b> iLingual <b>E</b> valuation <b>U</b> nderstudy
<b>FAQ</b>	<b>F</b> requently <b>A</b> sksed <b>Q</b> uestions
<b>HIV</b>	<b>H</b> uman <b>I</b> mmunodeficiency <b>V</b> irus
<b>IR</b>	<b>I</b> nformation <b>R</b> etrieval
<b>IV</b>	<b>I</b> n- <b>V</b> ocabulary
<b>LCD</b>	<b>L</b> east <b>C</b> haracter <b>D</b> istance
<b>LCSR</b>	<b>L</b> ongest <b>C</b> ommon <b>S</b> ubsequences <b>R</b> atio
<b>LM</b>	<b>L</b> anguage <b>M</b> odelling
<b>mHealth</b>	<b>m</b> obile <b>H</b> ealth
<b>MT</b>	<b>M</b> achine <b>T</b> ranslation
<b>NLP</b>	<b>N</b> atural <b>L</b> anguage <b>P</b> rocessing
<b>NSW</b>	<b>N</b> on- <b>S</b> tandard <b>W</b> ord
<b>OOV</b>	<b>O</b> ut- <b>o</b> f- <b>V</b> ocabulary
<b>POS</b>	<b>P</b> art- <b>o</b> f- <b>S</b> peech
<b>QAS</b>	<b>Q</b> uestion <b>A</b> nswering <b>S</b> ystem
<b>SASL</b>	<b>S</b> outh <b>A</b> frica <b>S</b> ign <b>L</b> angue
<b>SCORE</b>	<b>S</b> earch <b>C</b> ompare <b>R</b> eplace
<b>SER</b>	<b>S</b> entence <b>E</b> rror <b>R</b> ate
<b>SMS</b>	<b>S</b> hort <b>M</b> essage <b>S</b> ervice
<b>SSH</b>	<b>S</b> ecure <b>S</b> hell
<b>TTS</b>	<b>T</b> ext- <b>T</b> o- <b>S</b> peech
<b>WER</b>	<b>W</b> ord <b>E</b> rror <b>R</b> ate

*Dedicated*

*to my beloved wife (Ajibola Olusola Adesina),*

*the children (Mofiyinfooluwa, Oluwadarasimi & Mogboluwagbo),*

*&*

*Professor Isabella Venter*

# Chapter 1

## Introduction

### 1.1 Introduction

This research sets out to investigate ways of receiving an accurate response when a short message service (SMS) text sends a query to a frequently asked questions (FAQ) database server in order to garner advice on a specific health domain. The SMS text must be translated from its informal, ambiguous state in order for it to be used to access information from the FAQ database; hence there is a need for SMS normalization.

This chapter is divided into nine sections. Section 1.2 discusses the motivation and background of the research. In Section 1.3 the problem definition of the research is described. Section 1.4 discusses the research question and Section 1.5 considers the research aims and objectives. In Section 1.6 the research methodology is outlined. The scope of the research and contributions of the research to knowledge are covered in Sections 1.7 and 1.8 respectively. Lastly, the outline of the thesis is presented in Section 1.9.

### 1.2 Motivation and background

SMS is the most frequently used service on many mobile phones, from low-end mobile phones to smart handsets. It has given rise to a unique and continually evolving language, the syntax of which is based on convenience and the homophony of the words. SMS provides a platform where messages can be delivered even when the recipient is engaged in voice communication or is otherwise unable to attend to a call. This computer-mediated communication has its own peculiarities, where users have their own patterns of writing, inventing new abbreviations, and using non-standard orthographic forms [85].

This style of language is ubiquitous, quick and easy for purposes of communication and sharing of information. SMS language has inconsistent spelling since it is constantly re-invented by users. This makes it difficult to provide a comprehensive dictionary of all English SMS words. SMS communication has a general appeal, especially to the youth, because of its flexible use of alphanumeric characters, with little or no regard for orthographical and grammatical rules. This flexibility and freedom poses challenges for translating SMS into formal writing suitable for information processing. SMS language has, however, been recognized and accepted as a variant of natural language [171]. Thus, there is compelling motivation to make it possible to build information-based services using SMS communication, through the process of normalizing the various forms in which the language appears [102, 121].

By definition, SMS normalization refers to the task of converting SMS text that could be *noisy* into its intended *non-noisy* form [9, 24, 68]. Normalization involves tokenization of SMS input text, identification of non-standard words (NSW) and their categories, and the expansion of NSWs into standard word representations. The tokenization is usually based on white-space delimiters [215]. SMS normalization is similar to spellchecking [186] but differs in that the lexical variants in text messages are often intentionally generated [95]. The noisy text could have been created by a range of different techniques, such as the use of: (1) acronyms and abbreviations—*omg* for *oh my god*; (2) clipping—*ystday* for *yesterday*; (3) contraction—*thanku* for *thank you*; (4) phonetic substitution—*4ward* for *forward*; (5) homophone—*u* for *you* and (6) typing errors—*belive* for *believe*.

There is a need to normalize the noisy text for any further natural language processing work. For example, SMS can be used for information access and retrieval in an FAQ system where the response to an enquiry can be sourced in healthcare-related applications. The following are the reasons why SMS normalization may be considered important.

### 1.2.1 The normalization of SMS text preserves the original language

Text normalization arose as a result of a quest for an effective way to refine very noisy and ungrammatical text messages. SMS normalization aims to preserve the original languages [174], as it may be argued that uncontrolled use of SMS messages might lead to a deterioration of native languages. Such an effect is felt in the ever-decreasing effectiveness of translation technology. The normalization of informal text is complicated by the presence of numerous different abbreviations for the same term, making SMS messages difficult to use in natural language processing systems for information extraction, filtering, indexing and retrieval, and also for spam filtering and summarization techniques [100, 132]. Any processing application that uses SMS text will have to normalize it first.

### **1.2.2 Normalized SMS text is important for natural language processing**

SMS text can be used to retrieve information from a search engine only when it has been normalized. Text messaging can be turned into a major tool for accessing information databases. For example, educational information on HIV/AIDS can be passed to the youth via SMS because an appreciable percentage of them embrace the technology. The structure of SMS may, however, disallow its use as a query in search engine architecture. In order to achieve successful SMS queries in search engine architecture, parsed SMS (transformed into an original English spelling) can be used. In this research SMS serves as a tool for accessing information about HIV/AIDS on an FAQ system querying English and medical dictionaries.

### **1.2.3 The cost-effectiveness of SMS has promoted other technologies to be developed on this platform**

SMS is commonly used in the health system, as may be noticed in the areas of appointment reminders, medication taking, telemedicine, accessing patient records, communicating test results, measuring treatment compliance, raising health awareness, or monitoring patient illness. It also acts in physician decision support [177, 230] and performs as a virtual health assistant. Patients make use of SMS to keep in touch with their family and friends during hospitalization [26]. Text messages can provide a missing link between a hospital and its field workers, patients, support group members, or community health workers, wherever they may be [147]. The trends and developments in telecommunications have been reflected in an increasing utilization of cell phones and SMS in the health services [18]. SMSs are inexpensive in terms of cost and are applied quietly without disturbing other patients [26]. SMS also offers an alternative or supplementary social support to patients in hospital [26].

### **1.2.4 The high penetration of SMS, especially among the youth, promotes information dissemination**

It is important to take advantage of the opportunities provided by the penetration of mobile phones, especially among the youth, as a means of disseminating information. It is very easy to use this technology as a way of exchanging ideas [49, 97, 219], even across language barriers. SMS normalization is the only way to go for this to be achieved [57, 100, 225].

### 1.3 Problem definition

SMS communication manifests itself in many different forms. Getting a machine learning technique to be able to recognize the different expressions of SMS, presents a great challenge. The germane issues addressed in this research are:

1. Normalization of SMS for use in accessing information in a special domain, in this case a medical domain, with particular reference to information related to HIV/AIDS.
2. The use of normalized SMS for secure information access to a repository of FAQ.

### 1.4 Research question

To investigate the challenges posed by SMS text, especially for use in information access, the following research questions are articulated:

1. How should SMS text be normalized in order to retain its lexical and semantic meaning? and
2. How should the retrieval efficacy of the developed algorithm be measured using existing metrics?

### 1.5 Research aims and objectives

The aims and objectives of the research can be summarized in two parts:

1. The design of an algorithm for translating and normalizing SMS text, and
2. The design and development of a secure information access system, using SMS.

### 1.6 Research methodology

In order to achieve the research objectives, objectivism was adopted as the epistemological stance of the research and positivism as the theoretical perspective. The methodology employed is that of algorithmic approach and the methods are as follows:

- Content analysis, i.e. the analysis of related literature and of SMS content
- Pilot study, i.e. the prototype of the experiment at the early stage of the research served as a medium to collect SMS samples for testing

—Sampling, data samples were used unbiased from time to time to strengthen the robustness of the algorithm

—Experimentation was undertaken to find the algorithmic approach and the best way of testing the samples

—Statistical analysis was performed on the results in order to compare the efficiency of the algorithms.

## 1.7 Scope of study

The research will focus exclusively on:

1. Translating, normalizing and processing SMS in order to be able to formulate queries to be used in an information retrieval system; and
2. Evaluating the retrieval efficacy of the systems developed.

## 1.8 Contribution to knowledge

This thesis presents a novel algorithm using a combination of three important techniques for achieving SMS normalization: unsupervised noisy channel, character-based, and rule-based techniques. Recently, normalization has been achieved by aligning parallel texts using word-level, phrase-level and sentence-level approaches. This **S**earch, **C**ompare and **R**Eplace (SCORE) algorithm describes a new paradigm: a character-level approach based on the ideas of Church [55] and Li et al.[138] that character-level metrics correlate better semantically with human assessment or translation than do word-level metrics. This thesis makes the following broad contributions to the field of SMS translation to satisfy the need to include SMS in natural language processing.

—The algorithm resolves the issue of needing to search an SMS corpus in order to create a pairing of text messages with Standard English. It may be seen from the literature that SMS corpora are scarce. This presents an added difficulty in achieving pair training. SMSs were first created using the method of vowel stripping (Section 3.6.1).

—A SCORE algorithm was developed and evaluated. This is a three-technique normalization algorithm that is more robust, based on the methods of resolving ties between candidate words. Ties are resolved by the use of *word error rate* and *order of vowel precedence*.

—The advantage of the developed SMS normalization algorithm, using the three techniques mentioned earlier, is proved from the performance of the experiment comparing the developed algorithm, SCORE, with the existing algorithm, **BiLingual Evaluation Understudy** (BLEU). The results performed 23% better than the BLEU score, in terms of mean average precision (Section 4.2.6).

—The input text pre-processing stage of the SCORE algorithm allows a character that is repeated more than twice to be reduced to one instance only. This aids efficiency and saves time in the normalization stage. It is similar to that of Kaufmann and Kalita [123], where any repetitions of more than 3 letters are reduced to only 3 repetitions, for example, *coooooool* (*cool*). A repeated letter is described as being pre-processed in Aw et al. [19] but they do not mention whether it was reduced to 2 letters or 1 letter.

—An algorithm was designed and developed for use in information access using SMS. The retrieval efficiency of the developed algorithm, *SMSql*, was compared with the existing algorithms—*tf-idf* and *naive*. The computational time was used as a metric for the retrieval efficiency of the SMS requests. The results show an improvement of 10% and 4% on computation speed when compared with *naive* and *tf-idf* respectively.

—A fast, accurate and efficient algorithm, measured in terms of precision, was developed to retrieve answers from SMS requests in an SMS-based FAQ system.

## 1.9 Thesis outline

The research carried out in this thesis is explained in five chapters. Chapter 2 reviews the literature of existing SMS normalization techniques, SMS classification, SMS-based information retrieval systems, different theoretical frameworks needed to achieve the research objective, and the SSH protocol used to secure the sending and receiving of SMSs.

Chapter 3 describes the research design and methodology used in developing the algorithms for (1) SMS normalization and (2) the mobile accessing of information through the use of normalized SMS text, i.e. SMS-based information retrieval systems. The overview of the research design is presented to justify the choice of specific experimental and algorithmic methodologies.

In Chapter 4 the evaluation of the developed SCORE and SMSql algorithms is offered. The results obtained from the performance of the SMSql algorithm were compared with



---

existing retrieval algorithms. The SMSql algorithm retrieves responses, using SMS enquiries, from an FAQ question and answering system. Finally, statistical evidence to justify the significance of the developed algorithms is presented.

The thesis concludes in Chapter 5 with a discussion of the contribution made by this research as well as recommendations for future work.

## Chapter 2

# Review of Previous Research

### 2.1 Introduction

This chapter presents the content analysis required by SMS normalization techniques and an SMS-based information access system. Also discussed are the various metrics used in achieving the research objectives of SMS text translation as a means of accessing information. The suitability of the SSH protocol as a means of ensuring SMS security is reviewed. The review also shows the various methods that have been adopted in solving the problem created by SMS developing its own jargon. This chapter is divided into fourteen sections. In Section 2.2 an SMS classification is presented. Normalization techniques are described in Section 2.3, and Section 2.4 reviews normalization with specific reference to SMS. Text entry errors and similarity measurements are covered in Sections 2.5 and 2.6 respectively. Sections 2.7 and 2.8 cover least character distance and vowel selection, using a rule-based approach. The scope of SMS lexical normalization is discussed in Section 2.9. Section 2.10 explains SMS normalization and mobile information access. An SMS-based FAQ information retrieval mechanism is considered in Section 2.11. The keyword extraction mechanism is discussed in Section 2.12. Section 2.13 handles SMS security and the Chapter is summarized in Section 2.14.

### 2.2 SMS classification

SMS classification is the process of grouping text messages according to syntactical structures which are common features of SMS messages. One feature is a lack of grammar. Idiosyncratic spelling also makes SMS classification a difficult task [79]. Fairon and Paumier [85] gathered a corpus of 30,000 text messages for classification and research purposes. Kobus et al. [127] developed a system for characterizing and classifying SMS

messages based on their deviations from the orthographic norm, as well as on their unconventional use of alphanumeric and other text symbols. Cook and Stevenson [59] used 400 texting forms paired with their standard forms for SMS classification purposes. The 400 texting forms are differentiated according to type and frequency. Recent work by Gadde et al. [91] generated controlled noisy SMS text from regular English, using SMS features such as phonetic substitution, character deletion, typing errors, word dropping and word merging.

From the SMS corpus gathered for this research, six categories of SMS have been identified.

(1) *Acronyms* and *abbreviations* are identified through the extraction of the first letter of each word in a phrase. The extracted letters are then combined, and may be pronounced as one word, for example, *as soon as possible* (*asap*), *acquired immune deficiency syndrome* (*aids*).

(2) *Clippings* involve the deletion of letters from the original word regardless of the position of the letter. Letters can be deleted from the front (*initial clipping*) e.g. *examine* (*xamine*), from the middle (*medial clipping*) e.g. *breastfeeding* (*bfeeding*), and from the end (*end stripping*) e.g. *discharge* (*discharg*), or in multiple positions in the word (*mixed clipping*) e.g. *treatment* (*trtmnt*). There are two things that may happen when words are clipped; the word-length of the original form will be longer than the clipped word and also the retained letters of the short forms are preserved in the same order or position as those in the original form [27]. Other clipping methods are *g-clipping* e.g. *saying* (*sayin*), *h-clipping* e.g. *what* (*wat*), *prefix-clipping* e.g. *yourself* (*urself*), *vowel dropping* e.g. *resident* (*rsdnt*) and *suffix-clipping* e.g. *laboratory* (*lab*) but they all appear as character deletion [91].

(3) *Contractions*. Here, words are merged together by deleting the spaces between multiple words e.g. *thank you* (*thanku*), *good for you* (*good4u*). In this case there is a need for text segmentation to achieve normalization. Text segmentation may be generated from the most frequent *bi*-grams, with the assumption that it may still be understood by the users even if the space between the two terms is missing [91].

(4) *Phonetic substitution*. A text message is written exactly in the way it is pronounced. The term used in the SMS bears a sound similarity to the full English form. The text word is mostly shorter than the original word, often with a foreign character that is not part of the original spelling, e.g. *photo* (*foto*), *night* (*nite*).

(5) *Homophones* (*Accent stylization*) comprise English alphanumeric terms that exhibit identical pronunciation with words or parts of words being used to replace words or

letter sequences within the word, e.g. *grate* (*gr8*), *see* (*c*), *to*, *too* (*2*), *information* (*in4mation*).

(6) *Typing errors* are another category common in SMS writing. These usually occur when there is a transposition of two close letters, e.g. *wrie* (*wire*), one additional repeated letter e.g. *forr* (*for*), one letter missing e.g. *beter* (*better*) and one additional wrong letter e.g. *beauxty* (*beauty*). The correction of these misspellings can be generated from correct spellings by a few simple rules of comparison and replacement. About 80 per cent of all spelling errors are as a result of the transposition of two letters, one letter extra, one letter missing and one letter wrong [65]. Several spellcheckers have been written for the sole purpose of checking the typing error [186].

## 2.3 SMS normalization techniques (supervised, semi-supervised and unsupervised learning approaches)

In order to normalize SMS text, a set of noisy SMSs and the corresponding clean versions will be needed [142]. This is referred to as *supervised normalization*. The clean texts are mostly manually generated and both sets together are referred to as the *training pairs*. A machine-learning algorithm works on the pair of *clean* and SMS text, (e.g. a pair of *child* and *chld*), in order to learn the normalization model [68]. The learning process involves setting a conditional probability as a model that the SMS word  $w$  is actually a variant of the cleaned word  $c$ . The conditional probability,  $P(c|w)$ , is the possibility of having a clean word  $c$  from any wrong words  $w$ , for example, *together* may have been learnt from SMS words *tgher*, *togeda*, *2geda*, *tgther*. The learned model uses the clean term to normalize the noisy input SMS and produce the clean text in a process referred to as the statistical machine translation (SMT) model [57]. The algorithm may be as simple as replacing  $w$  with  $c$  by considering the maximum probability value involving  $w$  in the set of training pairs. The selection of the training pair involves getting a word alignment and mapping the training pair for each word in the noisy version to a word in the *clean* version. Word alignment probabilities are used to populate word-to-word mapping probabilities between the SMS text and the full terms.

Statistical machine translation [57] is a general way to normalize SMS text [20, 57, 127]. This may be accomplished by a *supervised* or an *unsupervised* approach. The former involves understudying and learning the training pairs using any available machine learning paradigm. Such a paradigm learns from observation or data, without a teacher, in order to classify observed objects and situations; or else it learns using data instances and generalization [161]. Generalization means that the system will be able to perform

well even with unseen data instances [204]. The SMT system translates a sentence from one language to another. An alignment set learns a mapping of words and phrases between the two languages, using a training corpus of two parallel sentences. During testing, the mapping is used along with language models to translate a sentence from one language to another.

*Supervised* learning generates a function that maps inputs to desired outputs. The desired output is referred to as a *label* and there are expert generated samples. This method is very familiar in the field of SMS normalization [20, 54, 127] but it is not robust with new words [184] and consumes time in that it involves hand labelling of the training pair [142]. The pairing of an SMS corpus with corresponding standard forms is relatively scarce [57] and therefore not readily available for experimental use. Supervised normalization depends on hand-annotated data, which necessitates the categorization of noisy tokens. The categorization process leads to three further problems: (1) cost, (2) the difficulty in establishing a standard taxonomy, and (3) the optimization problem for different category-specific models can compromise the performance of the system [59].

The *unsupervised* SMS normalization process [3, 4] involves the selection of possible *clean* tokens from a previously obtained weighted list. The list offers a possible translation of the *noisy* token or sentence. The clean token or sentence is then obtained by maximizing the product of the weighted lists and the language model [57, 59]. The list stands as a cluster of inputs. *Labels* are not known during the training process, unlike in the supervised approach. For example, *bat*, *bet*, *bit*, *bot*, and *but* all have equal chance to be the clean variants of an SMS token *bt*. There are criteria to be set through knowledge discovery and associated rule learning and data mining activities [86] before the final selection can be chosen. Data are neither labelled nor mapped in the *unsupervised* approach, which instead uses knowledge of the linguistic properties of SMS creative word formations. Such words have the potential to be adapted for normalization of text in other similar genres.

Creativity in SMS text is observed as the product of a small number of specific word-formation processes. Rather than capturing the errors using a generic error model, which is a familiar approach in the *supervised* learning paradigm, a mixture model is used, in which each word formation process is modelled explicitly according to linguistic observations specific to that formation. A generic error model is an error classification scheme that focuses on cognitive factors (skill, rules and knowledge) in human actions, as opposed to the environment or other context-related factors [84, 196]. A mixture model is a combination of *supervised* and *unsupervised* learning techniques i.e. *semi-supervised*. It is referred to as a *semi-supervised* model and is used where a problem arises of employing a large unlabelled sample, to boost the performance of a learning

algorithm when only a small set of labelled examples is available [28, 195]. The reason for making use of both labelled and unlabelled data for training is that typically a small amount of labelled data with a large amount of unlabelled data can yield considerable improvement in learning accuracy [168].

Any normalization model based on a spellchecking approach has the shortcoming of placing excessive confidence in word boundaries [25]. Available spell checkers, natural language processing (NLP) algorithms and tools have been found ineffective in translating and analysing SMS text accurately [143, 235]. The level of noise in microtext has crippled the efficiency of the famous tool, Named Entity Recognition, which yields a lower performance on noisy texts than on structured text [62, 172]. Named Entity Recognition focuses on the way to locate and classify atomic elements in text into pre-defined categories [173] such as the names of persons, organizations, locations, and the expressions of times, quantities, monetary values or percentages. Normalization is very important to retrieve or mine data from microtext, so that it becomes more readable for machines and humans and more suitable for further treatment using standard NLP tools [143, 235].

The approach of Beaufort et al. [25] towards SMS normalization is based on an SMS-to-speech synthetic general architecture system using spellchecking and machine translation approaches. SMS models were trained from an SMS corpus aligned at the character-level, rather than the word-level, in a *supervised* paradigm, in order to get parallel corpora. Character-level training considers each of the characters of the SMS input for its normalization process. Word-level training identifies the delimiter, which is not mostly present at the character-level, as words consist of at least one character [138]. There were two spellchecking type modules surrounding the SMS normalization module: the first one detected unambiguous tokens and the second part identified non-alphabetic sequences of characters and labelled them with their corresponding tokens.

## 2.4 SMS normalization: a review

SMS normalization approaches are different and have resulted in the customization problem [20]. The customization problem is a situation in which text messages are adapted or modified by the language model (LM) of an existing translation system [20]. The LM is defined as a function of assigning a probability distribution  $\Phi(w_j)$  to a sequence of the  $n$  words, having considered an earlier word; this is referred to as an  $n$ -gram LM [33, 37]. The simplest form of LM, the *unigram*, estimates each word independently in a sentence and assigns values disregarding all other context conditions. But the complex LM type

(*bigram, trigram, quadgram* LM, which is important in spelling correction, speech recognition and machine translation) will have to consider the probability of the surrounding words as a condition for upholding its semantics. A probability distribution will enhance any solution to the customization problem, as texts are grouped or customized into their corresponding original terms or texts for the SMS normalization process. The most recent token,  $(n - 1)$ , will be relevant when predicting the next word,  $n$ , and the probability increases with early translation in the search process of text normalization [33, 212].

The following are categories of SMS normalization.

### 2.4.1 Noisy channel model

The noisy channel model is based on two components: a source and a channel model. This model attempts to find the most probable word sequence given an observed noisy message [19, 59, 215, 231, 235]. For instance, if an English sentence  $c$ , of length  $N$ , is corrupted by a noisy channel to produce an SMS message  $w$ , of length  $M$ , the English sentence  $c$  could be recovered through a *posteriori* distribution for the same channel target text given the source text  $P(w|c)$ , and a *a priori* distribution for the channel source text  $P(c)$ . Usually there will be an alignment between the English and SMS words,  $c$  and  $w$  respectively. The two words can then be compared in terms of their alignment [20, 54], for instance in a sentence.

$$\tilde{c}_1^N = \arg \max_{c_1^N} \Pr(c_1^N | w_1^M) \quad (2.1)$$

In equation 2.1, the posterior probability is then derived from its alignment to the original message. This normalization technique uses the orthographic edit distance algorithm. As a *supervised* model, it uses a web crawler to generate automatically a large volume of noisy data for training and spelling suggestions [231].

The three major setbacks or challenges with this technique are as follows:

- (1) By using word substitutions for non-standard acronyms, *lol* could not be changed to *loyal* or *lobola*; (but rigidly substituted for *laughing out loud* or *lots of love*) and *tlphne* may assume the form of *telephone*.
- (2) There may be an insertion of a flavour or synonym word which takes the same supervised process for its normalization
- (3) There may also be omissions of auxiliary verbs and subject pronouns [19, 54].

The noisy channel model is cumbersome, as corresponding SMS/English words need to be mapped before translation. Using the conditional probabilities  $P(w|c_i)$  for  $i = 1, 2, \dots$ , there will be a need to have the highest posterior probability of the SMS  $w$  given an English word  $c_i$ . The translation may be wrong because one SMS word may stand for several candidate terms (e.g. *rpt*  $\rightarrow$  *report, repeat, repent, repute*, etc.). There is no certain way to determine the right translation for the source text. A confusion set will therefore be generated [95], from which to identify the right normalization candidate for a given lexical variant may be difficult. The input text has first to be pre-processed to remove erroneous text and this is an additional time consuming stage.

### 2.4.2 Phrase-based normalization

Phrase-based normalization is a statistical modelling approach that is comprised of three stages: word modelling, training and decoding [20]. The one distinct advantage to this approach is that there is no need to adapt the language model of the machine translation system for each SMS term. Phrase-based normalization involves splitting sentences into their  $k$  most probable phrases. The use of phrase-based normalization, as opposed to word-based normalization, enables incorporation of contextual information into the translation model and thus improves lexical affinity and word alignment. While this model is, in general, satisfactory, phrase-based normalization does not easily handle the lexical flexibility in SMS messages and lacks character-level analysis [111].

Phrase-based statistical machine translation uses a statistical algorithm to decide the most likely translation of an SMS word, that is, the string with the highest probability. The basic approach of phrase-based translation is the segmentation of the given source sentence into units (phrases), then the translation of each phrase and finally the compositions of the target from these phrase translations. Phrases are taken as sequences of words [12, 129].

For instance, given source string  $s_1^k = s_1 \dots s_k \dots s_K$  to be normalized to a target string  $t_1^j = t_1 \dots t_j \dots t_J$ , the highest probability string can be calculated as:

$$\check{t}_1^j = \arg \max_{t_1^j} \Pr(t_1^j | s_1^k) \quad (2.2)$$

where  $J$  and  $K$  are the number of words of the target and source sentences respectively. Equation 2.2 represents the maximum probability value obtained from the individual set of the target string,  $t$  (phrase or sentence), when it is mapped with the corresponding set of the source string,  $s$  (phrase or sentence). The challenge in this model is in deciding the translation of the SMS term. It may be unfortunate that the string with the highest



probability value may not be the right translation of the SMS phrase or word, especially when there are many candidate words.

The challenge also includes the use of a large annotated corpus in the *supervised* learning method, since the learning is performed at the word level. Phrase-based machine translation cannot suggest a match for an informal text that did not appear in the training set. The effect of this is felt greatly in a domain where new words turn up frequently and irregularly. Contextually, phrase-based machine translation is better than both word-based machine translation and character-based normalization, but lexically it is not a good choice.

### 2.4.3 Character-level MT

This technique uses letters that reconstitute the word or phrase by mapping SMS terms with corresponding English words in the order in which they are written. According to Oliva et al. [175], the order of arrangement of SMS characters is the same as that of English words. Similarly, Pennell and Liu [185] approach SMS normalization using a two-phase method, character-level and word-level methods. The system learns to map between character-level phrases in both languages. Previous research work has been centred on word-level [54, 228], phrase-level [20] or has been statistically-based [59, 63]. Character-level normalization focuses on modelling words formed by dropping a character from the original text. Text normalization techniques expand the number of possible abbreviations found in SMS text by using a machine translation system trained at the character-level in the first phase. A translation model that ignores the position of an abbreviated character in the formation of a word shows little degradation compared to trained, type-dependent models. An advantage here is that abbreviated forms are recognized independently of their position, but are later decoded for the final prediction.

Part of the challenge of this approach is that the contextual information for this model is not realistic; it works only on abbreviations and may not perform well for out-of-vocabulary (OOV) words, especially proper nouns, as most proper nouns are OOV. The system learns character-level mapping and performs better with the new abbreviations than a word-level system. A further challenge of this method is that it uses a pair of terms as the annotated data for the training (i.e. it is a *supervised* approach). The pre-processing stage is an additional cost for the translation process. In the pre-processing stage, the technique does not consider the deletion of repeated characters; hence such SMS words featuring repeated characters may not be normalized.

#### 2.4.4 Character-string-based

In a recent publication, Xue et al. [235] addressed the problem of normalizing micro-text using the source-channel model. Microtexts are text generated through computer-mediated communication. Four important factors are considered in carrying out micro-text normalization.

They are:

- (1) character-string-based typographical similarity,
- (2) phonetic similarity,
- (3) contextual similarity, and
- (4) popular acronyms.

The first factor is concerned with string-based normalization. It normalizes a micro-text term to its corresponding full terms with a one-to-one character mapping, but the challenge is to determine the equivalent term or a sequence of terms for each microtext term. The corresponding terms may or may not have the same meaning as the micro-text term. The second factor handles distortions caused by pronunciation; it determines similarity between two phonetic terms on the basis of phonetic representations instead of orthographic forms. The greatest challenge is the difference in regional pronunciation [105, 193]. The third factor concerns context, which provides useful clues in finding the most likely selection of a normalized candidate. Microtext terms may have to consider the  $n$ -gram probability of the surrounding words in order to determine their context. This model will work only if the surrounding terms are already normalized. The last factor, acronyms, involves word-to-phrase mapping. Due to the dynamism of acronyms, it is very difficult to create an exhaustive list: new acronyms spring up daily [142]. The model outlined in this section performs better, using the same data set, when compared to the baseline algorithms of *Aspell* and *Moses* [54].

#### 2.4.5 Letter transformation model

A unified letter transformation approach that will not require pre-categorization and human supervision was proposed by Liu et al. [142]. The model generates OOV from a dictionary using a sequence-labelling framework, where each letter in the dictionary word can be kept, eliminated or exchanged with other digits or letters. A large set of noisy, training word pairs were automatically collected, using a novel web-based approach, and aligned at the character level for model training. Character-level alignment for model training, using a set of noisy training pairs, was performed in order to form a

non-standard token. Each letter in the dictionary word can be labelled with: (a) one of the 0-9 digits, (b) one of the 26 alphabetic characters, (c) the null character “\_”, and (d) a combination of letters. The automatic learning process involves dictionary words being changed to non-standard tokens by a sequence-labelling framework that integrates character-level, phonetic-level, and syllable-level information [142].

The following features are used to create the non-standard token:

- (1) the relative position of the focused character  $c_i$ , in the word (character  $n$ -grams) and in relation to other character positions  $c_{-2}, c_{-1}, c_0, c_1, c_2$ ;
- (2) the use of three features (phoneme  $n$ -grams) to specify whether the current, previous or next character is a vowel  $p_{-1}, p_0, p_1, (p_{-1}, p_0), (p_0, p_1)$  and
- (3) the relative position of the current syllable in the word will determine whether the character is at the beginning or the end of the current syllable.

The conditional random fields (CRF) model was used to perform the sequence labelling. This model will first generate a set of variants,  $S_i$ , by varying the repetitive letters e.g.  $C_i = correct, coorrreeect, cooerrreeect, cooerrrrrect$  for  $T_i = cooerrrrrrrect$  are transformed to a set of variants. Then the maximum posterior probability is selected from among all the variants,

$$Pr(T_i | S_i) = \max_{T_i \in C_i} Pr(T_i | S_i)$$

The repeated character in the emotional expression is reduced to 1 character, e.g. *freeeeeedom* → *fredom*. This may not give the expected translation, as may be seen in the example. The system also depends on a simple rule to recover possible original words by substituting digits like 2, 4, and 8 in a supervised manner. The Liu et al. [142] approach aligns the letters of the longest common subsequence (LCS) between the dictionary word and the variant of the OOV. This gives letter-to-letter correspondence in common subsequences. The letter-transformation model uses the supervised learning approach, and therefore needs categorization. There are three advantages gained from not categorizing SMS: (1) the costs of labelling and timing are excluded, (2) the difficulty of standard taxonomy or SMS categorization is eradicated, and (3) system performance is standardized through integrating various categories of labelled SMS. It is very difficult to pre-categorize SMS text because of the number of variants that are generated by combining the insertion, deletion and substitution operations e.g. *tmrw*, *tmrrow*, *2moro*, *2morow*, *tmrw* etc. are generated from the English word for *tomorrow*.

### 2.4.6 Three-module architecture

Oliva et al. [174, 175] present SMS normalization in Spanish that involves three modules: pre-processing, translation and disambiguation. This idea is similar to the work of Aw et al. [19], who presented their SMS translation system as having two modules—normalization and translation modules; the latter moderates the input text and the former handles the translation. Aw et al. [19] further performed pre-processing of the extraneous text, through a conversion of text into lowercase and segmenting sentences into small units.

In describing the three modules, Oliva et al. [174] identified the pre-processing module as involving the SMS tokenization of alphanumeric characters from the corpus, and the uppercasing of SMS tokens that immediately follow a dot sign. The fact remains that texts in their raw form, however, are just sequences of characters without explicit information about word and sentence boundaries. Before any further processing can be done, a text needs to be segmented into words and sentences. The process of achieving this is referred to as tokenization [17]. Spanish SMS texts are compared with a Spanish dictionary if it is available; otherwise the system breaks a single word into tokens. The pre-processing module splits alphanumeric SMSs into simpler tokens if a word cannot be found in the Spanish SMS dictionary (e.g. *2telfs* is broken into *2* and *telfs*).

The translation module in Oliva et al. [174] allows all possible translations for an SMS word. This is achieved by the use of the two dictionaries (SMS and Spanish). SMS words are categorised into phonetic and non-phonetic abbreviations and real words. The SMS dictionary contains non-phonetic abbreviations while Spanish phonetic dictionaries contain both phonetic abbreviations and real words. The outputs consist of a list of possible translations from the two dictionaries. The translation module tries to discover whether the SMS word is a phonetic abbreviation of a real word by first searching the Spanish phonetic dictionary. The translation module in Aw et al. [19], however, uses a translation engine that consists of a set of rules, based on linguistics fundamentals, kept in the database for the translation process. The stage involves analysis, transfer and generation using both rules and dictionaries for every step of the transformation.

Oliva et al. [174] worked further on the disambiguation module, which performs lexical and semantic tasks using an open-source suite of language analysers (*Freeling 2.1* and *WordNet 2.3*) to resolve the ambiguities that still remain. In case a list of tokens still appears after all three modular stages, the most probable part-of-speech tag is selected as the most likely SMS translation of the token. The shortcoming of this technique is that it is limited only to Spanish and it is domain specific. There is the chance of errors being introduced at the evaluation stage when the translation is done for each token. If

the expected translation or real word is not included in the list of possible translations given by a particular module, the possibility of errors is increased because early error detection in a module does not stop the translation process as it continues to subsequent modules.

### 2.4.7 Statistical and rule-based modelling

This model uses the combination of statistical and rule-based techniques to normalize short text [63]. This normalization approach is based on a statistical machine-translation system which translates noisy data into clean data. The rule-based approach of this model becomes stronger as the number of automatic machine translations it encounters increases. More conditions may need to be defended, especially if the model includes an automatic spellchecking system, which can both extract rules for a manually constructed correction corpus and apply rules to correct errors of spoken text [43]. This work is similar to Aw et al. [20] in which a phrase-based statistical MT system was trained, as described in Section 2.4.2, and a translation dictionary was used to extract automatic rules for normalizing the text. Mostly the application of rule-based techniques comes as the result of disagreements in alignment procedures between the source and target components. This method involves further filtering procedures to make the alignment process perfect. Filtering procedures are normally implemented to extract good correction rules and to discard or reject noisy units. The area of similarity with the recent work of Costa-Jussá and Banchs [63] is in the use of a dictionary and rules, as with Liu et al. [143] and Oliva et al. [174]. The noticeable deficiency of this model is the small size of the dataset. This limits the testing to the available dataset only, and the system may not be robust in handling SMS beyond the data set.

### 2.4.8 Spelling error and its normalization/correction

Spelling errors and their normalization/correction have been identified as a major issue in constructing SMS and its normalization. SMS normalization is very similar to traditional spelling normalization, which has a long history [235]. It is important, on one hand, to review spelling errors as a unique way of writing SMS, and on the other hand, also to consider the ways that SMS is corrected in order to justify the language lexically. Spelling errors occur when there is a deviation from a language-dependent set of strings by character substitutions, insertions and/or deletions. A comparison of such strings with the dictionary reflects the difference in the string arrangement. Spelling normalization consists of detecting and correcting the error. This is addressed both as an isolated-word error detection (e.g. *teh* for *the*) and word-correction where the words

are checked singly (e.g. the use of *from* instead of *form*, i.e. real word error). The problem is that the system may not work for the case where a spelling error produces another correct word, for example, *went* for *want* or *hole* in place of *hope*. The semantic interpretation of the context of a phrase requires grammatical analysis and is more complex and language dependent. It involves context-dependent error detection and normalization. Corrections are made by using various lists of suggestions from an isolated-word method before making a choice based on the context. In interactive spelling correction, the accurate word is chosen by the user. The correct word can either be chosen from the list of substitutions provided or can be automatically picked as the only correct word [70, 83]. Both text normalization and spelling mistakes can be handled statistically with a noisy channel model [35, 221].

Elmi and Evens [83] show limited detection of spelling errors in isolated words. In a series of tests, the word  $W$  is selected for spelling normalization and replacement of the word is done (if it is misspelt), from the lexicon of words close to  $W$ . This is done by considering likely replacement words. The context of the sentence is used for selecting the most appropriate words. Syntactic and semantic information can assist in the selection, as well as phrase look-up. The omission of a character in some words may bring another meaning to the generated word. Common identification of such errors are based on algorithms such as reverse order (for example *haert* instead of *heart*); missing character, (for example *hert* instead of *heart*); added character (for example *hheart* instead of *heart*); and character substitution (for example *huart* instead of *heart*). All these depend on the *edit distance*. Edit distance counts the process of deletions, insertions and replacements that transform  $W$  into the correct word  $C$ . There is a weight assigned to the edit distance which takes into account the position of the character in error. When the character at a particular location, say  $n$  in  $W$ , does not tally with the character at location  $m$  of  $C$ , then we have an error. Normalizing (correcting) the SMS may involve some of these measures. For instance, omitting a character is a deliberate option of SMS users especially when vowels are stripped off e.g. *frm* may be counted as an error for all of the following: *form*, *from*, *farm* and *firm*. There may be further confusion on the way to making an accurate correction. This is another challenging issue that cannot be resolved by spelling correction or machine translation. The SCORE algorithm takes a special look at this situation and provides an alternative solution through a rule-based approach called the *order of vowel precedence* (in Sections 2.8 and 3.5).

Detecting non-word spelling-errors involves looking up the word in a set of all likely words, but a large database of possible words may contribute to the problem because of space, search time and contextual information. A further problem is posed by the presence of what are referred to as *real word errors*. These are words spelt correctly but

not intended by the user e.g. *then* for *than* or *theme* for *team*. Spellchecker algorithms may not detect this error unless there is an identification of tokens that are semantically unrelated to the context, or a lack of spelling variants of words that would be related to the context [99]. The bigger the size of the lexicon the more esoteric words it contains, so increasing the probability of real word errors [70].

Text typing involves thinking of an idea and then collecting the characters together and typing the strings. Ideas will crystallize into thought. Spelling errors can occur when there is an attempt to negate with a prefix but uncertainty about whether it should be e.g. *un*, *in*, or *im*. The choice of *imperfect* instead of *imperfect* will throw up an error. Errors can also be created by the selection of the wrong suffix, as with *tragicly* instead of *tragically*. Interestingly, some errors come to be seen as another style of language as may be seen on social networks such as Twitter, Facebook chat as well as in Instant Messaging applications.

People commonly make spelling mistakes unless contextual information reveals which word is intended. The document type may help a user to recognize a correct usage. For instance, it would be more common for a document containing the keyword *bullet* to be associated with military than with religious issues. There is also the possibility of a typist guessing a spelling from ignorance of the correct term e.g. using *filanthropist* instead of *philanthropist*, *acomodation* instead of *accommodation*. Correcting this may prove difficult without contextual information.

## 2.5 Text entry errors: a review

Text entry error is defined as any textual discrepancy between the original and transcript text. A high proportion of text errors arise when a key adjacent to the correct one is pressed. The presence of a textual error is represented as a symbol, letter, space, or punctuation mark [205]. Further work on error studies looks at analysing word-level errors and labelling a word as incorrect if multiple errors occur within it. These word-level errors are classified depending upon whether the resultant *word* is a *real word* (an unintended but valid English word), or *random/nonsense* (the meaning could not be ascertained) [206].

Spelling errors can be introduced in word processing in many ways by users' deliberate or careless attitudes. Such errors can lead to consistent misspellings and are probably related to the difference between how a word sounds and how it is actually spelt, for example *in4mation* for *information*. *Typographical* errors are not very common in long

essays but their occurrence is not ruled out. The position of keys on the keypad or keyboard may help to generate errors that arise with finger movement [106]. An *interactive spellchecker* can also be helpful. In the simplest case, the checker remembers all tokens which the user has indicated *should be replaced* and the words with which the tokens are to be replaced. After the first such replacement, future occurrences of the misspelt token can be automatically corrected, although this feature might be undesirable in some instances. Analysis indicates that 80 per cent of all spelling errors are as a result of transposition of two letters, one letter extra, one letter missing and one letter wrong [65].

SMS text entry errors are unique because they deviate strongly from formal language or normal spelling. For example, a phrase like *What is* can have over ten SMS versions e.g. *Wat is, Wats, Watz, Whats, Whts, Wots, Wt s, Wt's, Wts, Wtz, Wht is, Wat's*. Auto-correction tools fail to recognize some of these words as they are not included in the dictionary. They completely deviate from the normal standard of the English language. The spellcheckers see almost all SMS writing as being mistakes. These are not considered as mistakes by texting language users as they are able to understand this type of communication. In this case there is a complete communication of information between the source and the target. Although there is a great advantage in spelling and grammar checkers, because they help users to correct spelling and grammatical errors, they are irrelevant when it comes to the creation of the SMS.

In the early work of MacNeilage [146], text errors were categorised into four parts, each containing several subcategories:

1. *Spatial* errors consisting of horizontal, vertical, and diagonal subcategories describing the errant finger movement that may have caused the error in the process of text entry.
2. *Temporal* errors consisting of *reversal* (otherwise correct but reversed in order), *omission*, and *equivocal* (during the process of committing a transposition error, the participant realises their mistake and stops typing), and *anticipation* (when a character appears more than one keystroke ahead of where it should) errors.
3. *Miscellaneous* errors consisting of *interpolation* (an extra character with no relationship to the correct characters), *phonemic* (substitution with a character with a similar sound to the intended character), *type* (when a different but valid English word is formed as a result of the error), *contralateral* (when a substitution error involves the wrong hand and so is the mirror image of the intended character), and *dynamic* (an error in character repetition when the sequentially neighbouring character is repeated, i.e., *eroors* instead of *errors*) errors.



4. *Other* errors include *multiple classification* errors (when a single error fits the criteria of more than one of the other categories), and *unclassifiable* errors (that fit none of the other categories).

Other categories of errors which are initiated by touch typing errors were identified as coming from hitting a key with the wrong finger (finger hits multiple keys, wrong hand striking). There was also a characterization of errors that resulted from actions such as *insertion* (extra character), *transposition* (reversed order of otherwise correct characters), *migration* (correct character but in the wrong location), *interchange* (when two non-adjacent characters have been swapped), *omission* (missing character), *substitution* (wrong character), *doubling* (accidentally repeated character), and *alternation* (where alternating characters are reversed, for example *thses* versus *these*) [233].

Characters may be wrongly captured or they may be substituted by the writer of the text. These alterations are called *character errors* and occur through the use of an input device or in an attempt to write words or phrases in the exact way they are pronounced, for example, *b4* for *before*. Errors can also be generated by character misplacement (wrong position) which will invariably lead to wrong word alignment. The difference between character-errors and word-errors is discussed below.

*Character errors* are caused by text entry devices like keyboards, handwriting recognition applications, stylus typing. Speech recognition systems usually recognise the unit of input as a word. If the speech recognition software does not correctly recognise a word, then it gets the whole word wrong. In the speech recognition domain, the word-level error rate is the most meaningful measure of error [214].

*Word errors* involve counting the number of words with at least one error, unlike character errors which are determined by the exact number of errors committed through insertion, deletion, substitution, transposition, etc. Counting the exact error is insignificant in measuring word error [214].

## 2.6 Similarity measurements: a review

Similarity measurement can be explained in terms of the degree of commonality between two objects, X and Y, and shows the resemblance level between X and Y. Intuitively, objects X and Y are said to be similar based on their common features. The extent of their commonality denotes how similar they are. Conversely, the similarity between X and Y is related to the differences between them. The more differences they have, the less similar they are. The maximum similarity between X and Y is reached when they are both identical [140].

There are various methods used in calculating similarity measurements between two text objects but spelling is a major yardstick for this metric. Here, strings are compared with respect to their word length, manipulation, position and arrangements of letters. This research considers various methods to establish the accuracy and consistency of the developed algorithms, but the bottom line of all metric evaluations is string matching. Various methods involve matching identical substrings in the *word pair*. A word pair can be interpreted as two words in separate languages being considered to determine their status as cognates. Cognates denote words in different languages that are similar in their orthographic or phonetic form and are possible translations for each other [131, 208]. Several similarity measures are used to confirm string similarity, and some of them are discussed below.

### 2.6.1 Dice's coefficient

This is defined as the ratio of the number of shared character *bigrams* to the total number of *bigrams* in both words, for example *expirt* and *experiment* share two bigrams (*ex* and *xp*) so their Dice's coefficient is  $\frac{2(2)}{13}$  i.e. 0.31. It simply measures how similar two strings are in terms of the number of common *bigrams* (a *bigram* is a pair of adjacent letters in the string). It is also defined as a method of string similarity measurements for cognate identification that is represented in a set form [16]:

$$S = \frac{2|X \cap Y|}{|X| + |Y|}$$

where  $X$  and  $Y$  are strings.

This is a ratio of twice information similarity or intersection shared between two strings to the sum of the independent strings. The information shared could be a set of keywords, like in information retrieval. Since the definition relates to string similarity measures, the Dice's coefficient may be calculated for two strings,  $x$  and  $y$ , using *bigram concepts* as follows:

$$S = \frac{2n_t}{n_x + n_y}$$

where  $n_x$  and  $n_y$  are the numbers of bigrams in strings  $x$  and  $y$  respectively and  $n_t$  is the number of character *bigrams* found in both strings.

### 2.6.2 Longest Common Subsequence Ratio (LCSR)

This algorithm determines the longest common subsequence between the two strings. It is a measure of string similarity that takes advantage of the observation that parts of a string may be similar while the prefixes and suffixes (or any other part of the string) are not. It is computed by finding the longest substring in common between the two strings and returning the ratio of the length of that string to the length of the two words in the word pair i.e. it returns a value that indicates Longest Common Subsequence (LCS) for the string [112, 237]. This is the measure of the two words' cognateness or similarity [131]. Using the example of the two strings, *exprrt* and *experiment*, the Longest Common Subsequence Ratio (LCSR) of two words is computed by dividing the length of their longest common subsequence by the length of the longer word. In this case the LCSR for the two words is  $\frac{5}{10}$  i.e. 0.5.

For further generalization or clarification purposes, let us assume there are two strings  $X$  and  $Y$ . Each of them is formed by a sequence of simple English words (in our context, one is an English word and the other is an SMS word); e.g.

$$X = \{x_1, x_2, x_k, \dots, x_K\} \text{ where } x_k \text{ is the } k^{\text{th}} \text{ character in the String } X$$

and

$$Y = \{y_1, y_2, y_j, \dots, y_J\} \text{ where } y_k \text{ is the } k^{\text{th}} \text{ character in the String } Y$$

$Z$  is the common subsequence between strings  $X$  and  $Y$  if the elements of string  $Z$  belong to  $X$  or  $Y$ . It should be noted that the LCS uses dynamic programming to calculate the length of two strings. The words in this subsequence just need to appear in the same order as they appear in the other string [226, 237]. Therefore the LCS is a common subsequence having the maximum length and allowed to be non-contiguous. For example, the LCSR of *initat* and *initiate* is 0.75 in *Figure 2.1*, with the longest common subsequence as *initat*,

$$LCS = \frac{6}{8}; 0.75$$

Let  $X = \{x_1, x_2, x_3, \dots, x_i\}$  and  $Y = \{y_1, y_2, y_3, \dots, y_j\}$  be sequences and the LCS algorithm is described as follows:

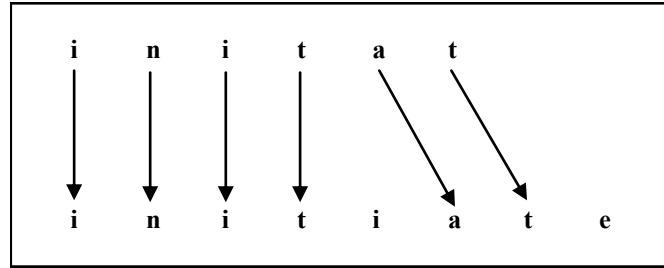


FIGURE 2.1: Schematic representation of LCSR

$$len[i, j] = \begin{cases} 0 & \text{if } i = 0 \text{ or } j = 0, \\ len[i - j, j - 1] + 1 & \text{if } i, j > 0 \text{ and } x_i = x_j, \\ \max(len[i, j - 1], len[i - 1, j]) & \text{if } i, j > 0 \text{ and } x_i \neq x_j \end{cases}$$

where  $len[i, j]$  is the length of an LCS in  $X_i$  and  $Y_j$

### 2.6.3 Word Error Rate (WER) or edit distance

WER is concerned with the amount of effort needed to convert an SMS string to its correct form. This is calculated as the minimum number of operations required to transform one string into another string [35, 137]. The intention is to find the smallest number of operations of replacement (R), insertion (I) and deletion (D) that can be applied on one string (SMS) and produce an error-free string (English). For instance, *expert* and *export* could stand as the same word with just a *replacement* of a letter, *e* to *o*, as this changes the first word to the second word. The word *heart* can be changed into *heat* by deletion of the *r* and on the other hand *heat* can be changed to *heart* by the insertion of *r*. The *word error rate* or *edit distance* of two strings  $S_1$  and  $S_2$ , is defined as the number of minimum point mutations required to make a change of  $S_1$  and  $S_2$ , where the point mutation is calculated from the operations of replacement, insertion, deletion and number ( $N$ ) of letters represented for the  $S_2$  [60, 61].

Transformation of SMS to English involves three operations which are illustrated in *Table 2.1* where some examples are given to establish the editing operations. For instance, to transform *Antirvral* into *Antiretroviral* the missing letters will have to be provided and placed in the right position. The comparison and exchange of the letters of the strings (SMS and English) are carried out using the edit distance. The minimum number of *insertion* operations to convert *Antirvral* into *Antiretroviral* is 5 (I=5), where all the missing letters *e*, *t*, *r*, *o* and *i* are inserted into the SMS word.

There also needs to be a *deletion* operation on repeated letters mostly common in exclamation expressions. This is an expression of feeling that is conveyed by the SMS.

TABLE 2.1: Transformation in RID operations

SMS word		English word	Replacement	Insertion	Deletion
Antirvral	→	Antiretroviral	0	5	0
Yeeeeesssss	→	Yes	0	0	9
Un4tun8	→	Unfortunate	2	0	0

Any repeated character within a string  $S$  that is greater than 2, is stemmed down to 1. At this juncture, it is worth noting that there needs to be a modification in calculating in the deletion operation in this research. For example the number of deletions in the second example (Yeeeeesssss→Yes) is 9 (i.e. 4e's and 5s's) from the repeated letters. The modification counts similar characters as one (1) operation, the deletion operation is achieved on 2 characters *eeee* and *sssss* as they are reduced to *e* and *s* respectively, therefore  $D=2$ .

*Homophony*: a common feature of homophonous words is that the digit or symbol with the common sound can be used interchangeably with the words or part of the words e.g. digits like *2* have sound of *to*, *too*, *two*; *4* sounds like *for*, *four*, and *8* has the same sound as *ate*, *eight*. An SMS texter uses these digits to replace English words. The digit is replaced within the string with the correct English form that is stored in the homophone table (*Appendix C*). In this research, for example, *2*, *4*, and *8* are exchanged for *to*, *for*, and *ate* in the SMS normalization application. The number of digits that are replaced is counted and this represents the number of replacements needed to translate SMS into more formal language. For instance, in the third example, *un4tun8* has 2 digits that are replaced with *for* and *ate*. The words are concatenated together to make *unfortunate*. In this case, the  $R=2$ . Other examples are *gr8*, *4low*, *2morow* that have *grate*, *forlow* and *tomorow*, where  $R=1$  for each of the examples.

Furthermore, in line with the editing operations, Aw et al. [20] found the top 10 most common substitution, deletion and insertion operations used in 700 messages that were randomly selected from 55,000 messages (see *Table 2.2*).

The messages were collected from chat rooms and correspondence between the university students. The results show that substitution accounted for 86.43% of the transformation, deletion 5.48%, and insertion 8.09%.

#### 2.6.4 BLEU and human judgement

BLEU (BiLingual Evaluation Understudy) is an algorithm for evaluating the accuracy and quality of language translation [45, 181], in this case the accuracy of the SMS text which has been normalized to its full English form. Accuracy is judged based on *success* (if the translation is the intended term), *failure* (if it is not) and *false positive* (a

TABLE 2.2: Top 10 most common substitution, deletion and insertion

Substitution	Deletion	Insertion
<i>u</i> → <i>you</i>	<i>m</i>	<i>are</i>
<i>2</i> → <i>to</i>	<i>lah</i>	<i>am</i>
<i>n</i> → <i>and</i>	<i>t</i>	<i>is</i>
<i>r</i> → <i>are</i>	<i>ah</i>	<i>you</i>
<i>ur</i> → <i>your</i>	<i>leh</i>	<i>to</i>
<i>dun</i> → <i>you</i>	<i>I</i>	<i>do</i>
<i>man</i> → <i>manchester</i>	<i>huh</i>	<i>a</i>
<i>no</i> → <i>number</i>	<i>one</i>	<i>in</i>
<i>intro</i> → <i>introduce</i>	<i>lor</i>	<i>yourself</i>
<i>wat</i> → <i>what</i>	<i>ahh</i>	<i>will</i>

translation that appears right but is not). Quality is considered to be the correspondence between a machine's output and that of a human. According to Papineni et al. [181], the *closer a machine translation is to a professional human translation, the better it is*. This is the central idea behind BLEU. Callison-Burch et al. [45] and Zhang et al. [243] recognized that BLEU has one of the first metrics to achieve a high correlation with human judgements of quality and remains one of the most popular automated and inexpensive metrics. Scores are calculated for individual translated words, phrases and sentences by comparing them with a set of good quality reference translations. Those scores are then averaged from the number of possible translation of different human judgment to reach an estimate of the translation's overall quality. BLEU's output is always a number between 0 and 1 [128]. This value indicates how similar the SMS texts and the parent terms translations are, with values closer to 1 representing greater SMS normalization. Few human translations will attain a score of 1. To achieve the highest score, the candidate text must be identical to a reference translation. It is not, however, necessary to attain a score of 1. Translation performance is better measured by comparing the closeness of a machine translation to professional human judgement. The MT quality is judged by measuring the closeness to one or more reference human translations according to a scale of relevance [181]. This research is fashioned against a highly successful *word error rate* metric used for lexical normalization of the SMS. The weighted average is used as the result of the translation. The translation is compared on *n*-gram on the SMS and references translation made by human judgement.

## 2.7 The Least Character Distance (LCD) calculation

The **Least Character Distance (LCD)** string comparison algorithm is based on character-level error analysis which is characterized by three major operations, namely replacement, insertion and deletion, which shall henceforth be referred to as RID. RID involves first aligning the reference-SMS character/word sequence with the recognized (English) character/word sequence using dynamic string alignment. SMS words such as *in4matn*, *hlth*, *Yeeeesss*, for example, will be translated to the formal language, e.g. *information*, *health*, *Yes* respectively, after some RID operations have been carried out. *Replacement* of 4 in the referenced SMS word (*in4matn*) will transform it to *informatn*. The insertion of missing characters *e* and *a* on *hlth* will give *health*, and repeated deletions of characters *a*, *o*, *h* and *!* in *Whaaaaoooooooohhhhhh!!!!* will result in the word *Whaoh!*. A similar approach was used by Brody and Diakopoulos [36], e.g. *niiiiice* → *nice*, and *realllly* → *realy* .

The LCD algorithm works on a reference character/word sequence, like *clndr*. The SMS string is aligned with any of the candidate words or character sequences, for instance *calendar*, *colander*, and *cylinder*. The three recognized words have equal probability in terms of character recognition, sequence of arrangement and alignment, which makes it difficult to choose between them. However, the general difficulty of measuring the performance of LCD lies in the fact that the recognized character/word sequences (supposed translation), sometimes differ in word length and spelling from the reference word sequence (supposedly translated). In this case the LCD algorithm not only faces the option of recognizing the correct SMS translation from a simple implementation of character/-word comparison between recognized and referenced words, but also more importantly gets the best normalization option for the reference word.

Similarly, LCD measures the minimum error rate of the similarity comparison in character combinations and order of positioning of SMS strings. Literature reviews [132, 133] confirm that SMS words are variants of a universal set of original English words having different input strings. For example, consider a single word, *tomorrow*, from the universal set of English words. This can have over twenty SMS versions (*tomoz*, *tomorro*, *tomorrrw*, *tomora*, *morrow*, *mora*, *tom*, *2mora*, *tomoro*, *2morrow*, *tmw*, *2mrow*, *2morow*, *2morro*, *2mrrw*, *2moz*, *2mrw*, *amoro*, *tomorrrrow*, *2moro*, *tmrrw*, *tomrw*). The translation of SMS variants into the Standard English form (*tomorrow*) is of utmost importance in SMS normalization.

In order to represent LCD as a percentage error rate, the following definition of text entry error rate is proposed, given an SMS text string and its various transformations into English variants,  $E_n$

$$\check{LCD} = \frac{R + I + D}{N} \times 100\% \quad (2.3)$$

where  $R$ ,  $I$ ,  $D$ ,  $N$  are the number of Replacements, Insertions, Deletions, and Word-lengths (English word) being referenced respectively.

Equation 2.3 is applied to each word of the English variants (candidate terms) of  $E_n$  i.e.  $E_1, E_2, E_3, \dots, E_k$ . Therefore each member of a set (each word) i.e.  $\{E_1, E_2, E_3, \dots, E_k\}$  will have its own LCD percentage when translating it to  $E_n$ . In general,

$$\check{LCD}_{E_n} = \frac{R + I + D}{N} \times 100\% \quad (2.4)$$

The LCDs of the variant are ranked and the smallest percentage error rate is taken as the best possible translation of the proposed algorithm.

Examples of LCD and error rate percentage are presented in *Table 2.3*. These were randomly selected from the research data set to demonstrate the LCD and the error rate as defined by Equation 2.4

TABLE 2.3: Examples of Least Character Distance and Percentage Error Rate

Ex.	SMS Word	Candidate words	N	R	I	D	LCD	% error rate
1	stdy	saturday	8	0	4	0	$\frac{4}{8} = 0.50$	50
		steady	6	0	2	0	$\frac{2}{6} = 0.33$	33
		stodgy	6	0	2	0	$\frac{2}{6} = 0.33$	33
		study	5	0	1	0	$\frac{1}{5} = 0.20$	20
		sturdy	6	0	2	0	$\frac{2}{6} = 0.33$	33
2	yeeeessss	yeast	5	0	2	2	$\frac{4}{5} = 0.80$	80
		yes	3	0	0	2	$\frac{2}{3} = 0.67$	67
		yesterday	9	0	6	2	$\frac{8}{9} = 0.89$	89
3	b4	beaufort	8	1	4	0	$\frac{5}{8} = 0.625$	62.5
		before	6	1	2	0	$\frac{3}{6} = 0.50$	50
		benefactor	10	1	6	0	$\frac{7}{10} = 0.70$	70
4	4wrđ	foreword	8	1	2	0	$\frac{3}{8} = 0.375$	37.5
		forward	7	1	1	0	$\frac{2}{7} = 0.286$	29
5	2moroooo	tomorrow	8	1	3	1	$\frac{5}{8} = 0.625$	62.5

The costs of translating SMS to formal English vary according to the number of editing operations performed. The least cost will always determine the choice to make. For instance, in *Example 1* an SMS word, *stdy*, has an equal chance of matching five candidate words (*saturday*, *steady*, *stodgy*, *study*, and *sturdy*) as the intention of the SMS sender



(i.e. the normalized form). There are various degrees of insertion that will be needed to translate the SMS word, with *study* offering the least. The smallest percentage error from the LCD calculation (i.e. 20%) and the technique of the SCORE algorithm make the choice to be *study*.

In *Example 2*, as explained in Section 2.6.3, the cost of the editing operation is incurred on the deletion of the repeated letters *e* and *s* in *yeeeessss* to give *yes*. The three candidate terms *yeast*, *yes*, and *yesterday* are selected from dictionary words. They have an equal chance of being the translation of the SMS text. *Yes* is selected as the normalized form from the three candidate words because it has the smallest percentage error rate.

The presence of a digit in an SMS word is a common feature of the homophonic nature of SMS. The digit is replaced with its most likely corresponding meaning in the database. The digit *4* is transformed into *for*. The new SMS term becomes *bfor* and is now put through the normalization process. Three candidate terms emerge (*beaufort*, *before*, and *benefactor*). The cost incurred in the transformation is mostly on two operations—Replacement and Insertion. The candidate word *before* has the least percentage error rate and it is chosen as the normalized form of *b4*. The fourth example is similar to the third.

In the last example, *Example 5*, the digit *2* is replaced with *to* from the database, forming another SMS word *tomoroooo*. The new SMS string is then normalized after the repeated letters *oooo* have been reduced to 1. The cost of editing the final SMS word, *tomoro*, is to delete the repeated letters and replace the digits and lastly the insertions of the missing characters. The only translation that emerged is *tomorrow*, which stands as the normalized word for *2moroooo*.

## 2.8 Vowel selection through rule-based approach

The rules of natural language enable us to represent knowledge [165]. In a rule-based expert system, the knowledge base includes *if-then* rules. In general, the condition part, the left-hand-side (LHS), of a rule can be patterned to match against the database. It is usually allowed to contain variables that might be bound in different ways, depending upon how the match is made. Once a match is made, the action part, right-hand-side (RHS), is executed. The actions can be adjusted arbitrarily by the addition of new data to the database, and the modification of old data in the database. The rule interpreter has the task of deciding which rules to apply. It decides how the conditions of a selected rule should be matched to the database conditions, and monitors the problem-solving

process. When it is used in an interactive program, it can turn to the user and ask for information that might permit the application of a particular rule.

Generally, English letters are arranged alphabetically in the dictionary. A dictionary is a book that lists the words of a language in alphabetical order and gives their meaning and other details about them [104]. In terms of traversing this order, such arrangement gives priority to a set of the letters in prefix position. This is observed in the arrangement of vowels as it takes the alphabetical order of *a, e, i, o, u*. With this arrangement *a* will be the most likely visited or favoured vowel whenever there is a search in the dictionary. Situations may arise, whereby this order may be distorted especially with the argument that there is a difference in the rate of usage of vowels in English words [145, 155] (see Section 2.8.1). A decision has to be taken about which vowel is likely to be parsed.

Decision trees are useful models that are based on self-learning procedures which sort the instances in the learning data by binary questions about the attributes that the instances have. It starts at the root node and continues to ask questions about the attribute of the instance down the tree until a leaf node is reached. For each node in the decision tree the algorithm selects both the best attribute and the question to be asked about that attribute. The selection is based on what attribute and question about it divide the learning data so that it gives the best selection in the classification system [180].

A decision tree can be viewed as a hierarchy of rules. Decision lists are a special class of decision trees. Decision lists may be the simplest model for hierarchal decision making. Despite their simplicity, they can be used for representing a wide range of classifiers [180]. When a classification is needed, the first rule in the hierarchy is addressed. If this rule suggests a classification, then its decision is taken to be the classification of the decision list. Otherwise, the second rule in the hierarchy is addressed. If that rule also fails to classify, the third rule is addressed, and so on. Often, programmers prefer presenting decision lists as sequences of *if-then-else* statements, intended for classifying an instance of an object.

The following attributes—completeness, consistency and continuity—are achievable [39, 88, 165] by applying rules. Completeness is attained if varying the input combination values results in one appropriate value as an output from one of the rules. Consistency is achieved in the sense that no contradiction results from the selection or combination of the rules. Continuity occurs when any change in the inputs would result in a smooth change of the output values.

Lastly, the rules are applied to the scenario of strings  $S_1, S_2, \dots, S_n$  having the same word error rate or LCD i.e.

$$S_1 = S_2 = \dots = S_n = WER$$

These strings can be disambiguated by ordering the words according to their content so that string  $S_1$  is preferred if it contains  $e$  and the rest do not (see Section 2.8.1). The next string  $S_2$  is selected if it contains  $a$  and the others do not, etc. The decision to choose  $S_n$  containing the vowel  $e$  is as a results of the vowel's usability and availability compared to others [145, 155]. This provides a shift in the normal presentation of  $S_n$  from the dictionary. For example, the normalization of  $sx$  can undergo the stages described in Table 2.4 where all the fourteen (14) words have equal probability to be the translation of  $sx$ . The LCD results came up with three terms ( $sax$ ,  $sex$ , and  $six$ ) tying i.e. they have the same LCD least results (0.33). The term with vowel  $a$  will always be favoured based on the alphabetical order of the dictionary. Since all the known algorithms give the results in alphabetical order the possibility of presenting  $sax$  is certain.

TABLE 2.4: Order of vowel precedence

SMS Word	English word	N	R	I	D	LCD	Vowel Precedence
sx	sax	3	0	1	0	$\frac{1}{3} = 0.33$	0.33
	saxifrage	9	0	7	0	$\frac{7}{9} = 0.78$	
	saxon	5	0	3	0	$\frac{3}{5} = 0.60$	
	saxophone	9	0	7	0	$\frac{7}{9} = 0.78$	0.33
	sex	3	0	1	0	$\frac{1}{3} = 0.33$	
	sexagenarian	12	0	10	0	$\frac{10}{12} = 0.83$	
	sextant	7	0	5	0	$\frac{5}{7} = 0.71$	
	sextet	6	0	4	0	$\frac{4}{6} = 0.67$	
	sexton	6	0	4	0	$\frac{4}{6} = 0.67$	
	six	3	0	1	0	$\frac{1}{3} = 0.33$	0.33
	sphinx	6	0	4	0	$\frac{4}{6} = 0.67$	
	spinifex	8	0	6	0	$\frac{6}{8} = 0.75$	
	suffix	6	0	4	0	$\frac{4}{6} = 0.67$	
	syntax	6	0	4	0	$\frac{4}{6} = 0.67$	

The proposed algorithm references the frequency distribution of English letters with emphasis on the vowels. The fact is, however, that  $e$  is the most common vowel in English words and should be the most likely character to be considered, therefore  $sex$  is chosen as the normalize term for  $sx$  (see Section 2.8.1). As a second example, consider the SMS classification for consonant skeletons, that is, vowel-stripped SMS word,  $bg$ , the likely candidate terms are ( $big$ ,  $bag$ ,  $beg$ , and  $bug$ ). The word error rate for these words becomes 0.335 and there is a difficulty in selecting which one should come first.

By applying the rule of order of vowel precedence in the proposed algorithm tying is eradicated.

### 2.8.1 Order of vowel precedence

In a frequency distribution list of English letters, *e* has been identified as the most used letter in forming English words. Six instances will be considered to support this argument. For instance, (1) in this chapter the distributions of vowels are as follows: *a*=6486, *e*=10571, *i*=6090, *o*=6103, and *u*=1883, with the proportion of *e* being almost double that of *a*, and (2) the statistics of the distribution of the vowels in the whole thesis *a*=19635, *e*=31270, *i*=18271, *o*=17829, and *u*=6101. This confirms the proportion and usage of these vowels.

Mackenzie and Soukoreff [145] recorded the most frequent letter as *e*=1523, *t*=1080, *o*=1005, *a*=921, *i*=829, so *e* is far more frequent than any of the other vowels. The result correlates with letter frequencies [155]. Several other studies that have made reference to this work include the study of finger-based text entry for mobile devices with touch-screens [102]. The fact that *e* is the most frequent is also substantiated by the word game of Scrabble where the relative frequency of *e* is 12% compared to the other 25 letters [118].

The frequency varies according to the language. For example, in Turkish, *a* is the most used vowel, as it is also in Italian where the ranking is *a*, *e*, *i*, [139]; but in English, the ranking of the letter usage is *e* followed by *t*, *a*, *o*, and *i* while the least frequent are *q*, *z*, and *x* [32, 38]. The most frequently used words in the 500-word article of Mackenzie and Soukoreff [145] are: *the*=189, *a*=108, *is*=85, *to*=57 and *of*=54; the frequency of the word *the* also helps to support the claims for *e*.

Letter frequency is frequently used in data communication and encryption [126, 139]. Encrypted text is sometimes achieved by replacing one letter by another, but to start deciphering the encryption, it is useful to get a frequency count of all the letters. For any encrypted text the most frequently used character is \* and standard compression algorithms can exploit this effectively [90].

In the alphabetic layout of the mobile keypad, Mittal and Sengupta [164] propose frequently used English words from a dictionary and attempt to minimize the number of matches for any given numeric key combinations. They optimize multi-tap usage in order to reduce tap frequency for commonly used alphabets. *Morse code* principles were adopted to assign small sequences to commonly used alphabets in English language. The

letter *e* was identified as having the highest relative probability value from the table of alphabets and their probabilities.

Fox [89], in an experiment using over 1 million Brown corpus terms taken from a broad range of literature in English, produced a list of 421 stop words, with *the*, *to*, *how*, *are*, and *what* as occurring at the highest frequency. The word *the* contains *e*, again supporting the fact that the letter *e* is the most used vowel.

In order to implement the order of vowel precedence, there is a need to introduce a rule-based system. As discussed in Section 2.8, a rule-based system is used as a way to store and manipulate knowledge and to interpret information in a useful way. In a rule-based system, much of the knowledge is represented as rules, that is, as conditional sentences relating statements of facts to one another, where if the *IF CONDITIONS* are true then the *ACTIONS* are executed [165].

## 2.9 SMS lexical normalization scope

The process of SMS normalization, in this research, is defined as processing only one word at a time. This means that the processor has to be fed a single token, that is, *tmrw* (*tomorrow*) but not *asap* (*as soon as possible*), as *asap* is assumed to be a multi-token word. Any lexical variant which is outside the dictionary (such as non-English words) will be considered outside the scope of the research. Abbreviations like *ARV* (*Antiretroviral therapy*), *RSA* (*Republic of South Africa*) are taken as acronyms and as such will be considered for text normalization. Although SMS should be formed freely from Standard English, acronyms and abbreviations are included to reduce the number of characters to be used in sending a message. Abbreviations such as *lab* (*laboratory*), *res* (*residence*) will be considered as single tokens that have a corresponding interpretation in Standard English. Abbreviations can therefore be taken as *in-vocabulary* (*IV*) or *out-of-vocabulary* (*OOV*). A *supervised* normalization technique assumes that the tokens are already labelled with their pairs [59]. For example, *frid* may be rendered as *friday*, but other words are equally possible as translations; it could mean *friend* or *fried*. This shows that the lexical variants of *supervised* normalization have already been identified. In this research, which uses an *unsupervised* approach, none of the tokens is identified as forming part of SMS-English normalization pairs. The proposed algorithm addresses the issue of identifying lexical variants from among many candidate words of possible SMS translation. A dictionary-based approach that decides the choice of the appropriate token among the lexical variants is proposed. This same proposal was recently made in Han et al. [95].

Phrase-based and word-based normalization have played a significant role in the SMS normalization process. The reviews show that lexical interpretation and semantic contextual information about SMS tokens arise, respectively, with those two modelling methods. Adjoining words constitute a phrase or sentence, and they contribute to the semantic interpretation of the item in question. Examining the atomic level of both sentence and word will help the research to focus on character-level SMS normalization. An *unsupervised* noisy channel method for SMS normalization is therefore proposed. This is based on a character-level mapping model. An *unsupervised* noisy channel method is cost effective because there is need neither for the use of a large corpus (for training) nor for the standardization of system performance [59]. Character-level normalization for translating SMS text into English is a new approach. Similar work was published by Pennell and Liu [183, 184] and Liu et al. [142].

The research aim is to output the normalization of SMS text through the use of high quality syntactic processing, combining rule-based systems, noisy channel models and character-based approaches. This may be achieved by using tokenization, character matching, word matching and replacement techniques in combination with a high quality, large scale English dictionary as a database. Typically, statistical machine translation systems are built with training materials that are sufficient for the training and data sets. The problem with this is that there is no corpus that might be used for such exercises [217]. Rule-based methods can be used to translate out-of-vocabulary words to their normalized form and human beings are good at text normalization because of their language proficiency [48, 191].

Several methods of SMS normalization require alignment of SMSs and their corresponding terms in natural languages, for training purposes. Mostly this alignment requires human expertise and may prove very difficult. A *supervised* technique that maps SMS and parent word together was studied by Aw et al. [20] and Kobus et al. [127] among others. The proposed algorithm (SCORE) approach recognises that errors and irregular language can be classified into several distinct categories (see Section 2.2) and therefore a multi-faceted approach will be the most effective way to deal with this problem. The SCORE dataset includes lists of: (1) frequently\_used\_SMS\_words; (2) acronyms/abbreviations; (3) punctuations/prepositions; (4) homophones; and (5) over 40,000 English\_and\_medical words. The entry to the database is either a single word or a phrase. In the review automatic letter-level alignment and letter-transformation using rule-based and unsupervised learning by comparing non-standard English to standard English/tokens have been examined. Algorithms of insertion, replacement and deletion will be applied. There has been a movement of SMS normalization from phrase, to word and character. Each of the methods has its strength and shortcoming, but this

research combines the advantages of every method to strengthen the performance of character-based SMS normalization.

## 2.10 SMS normalization and mobile information access

SMS-based information retrieval is a form of accessing information necessitated by the rapid development of mobile telecommunication. The technology is characterized by instant access to information as a response to SMS enquiries. A mobile search request is considered unique because of the restricted size available for the reply, so only a few results can be returned for any given query. Mobile search systems, for example, enable users to obtain extremely concise and appropriate responses from queries across arbitrary topics. Users may be forced to rephrase or reformulate the query if their answers are not made available in the preliminary pages of the search response. There is a limitation to what a mobile phone can download, compared with that available to a desktop search user. However, mobile search users rarely employ advanced search feature of the search engine but prefer to expend extra energy in reformulating the queries [182, 200].

Using a Google search as a benchmark, the typical results of an SMS-based search can be considered by using a query sentence—*wn d u intt arv thrpy*—extracted from an English query *when do you initiate antiretroviral therapy?*. Google responds only with a normalized form of *thrpy* as *therapy* and translates the abbreviation *arv* to *antiretroviral*. This is a usual experience for SMS users. Google appears to be the best search engine in terms of average precision and response time [80]. SMS query results mostly take the form of *Garbage In Garbage Out (GIGO)*, and as such are not helpful to the SMS user. Normally, when a user mistypes an input query, the system will suggest an alternative query sentence, in order to continue the semantic-based search [8]. Sometimes, suggestions made by the search engine are far from the intent of the SMS user (for example, in the search that was performed *wn d u* were joined together as *wndu*). SMS communication has made it difficult to build automated question-answering systems because of the many variants employed by SMS users.

SMS-based search benefits from small form factors, low bandwidth and a non-interactive model. A search response takes an average of a few seconds to several minutes [50]. Its growth has spread by leaps-and-bounds across all facets of human activities from hotel reservation [6], examination time-table scheduling [7] to agricultural marketing [110] just to mention a few. Mobile device users do not always have the privilege of reformulating a query or interacting with the search engine, as is common for desktop searching [211]. Most mobile search algorithms are known to have query terms in pre-defined topics as

keywords within the search space, or have specialized parsers to determine which topic is intended.

An SMS-based question-and-answer (Q&A) retrieval system accesses information in the form of questions and answers through the use of SMS services on the mobile phone platform. Q&A systems may appear in four guises: (1) *natural language processing* is a situation whereby users send a query in a natural language for enquiries on phones, and the answers are returned in a natural language; in (2) *human intervention*, messages are sent in the form of natural language to a particular agent. Normally, the agent who is an expert, gives the answer to the request; with (3), the *information retrieval method*, the corpus will be searched for a possible answer to the request and the answer may be delivered after the enquirer has responded to a further request from the machine, for instance, to type in a specific code to retrieve specific information; (4) *frequently asked question retrieval* offers a ready-made answer to every enquiry that may be requested from the user. The database is searched for the enquiry and an appropriately matched answer is returned. The second part of the research focuses on the FAQ retrieval system using SMS queries.

The FAQ system was able to be transformed into an SMS-based FAQ retrieval system. This is designed to give a set of FAQs for a query written in SMS language. An FAQ may be (1), *Mono-lingual FAQ retrieval* in which the FAQ and SMS datasets are in the same language and the challenge is to get the best matching between the two datasets. An FAQ may also be (2), *Cross-lingual* and here the FAQ and SMS datasets are not in the same language. The challenge here is to get the best match between two dissimilar datasets. The FAQ may also be (3), *Multi-lingual* where the FAQ and SMS datasets are many languages and the challenge is to get the best matching between various languages or datasets. In this thesis a monolingual SMS-based FAQ retrieval system is used for the research purposes. The algorithm is based on this platform too.

This research aims at showing how SMS can be used to access an FAQ system. Efficient searches and effective retrieval are the primary concerns of any information retrieval system. This is done by increasing the precision and rate of recall when enquiries are made. Precision and recall are two important metrics used in evaluating search strategies. The results of a search can be relevant or otherwise [120]. To examine the effectiveness of retrieval, the degree of relevancy of retrieved items is considered. Relevancy judgment can be binary (Excellent or Very Poor), or continuous (ranging from 0 to 1 i.e. Excellent, Very Good, Good, Moderate, Poor, Very Poor), depending on the user's judgement and satisfaction. It may be difficult to pass judgment because of four main issues: *Subjectiveness* makes the outcome depend upon a specific user's judgment; *Situational* relates to users' current needs, which are changeable; *Cognitiveness* depends



on human perception and behaviour which are also not stable; and *Dynamism* reflects changes experienced over time [22, 244].

Figure 2.2 shows how an SMS query is presented to a search engine. A normalized SMS is made to interface with the Q&A database. A set of documents relevant to the request are extracted through similarity computation, matching processing and inferences in order to meet the need of the user before a set of retrieved documents can be presented [21, 58]. The set of retrieved documents (answers) may sometimes be relevant or irrelevant to the user's needs, in which case the query may need to be reformulated. Every time a new set of query words are applied, with the same semantics, a new crop of documents (answers) are retrieved and presented.

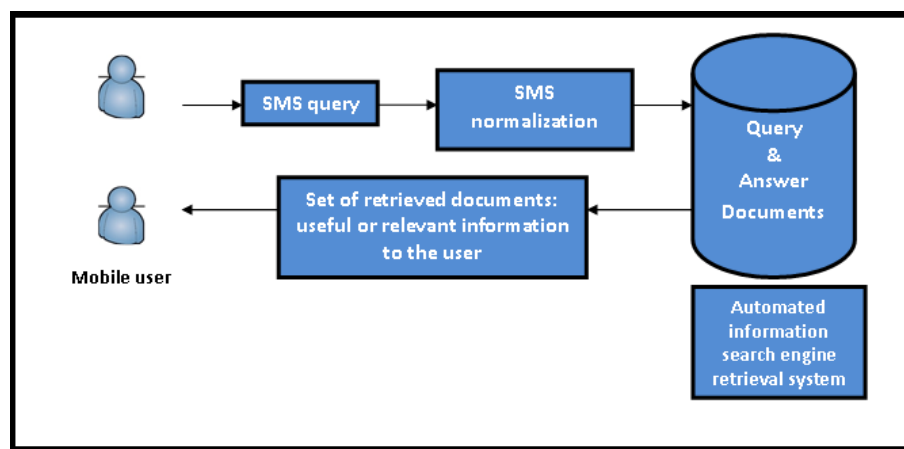


FIGURE 2.2: Automated FAQ information retrieval system

With regard to mobile information retrieval, the length of time spent by a mobile user at a particular search-service may be very short because the answers retrieved are satisfactory/not satisfactory or available/not available. Mobile searchers vary in persistence. The vast majority of mobile searchers approach queries with a specific topic in mind and their search often does not lead to exploration, unlike desktop searches [234].

Information retrieval is usually keyword dependent but the challenge arises when the search engine not only has to know how to extract keywords and determine the weight of each, but also has to determine the distribution of words and compare them with the document and the corpus distribution [227].

## 2.11 SMS-based FAQ information retrieval mechanism: a review

The approach of Burke et al. [41] is to produce a natural-language-processing question-answering system that uses FAQ files as its knowledge base. The technique is based on

four assumptions used to convert the *FAQFINDER* system: (1) organizing the FAQ file in QA format; (2) setting the information locality within the QA pair; (3) determining the question's relevance, within the QA, to find the match; and (4) possessing a general knowledge of the languages for question matching. The user's query terms are matched with the FAQ files. The searching process is conducted on a small set of FAQ files that are likely to have the best match to the user's query. Mogadala et al. [166] use a *language modelling* (LM) approach to match noisy SMS text with the right FAQ. The team developed a dictionary-based approach to clean the SMS text. The cleaned SMS text is then matched with the FAQ using an LM approach (for retrieval purpose, after SMS normalization) before the corresponding response to the query is released. The experiments by Mogadala et al. [166] were conducted by combining SMS datasets of English, Hindi and Malayalam languages with their corresponding FAQs in different combinations for the mono-lingual task, and FAQs in Hindi and the English language for the cross-lingual task. In both sets of experiments, the percentage of the languages was continuously varied in order to retrieve information from their FAQ databases using English SMS queries. The FAQs were divided into 3 different collections: (1) the questions only; (2) the answers only; and (3) combinations of questions and answers on the three languages. The results show that developed LM questions outperformed both answers, and combinations of questions and answers, for matching SMS queries. The LM model does not give consideration to synonyms; it is word-dependent. This means that any other answers that could be chosen in the FAQ answer dataset may not be considered.

An *n*-gram *count-based algorithm* developed by Jain [115] takes account of various *n*-grams in order to calculate the score of questions from the corpus. This is similar to the approach used to develop *SMSql*, (Section 3.10C). The score of different FAQ questions from the candidate sets is calculated. The maximum score among the set is then returned with its corresponding answer in the FAQ database. Two factors are considered that lead to an enhancement in evaluating the FAQ score in the candidate set. They are the proximity of the SMS query and FAQ tokens, and a comparison of the question sentence length of the matched tokens from the SMS query to the FAQ questions under consideration [115, 121]. When the algorithm was evaluated on many real-life FAQ datasets from different domains, the results show significant improvement in terms of the accuracy compared to Kothari [132].

Hogan et al. [100] identified SMS-based FAQ retrieval systems as having three steps—(1) SMS normalization, (2) retrieval of ranked results and (3) identifying out-of-domain query results. In order to normalize the SMS FAQ queries, a set of transformation rules were created and the corpora were manually annotated. The rules were never

published. The tokens were aligned with the original text messages to give a one-to-one correspondence between the original and corrected tokens. The documents and SMS questions underwent the same pre-processing. In the research of Hogan et al. [100], each SMS token is examined (if it remains unchanged) and the corrected token is substituted. A set of candidate lists are generated. The best candidate in the context is selected as the correction. The best candidate was selected using 3 methods: (1) manually annotated data was used as a correction rule, to get the best translation for the SMS tokens. The frequency of use of the correction rules becomes a criterion for calculating the normalized weights of the replacement of SMS token in the corpus. (2) Candidate corrections were created by consonant skeletons. The mapping between the consonant skeletons and the words produces additional correction candidates for the query words. (3) Candidates are generated when all words in the corpus are compared with the prefix of the question words, to confirm if there is truncation.

The three methods produce replacement candidate lists, which are merged by adding their weightings from their term frequency. The token scores are calculated using the maximum product of that weight and the  $n$ -gram score of the corrected token. The disadvantage of the model is that it uses a manual annotation of the dataset, which may be cumbersome for large corpora. The experiment was performed on monolingual English SMS datasets with different retrieval engines (*Solr*, *Lucene*, and *a combination of the two search engines*) and approaches. The best result from the candidate list is retrieved by ranking the weighted scores of a list of question-answer pairs. The evaluation of the results involved comparing out-of-domain results when tested on the two search engines. The SMS normalization approach is token based. All the tokens are processed.

*SMSFind* is another SMS-based information retrieval model proposed by Chen et al. [50]. It is designed to deliver the final search response to a normalized SMS query. It uses a conventional search engine in its back end to provide an appropriate answer for the SMS request. *SMSFind* uses translated SMS queries. Typically, the arrangement contains an SMS term or a collection of consecutive terms in a query that provides a *hint* as to what the user is looking for. The *hint*, provided by the user or automatically generated from the document, is used to address the information extraction problem. *SMSFind* uses this *hint* to address the problem as follows: given the top search responses to a query from a search engine, *SMSFind* extracts snippets of text from within the neighbourhood of the *hint* in each response page. *SMSFind* scores snippets and ranks them across a variety of metrics. The *hint* extracted is used to determine the answer to the request. It is scored based on a *top-n* list for each page. The highest score is released as an answer to the request [50]. The use of *hints* in the algorithm is considered a supervised learning approach [3, 59] and it is expensive to generate and store. The research never considers the contextual information of the searches.

Kothari et al. [132] designed an automatic FAQ-based question answering system. The method involves promoting SMS query similarity to FAQ-questions. This is done through a combinatorial search approach. The search space consists of combinations of all possible dictionary variations of tokens in the noisy query. The combinatorial search system models an SMS query as a syntactic tree matching so as to improve the ranking scheme after candidate words have been identified. Initial processing of noise removal was introduced so as to improve the information retrieval efficiency. The model involves the use of a dictionary, and maps the SMS query to the questions in the corpus. The noise removal step is, however, computationally expensive [134]. The system developed by Kothari et al. [132] does not involve training SMS data on text normalization. It has the advantage of handling semantic variations in question formulation but the method fails to discuss the choice of homophonic words in the context of automatic speech recognition. Kothari et al. [132] depend on a scoring function for the choice of selecting FAQ questions. In cases where there is a tie over the score function, it will be difficult to rank the question, and other factors, such as the proximity measurement of the SMS query and FAQ token, proposed by Jain [115] and Joshi [121], may be considered.

Recent work by Darnes Vilarino et al. [223] is based on the probability model of an SMS-based FAQ retrieval system. Monolingual, cross lingual and multilingual approaches were implemented on the dataset from three sources, English, Hindi and Malayalam languages. SMS normalization was carried out initially by substituting each query term with the closest translation offered by a bilingual statistical dictionary. The dictionary was used to calculate the most frequent calculated term from a training corpus of the SMS query term that is associated with FAQ terms. The *Gizza++* tool is used to calculate the most frequent term through the use of IBM-4 model, by using a training corpus composed of a set of aligned phrases (i.e. one SMS to its corresponding FAQ). IBM 4 model works on relative reordering of previously translated words (cepts) [45, 128, 129]. The similarity among the SMS terms and each of the FAQ questions was calculated using the Jaccard similarity coefficient. Jaccard coefficient measures similarity value,  $N$ , between SMS and FAQ sets by calculating the size of the intersection divided by the size of the union of the sample sets [114, 135]. All values of FAQs above  $N$ , is returned as the answer set of FAQ. There are two shortcomings on this method, (1) the contextual information of the SMS query and FAQs are better measured by considering a phase-based approach than being word, and (2) the approach did not take into account the frequency of the terms among the documents that are compared.

*SMSFR* is another recent SMS-based searching technique developed by Pakray et al. [178]. It has a multi-lingual text corpus (English, Hindi and Malayalam) acquired from different FAQ datasets. A Bing spellchecker (open source and of high quality) was used as the dictionary for SMS normalization. The retrieval technique involves the *unigram*

matching, *bigram* matching and *1-skip bigram* matching modules created for the SMS and FAQ datasets. The research has the goal of acquiring the best FAQ for the SMS query. In the monolingual technique, a rule-based system for ranking the candidate FAQ terms is applied. The system also has four modules (pre-processing, unigram, bigram, and 1-skips bigram matching modules) for the normalization processes. (1) The pre-processing involves SMS translation. (2) For the *unigram* matching, the Bing spellchecker module processes the SMS and FAQ datasets to discover a match for a new word. The similarity in the word of the SMS and FAQ confirms the search. If there is no match, *WordNet 3.0* is searched for hyponyms, synonyms etc., of the FAQ terms for the comparison. This is an extra cost to the FAQ dataset, as it is assumed to be error free. The *WordNet* is a lexical database for the English language that groups English words into sets of synonyms called synsets [82]. (3) The *bigram* matching compares the match between the two statements by considering the bigram occurrences of their words. The two consecutive words in the two datasets are compared. If there is a match, the next consecutive bigram is searched; otherwise the WordNet is searched for the *bigram* sequences of the SMS and FAQ. (4) *1-skip and inverse bigram* matching consider a sequence bigram with one gap of two words. The similarity of the two words (SMS and FAQ) in the list of SMS (S') that is found on the inverse order of FAQs list (F') is considered. A set of semantic rules is applied to confirm the match when the pairs are not rejected. However, the sets of rules applied to confirm the store were not stated.

The output of the top five scores is used for the single SMS query, considering all the processes. The use of Bing Speller is restricted only to those words found in the dictionary. If the term is not in the database the right answers are not provided. This approach is economical because Bing speller is freely available online.

Healthcare FAQ information retrieval systems using SMS in the form of a Question and Answer (Q&A) System were proposed by Anderson et al. [13] and Masizana-Katongo et al. [153]. SMS users submit queries to the portal through a mobile phone interface. A *parsing technique* was proposed as a retrieval mechanism to match the relevant answers [12]. The parser extracts and processes keywords from the SMS input text. This leads to matching the SMS keywords to a relevance FAQ dataset. 20 HIV/AIDS questions written in English were written in SMS format. Frequently occurring SMS terms were extracted from each question. Each question could be evaluated on its merits from the combination of the frequently occurring phrases and/ or words within the phrases. This may be achieved by statistical analysis. The SMS input format in the form of grammar is then parsed through the automatic parser generator or compiler. A parser generator reads a grammar specification and converts it to a program that recognizes grammar matches. A method is generated (in the code) that corresponds to each production in the grammar. The technique involves the translation of the grammar provided in

*Backus-Naur Form* (BNF) format into pre-processed parsed tree building blocks that can be easily implemented in Java code. The system is evaluated using the metrics of recall, precision and rejection. Their procedure did not consider ranking of the SMS query in presenting the answer.

## 2.12 Keyword extraction: a review

Keywords can be defined as the index terms that contain the most important information for the user. Their purpose is to identify a small set of words from a document which will represent the meaning of the document. Keywords can be stand-alone terms or appear as part of a group of terms with adjacent keywords [224]. They can also be defined as the smallest word unit which expresses the meaning of the entire document, referred to in automatic indexing, text summarization, information retrieval, topic detection and tracking, report generation, web searches, question and answering, etc. [23]. In text summarization, keywords can be used as a form of semantic metadata [23, 67], beyond content search, index and rank. Intuitively, the word that appears often in a document but not very often in the corpus is more likely to be a keyword and, conversely, keywords that occur in many documents within the corpus are not likely to be selected as statistically discriminating keyword terms [201]. It is essential that keywords cover the important areas of a document.

There are automatic keyword extraction or summarization methods that provide the actual contents of a given document, in the form of key phrases or keywords [108, 117, 125, 201]. Early approaches to automatically extract keywords focused on evaluating the corpus-oriented statistics of individual words [201]. Statistical methods were adopted in carrying out keyword extraction by Salton et al. [203]. They discovered a positive result from selecting an index vocabulary statistically, across the corpus. The statistical method involves statistical information such as word frequency, term frequency (*tf*) and inverse document frequency (*idf*), as well as word co-occurrence, as a means of identifying keywords in the document or text against the reference corpus [201]. According to Zhang et al. [242] the *n*-gram concept is used automatically to index the document. With a search query of *n* keywords, the maximum size of the keyword group is equivalent to the number of query keywords in the document. This means that a query with *n* keywords will contain a maximum of *n* keyword groups. The idea of grouping the keywords in a document is motivated by the assumption that terms found in keyword groups should be more significant in the document and be given more weight than stand-alone terms [224].

Previous work on document-oriented methods of keyword extraction has combined natural language processing approaches to identify part-of-speech (POS) tags with supervised learning, machine-learning algorithms, or statistical methods. The research work of Hulth [108] compares the effectiveness of three term-selection approaches: noun-phrase chunks,  $n$ -grams, and POS tags. Four discriminative features (term frequency, collection frequency, relative position of the first occurrence and POS tags assigned to the term) were used as inputs for automatic keyword extraction using a supervised machine-learning algorithm [108]. Masizana-Katongo et al. [153] implemented an SMS parser under a FAQ system of HIV/AIDS queries using an example-based parsing solution, and keyword extraction was performed based on the available data set [153].

Mihalcea and Tarau [163] worked on a graph-based ranking model for keyword extraction from natural language texts. The system describes how a syntactic filter is applied to identify the POS tags used in selecting words to evaluate keywords. Word-occurrence graphs accommodate a selected word within a fixed-size sliding window which is then ranked in accordance with a graph-based algorithm (TextRank). Highly ranked words are placed on top, based on their association in the graph and are also selected as keywords. The performance of this experiment is at its best when only nouns and adjectives are selected as potential keywords.

Matsuo and Ishizuka [154] apply a Chi-squared measure to calculate how selective words and phrases co-occur, within the same sentences, as a particular subset of frequent terms in the document text. The Chi-squared measure is applied to determine the bias of word co-occurrences in the document text, which is then used to rank words and phrases as keywords of the document. The research concluded that the degree of bias is unreliable when term frequency is small, and that the method operates effectively especially when the documents are large.

Rapid Automatic Keyword Expansion (RAKE) [201] is a developed method that operates effectively on individual documents to enable applications to work on dynamic document collections. The RAKE technique is an unsupervised, domain dependent, and language independent method for extracting keywords from individual documents. The RAKE algorithm ignores the use of grammar specifications and is based on the observation that keywords frequently contain multiple words but rarely contain standard punctuation or stop words, such as the function words *and*, *the*, and *of*, or any other words with minimal lexical meaning [201]. The algorithm uses stop words, word delimiters, and phrase delimiters to partition the document text into candidate keywords and content words (uninformative words) as they occur in the text.

A further approach to identify a set of likely cognates in sentence form is to align the segment based on words and their pairings [131, 234]. There is string matching based on

one-on-one word alignments of the bi-text [159]. There may be a need to compare each word pair by computing their similarity values. Word pairs that have high values indicate great similarity and these are ranked higher. The set of likely cognates is obtained by selecting all pairs with similarity values above a certain threshold value. This can be compared with the frequency of words chosen in the wordlist when the SMS is gathered. The threshold value determines the extent of similarity between a pair of documents [149]. The keyword phrases and idioms are ultimately used to determine the question from the database and used in the similarity comparison with the FAQ data set.

In this research the keyword extraction technique is used in the FAQ system to identify stop words (or stop lists), phrases, and word delimiters. Candidate keywords are isolated by removing stop words from the FAQ text. What this means is that the word or phrase delimiters will now represent the keyword or key phrase, which are sequences of content words as they occur in the FAQ text. It is on this basis that the scoring function will be calculated. The array of stop words (or stop lists), phrases, and words is split into sequences of contiguous words at phrase delimiters and stop word positions. Every word that is represented in the FAQ files is either a stop word or a candidate keyword, and these categories of words are selected and stored in preposition/punctuation tables (see *Figure 3.10*) in the *MySQL* relational database. In practice stop lists are often based on common function words and are hand-tuned for particular applications, domains, or specific languages [201].

This research focuses on methods of keyword extraction that operate on individual documents (i.e. FAQ-query), rather than on a corpus because it will extract keywords exactly from the FAQ query sentences, regardless of the state of the corpus document. Part of the design of the research is to augment more questions, out-of-domain, from other sectors outside the HIV/AIDS domain, to verify and evaluate the research efficiency. The best-matching words can then be found by processing just those lists that are associated with the  $n$ -grams comprising the query word for which the variants are required [199].

### **2.13 SMS security**

SMS is based on a store and forward service where messages received from the mobile user are stored in a central server message centre, and forwarded from there to the mobile recipient. Storage is very important to ensure that the message will eventually be sent if at the time it is sent, the recipient's phone is switched off or out of coverage. Security issues become important as it has been noted that advanced technology applications like m-commerce or banking [220] and electronic health records [74] depend on the system of sending and receiving text messages to authenticate the user. SMS security is vital



during transmission or when data is retained in the database through the encryption techniques of secure shell (SSH) protocols.

Encryption prevents a third person from understanding SMS information should it be intercepted. A patient's records, in SMS form, can be digitally scrambled in such a way that only authorised people who possess the key to the encryption code can decrypt the data. Encryption can be *symmetric* (Figure 2.3) or *asymmetric* (Figure 2.4)[141]. *Symmetric encryption* systems provide a two-way channel for their users: sender and recipient share a secret key and they can both encrypt information to send to the other, as well decrypt information in the reverse direction.

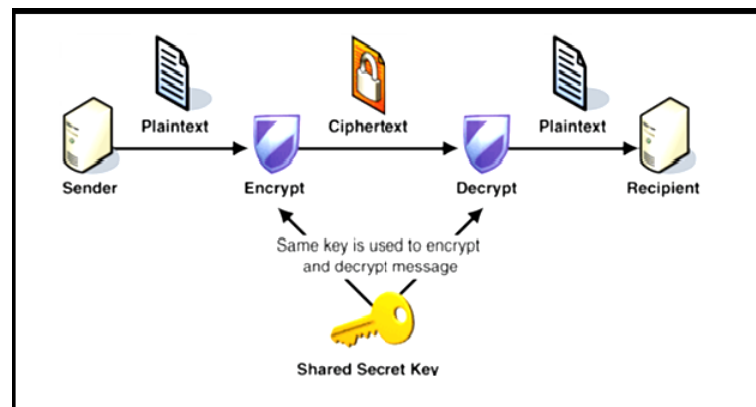


FIGURE 2.3: Symmetric encryption diagram [101]

Authentication is genuine as long as the SMS message received was not fabricated by someone other than the declared sender. The only challenge to this scheme is the manner in which the secret key is sent to the recipient. Key distribution can be difficult, especially if there is a need for another user. In general,  $n$  users who want to communicate in pairs will need  $n(n-1)/2$  keys. What this means is that the number of keys needed increases at a rate proportional to the square of the number of users [1, 5].

Conversely, *asymmetric encryption* systems involve each user having two keys that are unique to them, a public key and a private key. A trusted third party is used to facilitate secure interactions between the two parties. The user may send the public key freely because each key is used for only half of the process. That is, one key decrypts the encryption made by the other, and vice versa. Only the corresponding private key (presuming it is kept private) can decrypt what has been encrypted with the public key [46].

Encrypting patient information in the web server before transmission can help to protect the information, although anyone who obtains the key can access the data. The key to successful encryption is to limit the number of persons who have the key to encrypt and decrypt the data, and to determine the appropriate length of the key [46, 187]. Table 2.5

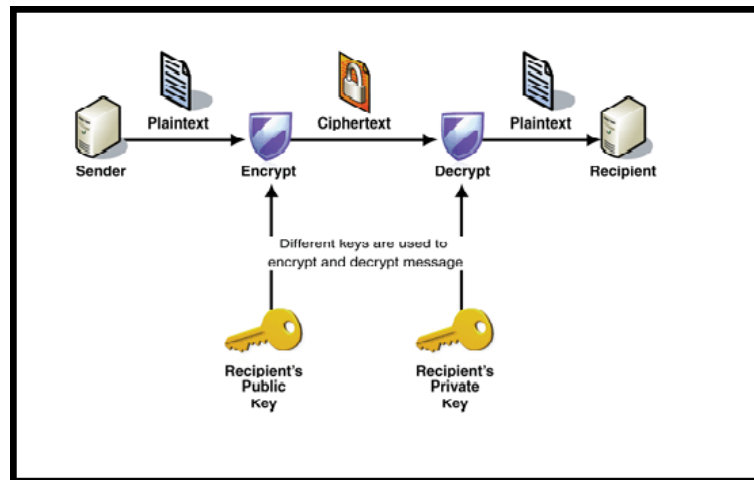


FIGURE 2.4: Asymmetric encryption diagram [101]

shows a comparison between symmetric and asymmetric encryption systems in terms of their transformational speed, diffusion of information, propagation of error and insertion of symbols.

### 2.13.1 Secure Shell (SSH) Protocol

Secure shell (SSH) is a secure application which enables a user to log into another computer over a network and execute commands on the remote machine [148]. SSH provides strong authentication and secure communications over unsecured channels [11, 239]. It is a protocol that permits a client to contact a server and run an application on it securely. When a session is established, the client and the server are authenticated and data runs through a secure channel to ensure its privacy and integrity [44]. SSH uses public-key cryptography to authenticate the remote server, to establish the authentication of the user and encrypt the communications over un/secured channels [10, 44]. The SSH server presents a public key, and the SSH client or mobile device uses standard cryptography to establish a protected channel, with the server knowing the private key, (*asymmetric encryption*). SSH can permit user authentication via a key pair (i.e. client/server)[10]. The purpose of the key exchange is dual. First it attempts to authenticate the server to the client and, secondly, it establishes a shared key which is used as a session key to encrypt all the data being transferred between the two machines. The session key encrypts the payload and a hash generated for integrity checking of the payload using the private key of the server. The client verifies the server's public key, verifies the server password received, and then continues with user authentication.

The cryptography algorithm, RSA, is an algorithm used for public key cryptography and is designed to secure communication between the FAQ information and the mobile user.

TABLE 2.5: Comparison between symmetric encryption systems (stream algorithms) and asymmetric encryption systems (block algorithms)

Encryption type	Advantages	Disadvantages
<b>Symmetric (stream encryption algorithm)</b>	<i>Transmission speed:</i> is high because the symbol is encrypted without regard for any other plain text symbols – each symbol is encrypted as soon as it is read. Encryption algorithm is the factor that determines the time to encrypt a symbol, but not the time it takes to receive the plain text.	<i>Diffusion is low:</i> each symbol is enciphered separately. The symbols information is contained in only one symbol of the cipher text.
	<i>Low error propagation:</i> an error in the encryption process affects only that character, because each symbol is separately encoded.	<i>Malicious insertion and modification:</i> the symbols are separately enciphered, which allows the code to be compared with a similar or previous message and allows a counterfeit or new message that may look genuine to be transmitted in place of the original.
<b>Asymmetric (block encryption algorithm)</b>	<i>High diffusion:</i> information from the plain text is diffused into several cipher text symbols. One cipher text block may depend on several plain text letters.	<i>Slow encryption:</i> all plain text symbols will have to be received before the encryption process commences.
	<i>Difficulty in symbol insertion:</i> enciphering is done based on blocks of symbols therefore it is rather difficult to insert a single symbol into one block, otherwise the length of the block will be incorrect.	<i>High error propagation:</i> if an error occurs in the block, it will spread across the block and affect the block transformation.

The RSA algorithm, named after the inventors Rivest, Shamir and Adleman [130, 192], is used for securing, among others, the email program called Mail Safe [98, 216], and is thus used for SMS security. This *asymmetric* algorithm consists of (1) key generation (the process of generating the public and private RSA keys), and (2) RSA function evaluation processes (this technique is used in transforming a plaintext message into ciphertext, or vice versa). Key generation aims to generate public and private RSA keys in the following steps: (1) generation of a large prime number, (2) creation of a modulus from the large number, (3) the totient of the large prime number is calculated, (4) the public key is generated, and (5) the private key is generated.

The RSA-ENCRYPTION algorithm requires two distinct large prime numbers,  $p$  and  $q$ , from which the product  $n \leftarrow p \cdot q$  is formed. Another prime number in the range  $[2 \cdot \phi(n) - 1]$ , and a co-prime factor  $e$ , is found which is relatively prime to  $\phi(n)$  and from

this the private key  $d$  is calculated such that  $d \cdot e \bmod \phi(n) \equiv 1$ . The key  $e$  is used to calculate the cipher in Line 6 by repeated exponentiation.

---

**Algorithm 1** RSA-encryption( $m$ )
 

---

- 1:  $n \leftarrow p \cdot q$
  - 2:  $\phi(n) \leftarrow (p - 1) \cdot (q - 1)$ , Eulers's totient for  $n$ .
  - 3: Find a random number  $e$  such that  $1 < e < \phi(n)$ , which is relatively prime to  $\phi(n)$ .
  - 4: Compute a number  $d$ , the private key, such that  $d \cdot e \bmod \phi(n) \equiv 1$ .
  - 5: The length of  $m$  must satisfy  $|m| < |n|$ .
  - 6: **return**  $m^e \bmod n$ .
- 

The RSA-DECRYPTION algorithm uses the partner of the public key  $e$ , i.e., the private key  $d$ , in Line 6 to decipher the enciphered message  $c$ .

---

**Algorithm 2** RSA-decryption( $c$ )
 

---

- 1:  $n \leftarrow p \cdot q$
  - 2:  $\phi(n) \leftarrow (p - 1) \cdot (q - 1)$ , Eulers's totient for  $n$ .
  - 3: Find a random number  $e$  such that  $1 < e < \phi(n)$ , which is relatively prime to  $\phi(n)$ .
  - 4: Compute a number  $d$ , the private key, such that  $d \cdot e \bmod \phi(n) \equiv 1$
  - 5: The length of  $m$  must satisfy  $|m| < |n|$ .
  - 6: **return**  $c^d \bmod n$ .
- 

---

A simple example to illustrate RSA-ENCRYPTION/DECRYPTION for SMS:

---

Choose  $p = 3$  and  $q = 11$ , and  $n = p \cdot q = 3 \cdot 11 = 33$ .

Compute  $\phi(n) = (p - 1) \cdot (q - 1) = 2 \cdot 10 = 20$ .

Choose  $e$  such that  $1 < e < \phi(n)$  and  $e$  and  $n$  are co-prime. Take  $e = 7$ .

Compute a value for  $d$ , the private key, such that  $d \cdot e \bmod \phi(n) = 1$ .

A possible solution is  $d = 3$ , since  $3 \times 7 \bmod 20 = 1$ .

Public key is  $(e, n) \Rightarrow (7, 33)$ .

Private key is  $(d, n) \Rightarrow (3, 33)$ .

$c = \text{RSA-ENCRYPTION}(m)$ , with  $m = 2$  yields  $c = 2^7 \bmod 33 = 29$  and

$m = \text{RSA-DECRYPTION}(c)$ , where  $c = 29$  gives  $m = 29^3 \bmod 33 = 2$ .

---

It should be noted that SMS messages are converted to their ASCII codes, then to strings. The strings are converted to a bit array for the cryptography to be accomplished. The bit array is later converted to a large number [31] that is suitable for SMS security. It is important to verify the unknown public keys, i.e. to associate the public keys with identities, before accepting them as valid. The cryptography algorithm verifies whether the same person offering the public key also owns the matching private key. Accepting an attacker's public key makes the system vulnerable to attack.

## **2.14 Chapter summary**

This chapter provides detailed reviews of literature that deals with the existing SMS classification and normalization algorithms, text entry errors, similarity measurements, SMS-based FAQ information-retrieval mechanisms, keyword extraction and SMS security. In Chapter 3, the research approach is described, including algorithms for SMS normalization and information access using SMS.

## Chapter 3

# Research Design and Methodology

### 3.1 Introduction

The relevant research literature was presented in the previous chapter. The focus of this chapter includes the two research objective areas, (1) SMS normalization and (2) the use of the normalized SMS for information access in a repository of FAQ. The research design, approach and methodology applied in the study are discussed in Section 3.2. The methodology presents the approach the researcher pursued in order to achieve the research objectives. The route can be arrived at by many different means. The important point is to see how the route is established between the starting and finishing points [64].

The collection of data used in the two research objectives is described in Section 3.3. As part of the first objective, Section 3.4 describes the data structure and methodology used in SMS normalization. The developed algorithm—the SCORE algorithm—is described in Section 3.5. The research experimentation for the developed SMS normalization algorithm is explained in Section 3.6. The second objective starts in Section 3.7 with the description of the data structure and methodology used in information access using SMS in a FAQ system. In Section 3.8, SMS-based FAQ analytical methods are further described and enumerated. This is followed in Section 3.9 by a description of an experimental methodology on SMS-based FAQ information access. The algorithms used for information retrieval experiments are discussed in Section 3.10. The statistical analysis for the two research objectives is carried out in Section 3.11. Section 3.12 summarises the chapter.

## 3.2 Research design and approach

Crotty [64] defines the research process (see *Figure 3.1*) in terms of four elements—epistemology, theoretical perspective, methodology and methods.

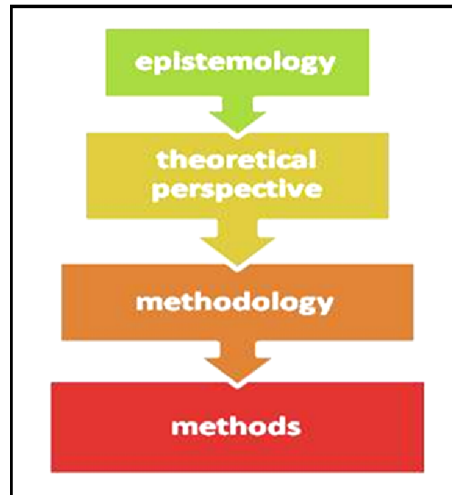


FIGURE 3.1: Four elements of the research process [64]

(1) The epistemological stance. Epistemologists acknowledge four main channels of knowledge. These are intuitive knowledge, authoritative knowledge, logical knowledge and empirical knowledge. Epistemology (the theory of knowledge) is described as the investigation into the grounds and nature of knowledge itself [71, 96]. The study of epistemology focuses on the means of acquiring knowledge and how one can differentiate between truth and falsehood. The research undertaken here adopts the epistemological stance of objectivism. Objectivism rejects the notion that a group of people or individuals establish their own reality without verification [72]. Accepting objectivism as the dominant epistemology, the assumption here is that user experience and the results obtained from the experimental work can be evaluated, verified and quantified [64].

(2) The theoretical perspective. A theoretical perspective is a non-explanatory general framework that is meant to define a point of view within a discipline. The framework may include basic assumptions that draw attention to aspects of a phenomenon which generate questions about it [66]. Positivism is a theoretical perspective that allows for systematic, practical and empirical evaluation of a natural occurrence that is based on scientific theory and hypotheses about interactions among such occurrences [156]. In the case of this research, positivism is an appropriate theoretical perspective mainly because the research requires a scientific or quantitative appraisal [30, 207] of the system and the algorithms developed.

(3) The methodology. Methodology is a strategy or action plan to choose appropriate research methods and link them to the desired outcomes. In this study algorithmic and

experimental methodologies were used to manage the research process. An algorithm is a step-by-step problem-solving procedure, especially an established, recursive computational procedure, for solving a problem in a finite number of steps [113]. Experimental research involves trying something and watching the resulting effects. The experiment can be conducted in a controlled condition (such as a laboratory) or in the field [176]. In order to guide the automation of SMS normalization and mobile information access, *PHP* and *MySQL* software were used in a 2-tier architecture i.e. client/server side architecture (see *Figure 3.2*). Client/server architecture is a design in which the user interface runs on the client and the database is stored on the server [222].

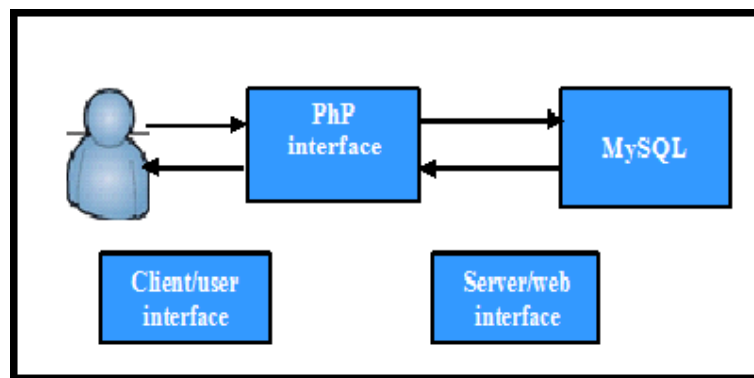


FIGURE 3.2: Application database connection

The actual application logic can run on either the client or the server [144]. The system runs on a Web platform. The database provides information to a *PHP* interface that retrieves information from the web server.

The methodology adopted in the first experiment, SMS normalization, is described in *Figure 3.3*. The SMS text is normalized in a five-stage modularized system. The five-stage system is a web server architecture whereby the SMS text is searched, compared and replaced with a data set that is available in the database of the modular system using the proposed (SMS) SCORE algorithm. The SCORE algorithm was adopted to achieve SMS normalization. SCORE is a character-based algorithm that processes SMS text one letter at a time, in sequence. Each letter of the SMS is *searched* in the dictionary database. The dictionary consists of over 40,000 English words with medical terminology. A sub sequence of the SMS text message is *compared* with a sequence of dictionary words following the order of the SMS. There is a *replacement* of a letter in the SMS input. In case there is more than one dictionary word that matches the SMS input, a *word-error-rate* approach is included.

The *word-error-rate* approach consists of three operations, *replace*, *insert* and *delete*, that can be performed on the candidate words. The candidate words have an equal chance of being a good substitute for the SMS word. If there is a tie, a *rule-based*



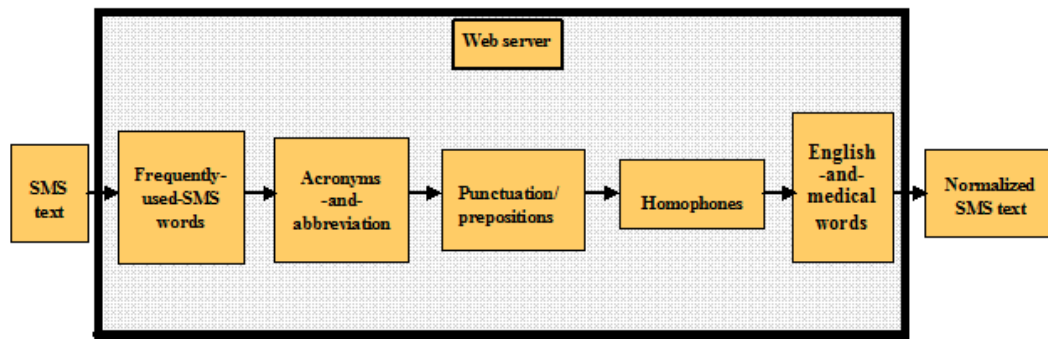


FIGURE 3.3: SMS normalization architecture showing various modules

condition is introduced. A rule is a decision list or a set of conditions set aside to make deductions or choices, in form of *IF...THEN* statements [180]. Rules are application-dependent and aimed at achieving the following attributes: completeness, consistency and continuity [39, 88, 165]. In this case, an *order of vowel precedence* is introduced so that any candidate word containing the vowel *e* will be selected as the replacement for the SMS input, follow by those containing (in this order) *a*, *i*, *o*, and *u*. Other hypotheses are also considered in building the algorithm to assist in the search and retrieval of the English word corresponding to the SMS word typed in by the user. The algorithm is described in Section 3.5.

The methodology adopted in the second experiment (mobile information access using SMS) is described in *Figure 3.4*, an information retrieval map of a set of SMS query terms which specify user information requirements. These are mapped to a set of objects referred to as answers (FAQ), in a given data collection. The SMS query is presented in the form of a sentence. The SMS query normalization process translates the SMS token into a *clean* English form to become normalized SMS text. This is the essence of the proposed SMS normalization (SCORE) algorithm. Stop words are extracted from the normalized SMS text sentence, leaving behind the keywords (Section 2.12). Vector spaces are created between the SMS text and the FAQ corpus. A set of answers is retrieved from an indexed FAQ-answer corpus. There is a mapping or string matching between the vector spaces of SMS query terms and the FAQ corpus. The sets of retrieved documents are ranked in order of similarity, matching the SMS query. They are presented as the answer. If the answer is not satisfactory, the user can then reformulate the query.

The architecture and methodology involved in the entire experiments is shown in *Figure 3.5*. It is a web-based design in which SMS is sent from the user, or the client side, and the text undergoes a normalization process to become a refined SMS query. Like in *Figure 3.4*, there is a matching process between the refined SMS query sentence and the FAQ repository which produces the answer to the request. The result is displayed

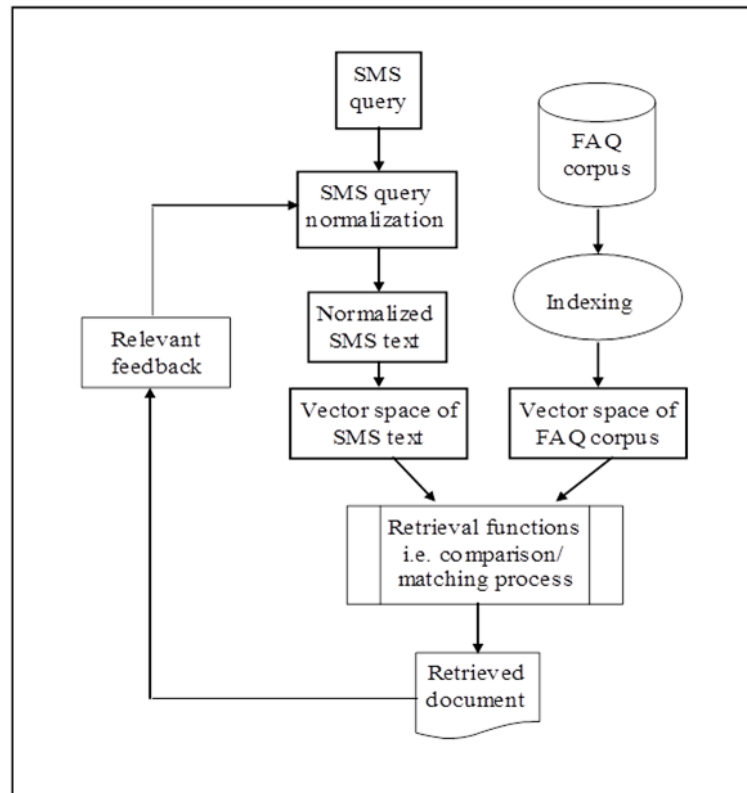


FIGURE 3.4: Information retrieval process

on the client side. If the user is not satisfied or wants further enquiry the process is repeated.

SMS query-term extraction is viewed as a stage where stop and unwanted words are removed, leaving behind the keywords. A keyword is used to determine or calculate the similarity between the user's question and the FAQ entry in the database [169, 170]. Huang et al. [107] describe the keyword-order relationship as an important factor, especially when keywords stand as adjacent terms. It is possible to consider the order in measuring a term's weight. Assigning more weight to adjacent terms in a query sentence results in the FAQ document vector being moved closer to the SMS query vector. This will increase the relevancy between the two vectors, and eventually result in documents with better relevance being retrieved. Different sentence-matching techniques, (word-based, semantic and a combination of the two methods) are used in similarity matching. Word-based matching techniques take the similarity of surface features of the two sentences, whereas semantic techniques use the lexical relationship between terms of the two sentences [122].

(4) The research methods adopted involve *content analysis*, a *pilot study*, *sampling*, *experimentation*, and *statistical analysis*.

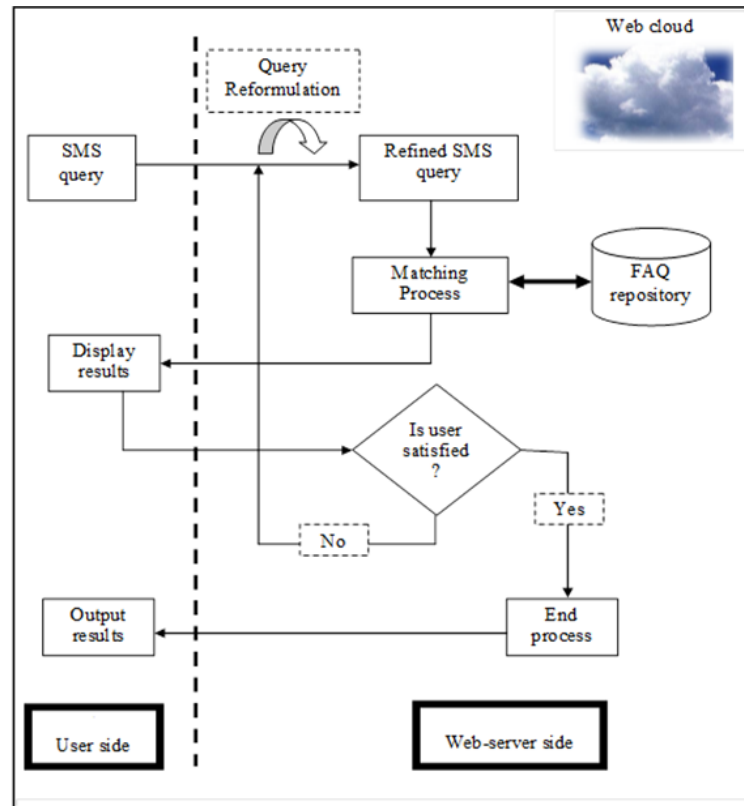


FIGURE 3.5: Web-based SMS normalization and information retrieval flow diagram

*Content analysis* refers to the act of reviewing the existing documentation of related research areas so as to retrieve and extract items of information that are useful to the current research and project. Hence it should be regarded as an important project requirement [176]. Articles or documents are reviewed so as to draw connections within the research area currently being studied [47]. Content analysis was carried out to achieve two objectives: (1) to determine existing SMS normalization techniques in order to develop a robust SMS normalization, and (2) to use normalized SMS text for information access in an FAQ system.

*Pilot study*: a pilot study is a trial run to test the research instrument with a subsample having characteristics similar to those identifiable in the main sample [78, 92]. According to Felicity Smith [210], a pilot study is done for two purposes: first, to ensure that it is workable in practice settings, in terms of study procedures and data collection, which must be acceptable to participants and others on whom the conduct of the study may impact; and secondly, to check that the study procedures gather reliable and valid data effectively and efficiently. Conducting a pilot study before the main evaluation allows potential problems to be identified and corrected. Since SMS is usually used among the youth, the experimental procedure was conducted among this population. In the early stage of this research, raw samples of SMS from 40 first-year students in the Department

of Computer Science, at the University of the Western Cape, were collected. This was done by transferring data from their cell phones to the experimental cell phone. The outcome of the *pilot study* was the generation of an SMS corpus used for initial testing of the algorithms.

In the case of the second experiment, information access using SMS, information was collected from the university community using sampled questions (see *Appendix A*) that centred on health matters related HIV/AIDS. The term “sampled question” is usually reserved for lists of questions to be used in the experimental evaluation. The set of FAQ was made short and simple to avoid ambiguity. This exercise was executed in two ways: (1) sampled questions were personally administered in the student community with instructions that answers to the set of questions were not required but the exact way the questions would be written if they were to be used with cell phones (i.e. their SMS forms); (2) online survey tools available on the internet ([www.surveymonkey.com](http://www.surveymonkey.com)) were employed for the same purpose. An account was opened on the web site and the same questionnaires were sent to student email addresses. *SurveyMonkey* was used because it has a number of useful tools, the data sets are made available in form of electronic text and the basic service is free. The dataset was collected over a period of six months.

*Sampling:* in testing the experimental results, SMS terms were randomly selected from the SMS corpus to investigate the robustness of the algorithms. FAQ written in SMS forms were submitted to the database server. Unbiased samples were selected from the population of the dataset.

*Experimental method:* this involves manipulating one variable to determine if changes in one variable cause changes in another variable [218]. The method relies on controlled procedures, random assignment and the manipulation of variables to test the research questions. The formal English word/phrase/sentence is taken as the controlled dataset while the SMS serves the purpose of random verification in the algorithmic test which addresses the two research objectives. The algorithms were worked on separately in order to improve translation efficiency and information retrieval in the FAQ system. This method was chosen to meet the research objectives of the study.

*Statistical analysis:* a quantitative approach was followed. Quantitative research is a formal, objective, systematic process to describe and test relationships and examine cause and effect interactions among variables [42, 53]. Sample questions were used for descriptive, explanatory and exploratory research. A descriptive survey design was used for the experiments. Descriptive statistics refer to statistics that are calculated from the characteristics of the population, sample or other group, and serve to describe the group [179]. The following metrics were used to confirm the level of significance of the

experiments: *paired-samples t-tests*, *tests of normality* and *correlations* for SMS normalization. Two methods of SMS normalization (BLEU and SCORE) were compared. The second set of significant tests conducted for information access using SMS terms were *descriptive analysis* and *multivariate tests*. Here the retrieval efficiency of three algorithms (the developed algorithm *SMSql* and the other two algorithms *tf-idf* and *naive*) were compared.

### 3.3 Test data collection

The developed algorithms were validated using the corpus prepared and collected in four different ways:

1. 1000 SMS messages were collected from a group of first year Computer Science and Statistics students in a university community. This set of SMS messages was collected using two different electronic platforms, (*Mxit* and *blue-tooth*) and some were collected in handwritten format. Participants in the latter were required to rewrite the same questions, assuming they were personally sending the questions via SMS. For all the methods used, a laptop was configured to serve as a database server. It received all forms of text message from the participants, capturing the way they responded to the question provided (*Appendix A*).
2. FAQs were gathered from more than 15 websites (see *Appendix E*), literature, books, journal articles, conference proceedings, HIV/AIDS seminars, and workshops, all of which talked about HIV/AIDS-related issues regarding awareness, education, prevention, medications, and therapy. For this experiment, an FAQ database consisting of over 350 sampled questions was built with the focus spread across various HIV/AIDS issues: drug administration, prevention, control and support, counselling, food prescription, awareness, sex education, and education and training. Of these sample questions, about 200 were extracted from Ipoletse call centres [109] and the remainder were retrieved from related websites. The Ipoletse database consists of most frequently asked questions about HIV/AIDS and ARV therapy, the booklet was prepared by the Ministry of Health in Botswana. The websites collate extensive information on the HIV/AIDS epidemic in FAQ forms, including, for example, aspects of drug administration, therapy, sex education, food and nutrition, physical exercise and treatment. The collections were assembled over ten months.
3. A corpus of SMS texts was collected from Liu [142] and Tagg [217] with their permission (see *Table 3.1*.)

TABLE 3.1: Liu and Caroline corpora

Features	Liu	Caroline
Number of messages	11,036	20,036
Number of words	190,099	85,866
Average word	4,012	*
Average number of words per message	15.28	3.5
Average number of characters per message	18.24	8.2
Average characters per word	4.65	1.8
Character (no space)	*	216,968
Character (with spaces)	*	301,837
Number of SMS in the corpus	*	14,012

\* Not available

4. The Online Collins dictionary, with a total of about 40,000 English words, was used for the research. In addition, terms such as abbreviations, acronyms, prepositions, homophones, punctuation and medical jargon related to HIV/AIDS were collected as part of the database. Words in the preposition database serve as stop words. Stop words is the name given to words which are filtered out prior to, or after, processing of natural language data (text) [121]. Medical jargon was retrieved from different HIV/AIDS websites when FAQ samples were collected. The FAQ collection forms a major component of the database used in this research. An electronic version of Collins dictionary was sourced from the web (<http://www.collinslanguage.com/wordlist.aspx>), and about 40,000 lexicon-type resources were constructed for use in this experimental system for the automated normalization of irregularly-formed English, used in day-to-day communication, in the research domain. This approach is similar to that used for the text normalization objective, where 1,255 entries of a lexical type were gathered in the rule-based approach introduced by Clark and Araki [56].

In pre-processing the English database dictionary, words that featured more than once were pruned down because such repetition is not important, either syntactically or semantically, for the purpose of normalization. For instance, *bank* can have more than four usages or meanings in different contexts, such as money, the river's edge, reliability (to "bank on" something), store (to "bank on his reputation") and so on. Storing just one appearance of *bank* is enough to represent all other forms of *bank*. The scope and volume of the dictionary database can be increased, which will become necessary because language is dynamic and new words keep appearing [167].

### 3.4 Description of the data structure and methodology used in SMS normalization

The description of the data structure and the flowchart methodology used in achieving SMS normalization are the main focus of this section. The data structure of the dictionary is modular. A modular system [87, 158] is an approach that subdivides a system into smaller parts (modules) that can be independently created and then used in different systems to serve multiple functionalities. The data samples in the database design are grouped in different tables, as shown in *Table 3.2*.

TABLE 3.2: SMS normalization database design

id_no	Table	FIELD and TYPE
1	frequently_used_smswords	id[int(255)]; word[(100)]; meaning [(100)]
2	acronyms_and_abbreviation	id[int(255)]; word[(100)]; meaning [(100)]
3	punctuation/prepositions	id[int(255)]; word[(100)]; meaning [(100)]
4	homophones	id[int(255)]; word[(100)]; meaning [(100)]
5	English_and_medical	id[int(255)]; a, b, c, ... , z[(100)]

The table design accommodates different individual modules, including *frequently\_used\_smswords* (*Figure 3.6*), *acronyms\_and\_abbreviations*, *punctuations/prepositions* and *homophones*, each table having three fields (*id*, *word* and *meaning*) and the specified types. The *English\_and\_medical terminology* table has twenty six (26) fields plus the *id field*; English words are stored alphabetically in the table. Each of the tables has its own design, to enhance the objective of normalization. Overall, the database structure has about 45,000 records.

The table of *frequently\_used\_smswords* (*Figure 3.6*) consists of three fields (*id*, *word* and *meaning*) and the other tables may be seen in *Appendix C*. SMS words commonly used are stored in the *word* column and the corresponding *meaning* field gives the meaning of the text. The idea behind the design is very simple because the aim is to produce an immediate result for SMS words during translation.

The *acronyms/abbreviations* table consists of three fields (*id*, *word* and *meaning*). The SMS abbreviations commonly used are stored in the *word* column and the corresponding *meaning* field gives their meanings. Terms like *brb*, *asap*, *lol* are stored in the *word* column of the table. During translation, the corresponding *meaning* is substituted for the word. The *punctuation/prepositions* table stores the punctuation marks and prepositions, e.g. @, ,, /, ", ?, ), (, I, r, is, a, the, on, etc. The *homophone* table consists of three fields (*id*, *word* and *meaning*). SMS commonly used words of this type are stored in the *word* column and the corresponding *meaning* field gives the meaning. Terms like *c*, *2*, *4* are stored in the *word* column of the table. During translation, the corresponding

id	word	meaning
1	wy	why
2	y	why
3	hu	who
4	wu	who
5	wht	what
6	wyt	what
7	wat	what
8	w@	what
9	wt	what
10	wot	what

FIGURE 3.6: Frequently\_used\_smswords table

meaning is substituted for the individual term. *English\_and\_medical\_terminology* is the largest table which stores the major part of the dataset. There are twenty-six fields, labelled with the letters a-to-z. Each table contains alphabetized English words under the initial letter as a label (e.g. all the words in field *a* are English words that start with the letter *a*).

As shown in *Figure 3.7*, this system is able to manage the collection of five independent sets of tables: frequently\_used\_smswords, acronyms\_and\_abbreviation, punctuation-s/prepositions, homophones and English\_and\_medical words to resolve SMS syntax. The design is intended to accomplish two purposes: SMS normalization and information access for SMS communication. It accepts queries written either in SMS or in Standard English. The query is taken as a token and is parsed down the modules for normalization processes.

SMS translation in this architecture is easy but requires intensive application of translation resources and information retrieval (IR) techniques to enhance the system's performance. The translation resources and IR techniques are exhibited in the developed normalization algorithms. This approach investigates not only single words but also phrases i.e. combinations of words in both normal English and health-related terminologies such as that related to HIV/AIDS. Different word lengths are used to discover or confirm the system's performance/robustness with respect to the language and the type of query. Part of the primary concern of the experiment is to investigate (1) how many SMS word queries adopted in the domain are answered using this translation technique and (2) how many SMS queries are answered correctly, especially in relation to the gold standard. The gold standard represent the ideal analysis which the translated results hope to achieve [153].



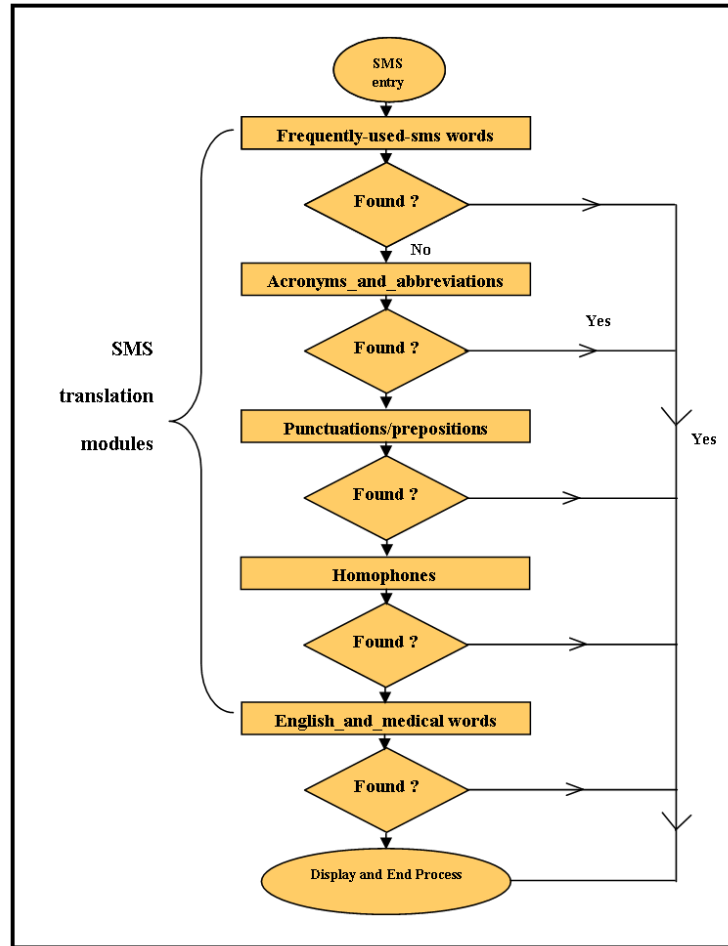


FIGURE 3.7: SMS normalization flowchart

### 3.5 The developed SMS normalization technique—SCORE algorithm

The proposed algorithm is referred to as the Search, Compare and Replace, or SCORE algorithm. The algorithm makes assumptions about the form and nature of SMS communication, such as the use of homophones, punctuation, preposition, acronyms and abbreviations. The following other assumptions also motivate the development of the proposed algorithm:

- SMS text tends to be shorter in word-length than the normalized counterpart [127, 241]
- SMS terms most often have the first character the same as the corresponding normalized terms [8, 83, 132, 238]
- The ordering of characters in SMS text corresponds to the normalized word [175]. This is a very logical assumption since it makes no sense to change the character position while writing SMS.

—Each SMS token is directly derived from its normalised form.

The dictionary is stored with words separated into 26 bins according to their initial letter, so that each bin contains only words starting with the same letter. The aim of the SCORE algorithm is to normalize SMS text into proper English.

The SCORE algorithm processes SMS input text  $T = t_1 t_2 \dots t_n$  where the  $t_i$  for  $i \in [1 \dots n]$  are tokens delimited by spaces or punctuation marks such as ., :, ;, ?, /, @, etc. The tokens are then usually presented to the system one-by-one in the order that they appear in the SMS text. Each token  $t_i \in T$  is in turn organized into its component sequence of individual characters  $t_i = [c_{i1} c_{i2} \dots c_{im_i}]$  where,  $|t_i| = m_i$ ;  $E_c$ , where  $E_c$  is the candidate word such that its wordlength  $|E_c| \geq 2$ ;  $H$  = homophone dictionary, and also can contain digits  $Z \in [0..9]$  and their interpretations, such as 4 meaning 'for', etc.

---



---

### SCORE algorithm

---

1. The first of the character  $c_{i,1}$  is used to determine in which alphabetical bin to search for contending or promising words.
2. If  $c_{i,m_i}$  is a digit in the set  $Z$  or a single letter like  $c, u, r$ , then run Step 10, i.e., the Replacement (R) or Homophone expression algorithm.
3. If  $c_{i,m_i}$  and subsequent  $c_{i,m_i+1}, c_{i,m_i+2}, c_{i,m_i+3} \dots$  are identical, run Step 9 i.e. the Repeated character deletion (D) algorithm
4. For subsequent characters  $t_i \setminus c_{i,1} = [c_{i,2} c_{i,3} \dots c_{i,m_i}]$  until the end of the word or token, retrieve database entries that have *common subsequences* that are close to  $t_i$  in the sense that characters in the entry are in the same sequence but may not contain all the characters in  $t_i$

---



---

SCORE ALGORITHM (continued)

---

5. The normalized SMS word is determined by the LCS *algorithm*:

---

LCS( $E, T$ )

---

 $m \leftarrow \text{length}(E)$ 
 $n \leftarrow \text{length}(T)$ 
**for**  $i \leftarrow 0$  **to**  $m$  **do**
 $\text{size}[i, 0] = 0$ 
**for**  $j \leftarrow 1$  **to**  $n$  **do**
 $\text{size}[0, j] = 0$ 
**for**  $i \leftarrow 1$  **to**  $m$  **do**
**for**  $j \leftarrow 1$  **to**  $n$  **do**
**if**  $E_i = T_j$  **then**
 $\text{size}[i, j] \leftarrow \text{size}[i - 1, j - 1] + 1$ 
**else if**  $\text{size}[i - 1, j] \geq \text{size}[i, j - 1]$  **then**
 $\text{size}[i, j] \leftarrow \text{size}[i - 1, j]$ 
**else**  $\text{size}[i, j] \leftarrow \text{size}[i, j - 1]$ 
 $I \leftarrow$  number of insertions

**return**  $\text{size}$ 


---

6. Compare the return sizes by looking at the Number of Insertion (I) operations that will be needed to convert  $n$  to  $m$  i.e.  $\text{length}(T) \rightarrow (E)$

7. If Step 6 has more than one value of  $LCS$  i.e. the translated  $E$  is more than 1; Then the matching words are sorted according to the value of *word error rate* or  $WER$ .

The  $WER$  is calculated as follows, where  $R$ ,  $I$ ,  $D$  and  $N$  are numbers of replacement, insertion, deletion and, word-length of the matching or competing words  $E_c$  respectively.  $WER = \frac{R+I+D}{N}$ .

The matching or competing word with the lowest  $WER$  is selected and is yielded as the result of the normalization, output and go to Step 11.

The  $WER$  can be used to calculate the *word accuracy rate*,  $WAR$ , that is,  $WAR = 1 - WER$

8. If there is tie in Step 7 i.e.  $WER$  is the same, ties can be broken by using the *order of vowel precedence algorithm* i.e. **if**  $E_c \in \{E_{cm_1}, E_{cm_2}, \dots, E_{cm_n}\} = WER$  **and**  $E_c$  contains vowels, then select  $E_c$  that has its first vowel as  $e$  follow by  $a, i, o, u$ , **return**  $E_c$  as the output and go to Step 11.

---



---

SCORE ALGORITHM (continued)

---

9. Run *repeated characters deletion algorithm*

---

**Deletion algorithm**


---

$c_{i,m_i} \leftarrow a, b, c, \dots, z, 0, 1, 2, \dots$

**if**  $c_{i,m_i}$  is repeated more than twice **then**

$T_{new} \leftarrow$  keep first  $c_{i,m_i}$  and delete rest

$D \leftarrow$  number of deletions

**return**  $T_{new}$  to Step 3

---

10. Run *homophone expressions algorithm*

---

**Replacement algorithm**


---

Search  $c_{i,m_i}$  in  $H$  dictionary

**if** found **then**

Replace  $c_{i,m_i}$  in  $t_i$  and then concatenate

$T_{new} \leftarrow$  concatenate

$R \leftarrow$  number of replacements

**return**  $T_{new}$  to Step 2

---

11. End.
- 

### 3.5.1 Further description of the SCORE algorithm

A step-by-step description of the SCORE algorithm follows.

**Step 1.** Starting with the first SMS input character, access the bin for the corresponding letter in the database.

**Step 2.** For every SMS input character that is a digit or single-letter, the searching is done in the homophone table (Section 3.4). This is a replacement algorithm whereby the digits or single-letter words are replaced with their corresponding meaning from the table. For example, *4evr* will be normalized by replacing the *4* with *for* to become *forevr*. The *homophone expression algorithm* is run in Step 10. The output is returned to Step 2.

**Step 3.** Exclamatory expressions usually involve repetition of a letter, e.g. *whaaaaoooo!!!*. For any character that is repeated more than twice, the first letter is kept and the other

repeated letters are deleted. The *repeated characters deletion algorithm* is run in Step 9 and its output is returned to Step 3.

**Step 4.** For subsequent characters, until the end of the SMS input word, output corresponding words from the database that have the same ordering of characters. Select the word(s) with the highest number of matching ordered characters.

**Steps 5.** The normalized SMS word(s)  $E$ , is determined through a process of character insertion (Step 5), deletion (Step 9) and replacement (Step 10) on the SMS token  $T$ . For example,

The insertion of character  $\sigma$  at  $k$ th position results in

$$T[i, j]T[i, j + 1] \dots T[i, j + k - 1]\sigma T[i, j + k]T[i, j + k + 1] \dots T[i, n + 1].$$

Insertion is the process of adding the missing characters to an SMS word in order to complete the spelling sequence. After insertion, the character position of the SMS and English word must be the same (Sections 2.7 and 2.8).

The edit distance between the two strings can only be zero provided they are similar, otherwise there is a cost paid for the transformation. The cost is measured as a weight. The weight for a given edit sequence is the ratio of edit distance operations to its word-length, and the minimum of this ratio over all edit sequences is the normalized edit distance. Edit distance is therefore the total number of operations that are needed to make two dissimilar strings similar [60]. Edit distance is sometimes referred to as the Levenshtein distance [51, 137, 198].

**Step 6.** From Step 5 the entry with the longest matching common subsequence is selected. In case the result, i.e.,  $E$  term, is more than 1, then follow the procedure in Step 7

**Step 7.** From the candidate (English) lists, calculate the word error rate (WER) given by:  $WER = \frac{R+I+D}{N}$ ; where  $R$  is the number of replacements,  $I$  the number of insertions,  $D$  the number of deletions and  $N$  is the word length. Output the word with the least  $WER$ , i.e., the least result is taken as the result for the normalization;

**Step 8.** In case there is a tie, the *order of vowel precedence* is used for disambiguation (Sections 2.8 and 2.9). In case there is a tie, for instance, the edit distance of *clndr* in the candidate lists of *calendar*, *colander*, *cylinder* is equal, 0.375, i.e. 37.5%. The highest probability of usage of vowels or percentage of occurrence of vowels in the text will be followed: Hence *calendar* will be selected as the most likely translation. In English literature, the occurrence of words that have letter *e* is highest and followed by the order

of precedence  $a$ ,  $i$ ,  $o$  and  $u$ . In the dataset English words with the vowel  $e$  are most abundant.

**Step 9.** The repeated characters deletion algorithm, i.e, the deletion algorithm is run.

The deletion of repeated characters  $\sigma\sigma\dots\sigma$  at positions  $k + 1, k + 2, \dots, k + r$  in

$$T[i, j]T[i, j + 1]\dots T[i, j + k - 1]\sigma\sigma\dots\sigma T[i, j + k + r - 1] \dots T[i, n];$$

results in

$$T[i, j]T[i, j + 1]\dots T[i, j + k - 1]\sigma T[i, j + k + r - 1] \dots T[i, n];$$

Where  $r - 1$   $\sigma$ s have been deleted.

The deletion operation occurs when a character,  $\sigma$ , which appears in an SMS input is repeated more than twice. The limit was taken as two because there are English words that have letters repeated twice, e.g. *good, ball, feed*, etc. The letters in excess of two is reduced to one letter by deletion (Sections 2.7 and 2.8).

**Step 10.** The homophone expressions algorithms, i.e., replacement algorithm, is run.

The replacement of character  $\sigma$  at the  $k$ th position with a homophone  $H[1]H[2] \dots H[r]$  of length  $r$  results in

$$T[i, j]T[i, j + 1] \dots T[i, j + k - 1]H[1]H[2] \dots H[r]T[i, j + k + 1] \dots T[i, n];$$

where  $r$  is the word length of the homophone.

The replacement operation occurs when a character,  $\sigma$ , which appears in an SMS input is not a constituent of an English word  $E$ , e.g. *b4, un4tun8, 2day*. In these cases, the digit is replaced with its corresponding *meaning* which have been saved in the homophone database (Sections 2.7 and 2.8).

**Step 11.** End of the algorithm and the results of the SMS normalization using the SCORE algorithm is displayed.

### 3.6 Experimentation methods for SMS normalization

Six experiments and the methods for carrying them out are presented. The results of each experiment will be discussed in Chapter 4. The experiments include: *Vowel stripping, Clipping positions (SMS taxonomy normalization accuracy), Frequency or probability*

*model (posterior probability measurements), Evaluation of SCORE algorithm on dataset of Caroline and Liu, Annotator's translation and Cross validation.*

Some of the experiments involve manipulation of variables in order to determine whether such changes will affect other fixed variables or parameters. The experimental methodology relies on controlled procedures, random assignment, and manipulation of variables, to confirm the research questions. An experimental methodology provides fuller detail for an evaluation, including greater reproducibility of parameters, data processing, details of toolkits used, etc. [160].

### 3.6.1 Experiment 1—Vowel stripping

*Vowel stripping* was used for systematically testing the efficiency of our algorithm. The method is known by Pennell as *deletion-based* abbreviation [185]. Pennell and Liu [183] generated multiple character extraction from English tokens and then performed a reverse translation of the extracted terms. They created a look-up table by listing all the reasonable translations of the abbreviated word. An annotator was used to decide the level of *reasonable* translation. Yang et al. [236] worked with abbreviation generation on spoken Chinese text messages. Their research used a *conditional random field* (CRF) as a binary classification to determine the probability of removing a Chinese character to form an abbreviation. In this experiment, an algorithm to strip off all vowels in each word in the English\_and\_medical terminology database was created. Words with a vowel as first character were left untouched in order not to contradict the hypothesis in Section 3.5. The hypothesis is that the initial letter of an SMS and the equivalent English word are usually the same. The intention of using *vowel-stripped* words is to pass the words into the machine translation algorithm, and then test if the exact word that had its vowel stripped will be returned. The process of vowel extraction, and processing the output to serve as an input, took about 103 milliseconds to complete. An Intel i3 Dual core, 4GB RAM computer with a 320GB hard drive running with the Windows 7 Operating system was employed to run and compile this experiment.

Prior to processing, word length and frequency were taken into consideration in building the dictionary database. The use of word frequency to estimate word difficulty is based on the assumption that difficult words appear less frequently in a corpus. Words with high frequency counts are used in training and testing. Breland [34] has shown that word frequency is a good measure for determining word difficulty. English has an average word length of 5 letters [29]: the shorter the word length, the higher the word frequency and vice versa [75]. A total of 15,000 English words, each with not less than 5 characters, was used for this and subsequent experiments.

### 3.6.2 Experiment 2—Clipping positions

The second experiment involved using the platform with the SMS classification reported in Section 2.2. One hundred (100) most frequent words were selected randomly from the data sets. The data set was stripped of characters, using *vowel dropping*, *medial clipping*, *mixed clipping*, *end stripping*, and *initial clipping*, each time in different positions within the word length, and a normalization evaluation was conducted in each case.

### 3.6.3 Experiment 3—Frequency or probability model

The method of posterior probabilities using the Viterbi algorithm was used to determine whether a given word will be the most frequently used SMS term for a corresponding English term. *Posterior probability* is the possibility of an event, A, occurring, given that event B has occurred. The Viterbi algorithm is often looked upon as minimizing error probability by comparing a set of possible state transitions that could occur, and deciding which of these has the highest probability of occurrence [202]. The SMS translation model in this research involves using the concept of a noisy channel model [19, 59, 235], i.e. a *supervised* learning approach, as described in Section 2.3, pairing SMS and the corresponding English expression in the training set. The probability table is created for SMS query terms made available to a set of students at the University of the Western Cape. In a phrase-based statistical machine translation system, the phrase translation table is the defined component which specifies alternative translations and their probabilities for a given source phrase. In learning such a table from parallel corpora, two related issues need to be addressed (either separately or jointly): which pairs are considered valid translations, and how to assign weights, such as probabilities, to them [69, 103, 129].

The frequency distribution (probability) model is used as the background theorem for the experiment. A histogram showing the highest representation within the samples collected in 10 SMS samples data from 100 students captures the term frequency i.e. the percentage occurrence in the results. For each query, 100 SMS samples were taken from the students. Corresponding English terms were randomly selected for the experiment. These collections show the pervasive, liberal and uncompromising communal creativity and intuitiveness among peers, predominantly the youth of the University, as represented in their SMS communication.



### 3.6.4 Experiment 4—Evaluation of two set corpora with SCORE algorithm

Another evaluation test for SMS normalization was established in two sets of corpora from Tagg [217] and Liu et al. [142]. This experiment depicts the comparative evaluation of various implementations conducted on these sets of data. The contrast was achieved using two baseline systems. Loading the SMS corpora of Liu into a Microsoft Word processor raised the error message:

*There are too many spelling or grammatical errors in the SMS set, to continue displaying them. To check the spelling and grammar of this document, choose Spelling and Grammar from the Review tab.*

This message is a characteristic feature of SMS. It shows the extent of creativity in the corpus, for example: cya → see you; tomoz → tomorrows; numba → number; prez → present; ursef → yourself; orite → all right.

To evaluate the two sets of corpora, the word error rate (WER) and sentence error rate (SER) were used. BLEU is an alternative method that allows comparisons with other similar studies, like those proposed by Aw et al. [20] and Kobus et al. [127]. The metric WER and SER show distribution of errors within the sentence or multiple words.

The evaluation task is to test whether the SCORE algorithm will be able to normalize or correct the erroneous terms that feature in the Tagg [217] and Liu et al. [142] corpora. A summary of the two corpora was presented in *Table 3.1*. The corpora are expected to possess a significant and reasonable percentage of normalization in relation to the English word equivalent present in the English dictionary. The two corpora are run with the SCORE algorithm and the results are presented based on words and sentences available in the corpora. The result could be either a single or a multiple word.

### 3.6.5 Experiment 5—Annotator translations

Annotators were used to conduct three different, related experiments, by first translating (1) English terms to SMS, (2) SMS terms to English and then comparing them with (3) translations provided by annotators having prior knowledge when producing their translations.

#### **Experiment 5a: From English → SMS**

Participants in the experiment are mobile users who were provided with 56 queries to be sent using SMS. Some of the messages sent will be out-of-vocabulary because of user

abbreviations or compaction of the queries. Students of the University of the Western Cape, with an average age of eighteen, were involved in this experiment.

These students were very adept at SMS communication. A simple input interface was created for the experiment in which SMS sentences will be interpreted or translated. These SMS messages were used to enrich the dataset with neologisms used by the students. The messages were then submitted to the SMS translator.

The translation of the queries was observed by a set of three (3) annotators hired for this evaluation in September 2012. The annotators were asked to translate the English terms into SMS words using the algorithms. The English terms were provided for the annotator to work from for such translation. Annotator judgment is based on the criteria of Success, False success and Failure i.e. (1) successful translation (2) unsuccessful translation and (3) return of the exact SMS input. *Successful* translation occurs if the translation yields the intention of the *texter*; otherwise it is unsuccessful. *Unsuccessful* may still be seen as a situation in which the interpretation is done but gives wrong results. This is a *false success* or *false positive*. The third category is when the algorithms cannot translate the SMS text and as such the input is returned unchanged.

#### **Experiment 5b: From SMS $\rightarrow$ English**

SMS variants of Standard English words were collected by reversing the translation tasks from Experiment 5a for each SMS query sentence. This was done by listing as many reasonable informal texts for a given English word found in the set of SMSs based on the query sentences.

Every SMS word used in the form of a query has a formal format in the English dictionary. The same 100 English words were given to each annotator from the 56 query sentences (each question has an average of two terms to be transcribed into English). Three students who had never had the privilege of knowing the query sentence were requested to reverse the SMS words created from Experiment 5a into their original format.

#### **Experiment 5c: Annotator with/without prior knowledge**

In this experiment, the annotators had the twin privileges of (1) no knowledge of the datasets before the translation was done; and (2) knowledge of the datasets to be interpreted before they were asked to translate from SMS to English. This is an approach used by Gouws et al. [94].

### 3.6.6 Experiment 6—Cross validation

*Cross validation* was used for the purpose of this experiment. The system is similar to that of Pennel [185] and Aw et al. [20], but the algorithms differ. The dataset used for this experiment was divided into ten (10) bins without bias. Each bin was given to an annotator (A). The datasets were later studied to confirm the proportion or percentage of (1) noisy text and (2) the type of noisy text. The proportions of SMS and formal English were recorded. The exercise was performed with Microsoft editor and two other professional editors from the Writing Centre at the University of the Western Cape. The Writing Centre provides a supportive academic environment in which students can receive advice, guidance and constructive assistance with written tasks and assignments, or any other creative or personal writing. Wrong spelling, acronyms, abbreviations and clippings are considered as SMS texts. The summary of each bin is given in *Table 3.3*.

The dataset contains a reasonable proportion of phrases set apart from single-word terms. The collection was selected for interest while the performance of the data was collected for evaluation [145]. The SMS corpus contains a set of phrases from the Internet: phrases such as *life is beautiful*, *mode of communication*, *transformation by legitimate intervention*, *University of the Western Cape* etc. The average number of characters used for the phrase sets in the experiment was between 13 and 42 (*mean*=25.6). Altogether, there are 1109 unique words, with an average number of words of 3.2 per phrase set for each bin. The advantage of using a predefined phrase set gives both internal and external validity to the results. The internal validity is attained if the effects observed are attributable to the controlled variables or parameters, while the external validity means the results are generalizable to other subjects and situations [145]. Phrases are worked upon randomly as just single words from the basket by all annotators. This is a procedure preferred by the majority of the research studies.

TABLE 3.3: Summary of the SMSs in each bin

Bin#	SMS Available
1	148
2	79
3	108
4	98
5	124
6	103
7	138
8	112
9	92
10	107
Total	1109

Postgraduate linguistics students (annotators) were hired and specially trained for this assignment. The annotators were requested to translate the SMS text into formal text. The BLEU scores awarded are based on the relevant judgements presented in *Table 3.4*. Twenty SMS query terms were selected at random by each annotator from the bin. The corresponding English word for the query terms (i.e. noisy  $\rightarrow$  clean) was labelled and hidden from the annotator. The annotator provided possible translations of each query term selected in the bin. The translation variants were compared with the relevant judgment scale in awarding the BLEU scores. The annotators could use the scale to determine the degree of correctness. The BLEU score measures the accuracy of the search results, i.e. how close the words listed are to the search results the user is looking for [52, 93].

The accuracy of a translation is judged from the BLEU score. A BLEU score requires a gold standard, i.e. the structure representing the ideal analysis which the translated results intend. The two results, machine translation (SCORE), and human judgment (BLEU), are compared. A score ranging between 0 and 1 is assigned. A score value of 1 shows that human judgment and machine translation are the same; if the translation is completely opposite the value is 0.

The metric values—SCORE algorithm values and BLEU scores—are calculated from each bin by translating 20 SMS terms initially provided for each annotator. The relevant score is coupled with the *N-best* approach according to *Table 3.4*. A maximum of 1.0 is scored when the translation is exact. The position of the exact translation determines the score it has in the relevance scale. The average score is recorded as the score for the operation.

TABLE 3.4: Relevance scores

Relevance scores	Range
Excellent	1.0
Very Good	0.8 - 0.9
Good	0.6 - 0.7
Moderate	0.4 - 0.5
Poor	0.2 - 0.3
Very Poor	0.0 - 0.1

The *N-best* approach is used to confirm the best result after SMS normalization has been carried out. The *N-best* list contains N ranked hypotheses for the user's text, where the top entry is the search engine's or annotator's best hypothesis. When the top entry is incorrect, the correct entry is often contained lower down in the *N-best* list. For an SMS normalization system to make use of the *N-best* list, it is useful to estimate the probability of correctness for each entry, and the probability that the correct entry is

not on the list [232]. In order to apply the relevant scores effectively, the annotators are helped by being given different translations of the SMS query terms. The annotators have to be aware that the order or position in which the results appear shows the degree of certainty and conformity for each interpretation. Each is rated using *Table 3.4* based on the position of the correctness of the actual word.

### **3.7 Description of data structure and methodology used in information access using SMS in a FAQ system**

In this section the data structure and the methodology adopted in carrying out information access in an SMS-based FAQ system will be examined. The methodology led us to adopt the SMS query algorithm for retrieval techniques in the health domain. The developed *SMSql* algorithm involves the use of a web-server application which automates the retrieval tasks of the FAQ system. The retrieval process entails providing the five most relevant answers to a user enquiry. This is similar to the baseline of Mogadala et al. [166]. Communication is triggered by the SMS sent by the user and received by the system which acts as a server. The preliminary process translates the SMS term into its English form and then the noise-free query is parsed using the SMS parser (SCORE algorithm) as described in Section 3.5. Extracting the best matching question-answer pair in the server is the ultimate goal. This is achievable by statistically selecting *keywords* and *idioms* from the query corpus in the FAQ query-set gathered earlier during data test collection (Section 3.3). The keywords and idioms are a combination of words or phrases that give a reasonable meaning to each query. From the keyword phrases, idioms can be derived. An idiom is a collection of words with a specific semantic meaning taken as a group, and which may not yield the same meaning when interpreted individually as words and not collectively as a phrase [14, 104, 197].

Based on the perspective of the research question, computational time is used as an evaluation metric for the effectiveness of the new algorithm, and in assessing the efficiency of mobile information access using the *SMSql* algorithms. This result is achieved by measuring the time it takes to return answers when SMS is used as a query to a FAQ search engine. There is a need to confirm whether the returned answer from the FAQ system is relevant or non-relevant. The system is expected to produce relevant answers to the normalized SMS query. If the searching process does not provide a relevant document for the user's information, the user can then modify and reformulate the query.

The methods adopted for the experiment in information access using SMS is divided into five parts for the purposes of discussion, with each section aiming to assess the relevance of the method to the developed *SMSql* algorithms.

### 3.7.1 Architecture, procedure and extraction process in *SMSql*

In *Figure 3.8*, the architecture of the *SMSql* consists of a web server connected to the Internet and to a mobile phone. The client is a user with a mobile phone who sends an SMS message to the server which then processes the search query. Before the SMS query is dispatched to an FAQ search engine, the process of SMS translation becomes necessary. There is a need to convert the SMS query into all possible representations of an English version. Mobile users may write several different SMS texts for the same query. This form is now used for pattern matching in the FAQ-SMS database and subsequently the result pages are downloaded. The reformulation process will take place if the answer given to the request does not satisfy the expectation of the user. The server extracts the results from the downloaded Q&A pairs, and distils them to a maximum of 140 bytes because of limited mobile phone capacity and bandwidth restrictions [50]. Finally, the server returns the results to the user that issued the request. It is worth mentioning that the results are a ranked list of FAQ queries that correspond to the SMS query. The extraction process at the FAQ database server is the heart of the SMS-query.

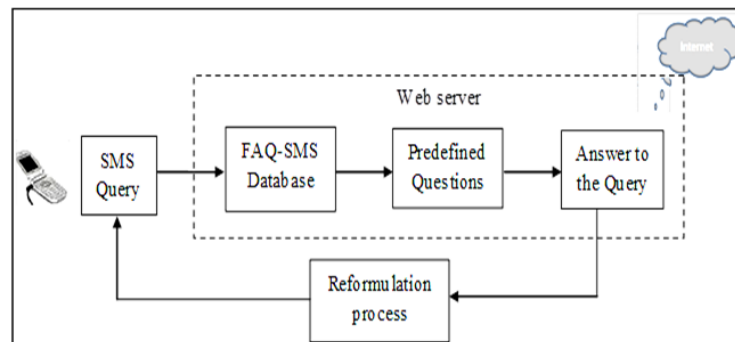


FIGURE 3.8: System architecture of an SMS-query and reformulation process

There is provision for reformulation of the query in the event of it not being available in the FAQ database. The FAQ database is updated to number among the predefined questions which serve as an area of supervised learning for the system architecture. Input to the system is a search SMS query in the form of a request, where the query represents the actual search terms and the context specifies the type of contextual information that the user expects the system to extract. During the extraction process, the system can gather results in the form of  $n$ -grams from a corpus of words from the FAQ database, where an  $n$ -gram is simply any set of  $n$  space delimited terms found amongst those FAQ corpus words. The  $n$ -grams are measured and then ranked. The most highly ranked result is then returned to the user as the answer to the request.

The objective of the SMS parser is to get a unique result that corresponds to its translation regardless of the text message format. Parsers process, analyse and, importantly,

reformulate the messages for further SMS normalization and natural language processing. The SMS parser requires many different mobile inputs that represent a particular query or question, and these are mapped within the FAQ databases. The parsing involves the training files (dictionary, HIV/AIDS queries, Ipoletse question sample), the input files/phrases/query (SMS queries) and the output results (mapped question-answer pairs) which lead to the retrieval of appropriate answers. The retrieval of ranked results was carried out on the local data FAQ database. The context of the evaluation was the health domain.

### 3.7.2 Flow diagram of *SMSql*

A text message is sent from handset. The text is normalized at a pre-processing stage before it is used for information searching. The process of SMS normalization/translation has been described in Section 3.5. From *Figure 3.9*, the noise-free text message serves as an input query to the FAQ English database. Queries are extracted from the database based on keyword and n-gram matching. The use of *n*-grams is described in Section 3.7.3.

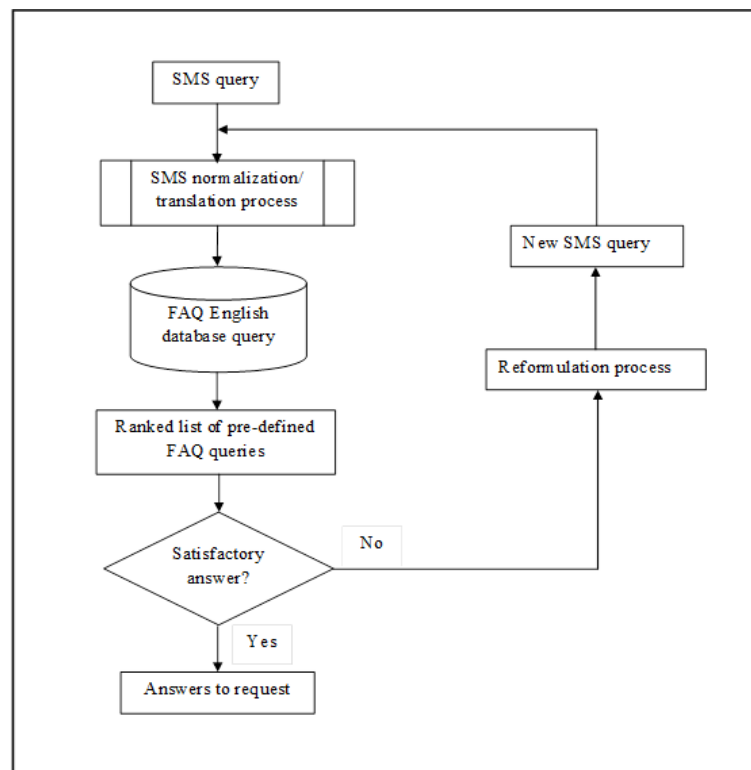


FIGURE 3.9: Flowchart of SMS question locator (*SMSql*)

These sets of queries are ranked according to their weight (relevance), and the user gets back the results of his/her enquiries and determines the level of relevance. The

user's judgment will determine if there will be a reformulation process for the reply, i.e. whether the results are satisfactory or not.

### 3.7.3 Applications of $n$ -grams in SMS-based information retrieval system

Text characterization and manipulation can be done on an individual character represented as a byte-level operation, or on the entire word used by the individual. The use of  $n$ -grams stands out as an effective tool for the textual computing process over conventional character-based or word-based approaches. As an illustration of their generality,  $N$ -grams play a role in word-matching, error detection, the correction of spelling errors, string similarity measurement, text retrieval and searching, language identification and biological sequence computing [199].

An  $n$ -gram is a substring of length  $n$  characters derived from a text string; usually, but not necessarily, a word, containing not less than  $n$  characters. The characters in the  $n$ -gram retain the same order as in the source text from which the  $n$ -gram has been derived [81].

Common example of  $n$ -gram operation on a word *medication*

---

$n = 2$ ( <i>diagram</i> or <i>bigram</i> )	<i>me, ed, di, ic, ca, at, ti, io, on</i>
$n = 3$ ( <i>trigram</i> )	<i>med, edi, dic, ica, cat, ati, tio, ion</i>
$n = 4$ ( <i>quadgram</i> )	<i>medi, edic, dica, icat, cati, atio, tion</i>

---

A character word *character word length, r* will yield  $(r-1)$  *bigrams*;  $(r-2)$  *trigrams*,  $(r-3)$  *quadgrams*, etc. There are many different types of string-similarity measures but  $n$ -gram based measures are probably the most widely used, where the degree of similarity between two strings of characters is based on the number of  $n$ -grams.

Comparing other variants of *medication*, e.g. *mdcaton*, reveals the  $n$ -grams

---

$n = 2$	<i>md, dc, ca, at, tn</i>
$n = 3$	<i>mdc, dca, cat, atn</i>
$n = 4$	<i>mdca, dcat, cato, aton</i>

---

The degree of similarity between the two words (SMS query and FAQ dataset) is then calculated by means of a similarity coefficient such as the Dice's or Overlap Coefficient (see Section 2.6.1). If one word (SMS query) contains  $X$   $n$ -grams, and another (FAQ data set) contains  $Y$   $n$ -grams, and  $Z$  of these are common, the Dice's coefficient is

$$\frac{2Z}{X + Y}$$



while the overlap coefficient is

$$\frac{Z}{\min(X, Y)}$$

### 3.7.4 Scoring and ranking techniques

The search technique ultimately aims at giving the user a ranked list of relevant documents. A lot of time is spent in looking for relevant information from a collection of documents (i.e. SMS queries and question-answer pairs). One of the methods adopted in arriving at a ranked list is assigning weights to the relevant terms. This shows the degree of importance of the terms (tokens) in the documents. The relevant score finally determines the position of the documents (the question-answer pair) when it is sent out as the end product of the enquiry process.

The following methods of acquiring the ranked list are, *term frequency-inverse document frequency (tf-idf)*, the *vector space model*, and *cosine similarity measurement*. This is useful in calculating the score's function before the document (question-answer pair) is ranked.

A query sentence is broken down into a series of tokens delimited by spaces, in the form of a term vector

$$[t_1, t_2, \dots, t_{n-1}, t_n] \text{ for } i= 1, 2, \dots, n$$

where  $t_i$  is the  $i^{th}$  term of the  $n$ -term normalized SMS query sentence. There is a comparison of the  $i^{th}$  term between the user's question (SMS) and the questions in the FAQ files, so that the relevant questions with the same terms are selected, based on similarity and some other factors, like the number of query sentence terms, the style and content of the FAQ collection, the length and specificity of the query sentence and, the number of relevant FAQ documents.

#### 3.7.4.1 *Tf-idf* measurements

This measurement approach indexes only the terms in the documents: SMS queries and FAQ corpus. The *tf-idf* method has been useful for vector metric fields in a multi-dimensional space. It assigns a high weight to a term, if it occurs frequently in the document but rarely in the whole document collection. On the contrary, a term that occurs in nearly all documents has hardly any discriminative power and is given a low weight, which is usually true for stop words. Its accuracy in picking out terms of high significance for performing further comparisons and classification is undoubted [119].

In calculating the *tf-idf* of a normalized SMS term in a document  $D$  of a FAQ corpus, it is necessary to know two things: how often the term occurs within the FAQ data set document (*term frequency, tf*), and in how many documents of the corpus the term appears (*document frequency, df*). By taking the inverse of the document frequency (*inverse document frequency, idf*), the weight of the term in the set of the collections, i.e. FAQ data set, is thereby calculated. *idf* is represented as the logarithm of the quotient of the total number of documents ( $D$ ) of FAQ corpus and the document frequency ( $df$ ) in order to scale the values.

Thus a document is represented as:

$$D_j = (w_{1j}, w_{2j}, \dots, w_{nj})$$

where  $w_{ij}$  is the weight of term  $i$  in the document  $j$  indicating the relevance and importance of the keyword term. The *tf-idf* method measures terms in the vector and assign weights which denote the importance to the terms.

$$w_i = tf_i \log \frac{D}{df_i}$$

where  $w_i$  is the term weight in the FAQ query sentence,  $tf_i$  is the term frequency of a term  $i$  that occurs in the query;  $df_i$  is the document frequency of a document of a term  $i$  that occurs in the FAQ corpus; and  $D$  is the number of questions in the sample range of the FAQ corpus.

### 3.7.4.2 Vector space model

The *tf-idf* can now be used to create vector representations of documents (SMS query and FAQ query sentence). Each component of a vector corresponds to the *tf-idf* values of a particular term in the compared corpus dictionary. This representation of terms is referred to as a *vector space model*. A vector space model is a statistical model that models FAQ query documents and SMS queries as vectors in a multi-dimensional space [203]. The relevancy of the paired document is judged statistically by computing the cosine of the angle between the FAQ query document and SMS query vectors. The size of the cosine angle determines the degree of relevancy. For instance, if it is a small angle, this means that they are conceptually similar and relevant to the users [225]. The irregular document length makes it difficult to use the *vector space model*. Documents with similar contents but different lengths are not regarded as being similar [152]. The model is good for ranking and scoring but the shortcoming is that two documents with similar content but different lengths are scaled as being dissimilar and far apart in terms of their relevancy [209, 225].

### 3.7.4.3 Cosine similarity measurements

This method resolves the bias caused by different documents (FAQ query and SMS queries) in the *vector space model*. The vectors are normalized to unit length and the angle between the vectors, more precisely the cosine of the angle, accounts for their similarity. Cosine similarity measurement is another technique to measure the similarity between the FAQ and SMS query documents. The angle  $\theta$  between the FAQ document vector and the SMS query determines the similarity between the two documents i.e. FAQ and SMS query sets, as it is written:

$$\cos(\theta) = \frac{\sum w_{q,j}w_{i,j}}{\sqrt{\sum w_{q,j}^2} \cdot \sqrt{\sum w_{i,j}^2}}$$

where  $\sqrt{\sum w_{q,j}^2}$  and  $\sqrt{\sum w_{i,j}^2}$  are the number of words in the SMS query and FAQ documents respectively.

If  $\theta = 0$  then the FAQ query document (Doc1) and the SMS query document (Doc 2) are similar. Otherwise there is a degree of dissimilarity in the two documents, and we can say Doc2 will be more similar to the Doc1 if the angle between Doc2 and Doc1 and SMS query is smaller than the angle between SMS query document and FAQ query document.

## 3.8 SMS-based FAQ analytical methods

This section presents and discusses the three major parameters that are considered when FAQ system is to be developed: (1) the data structure on both the SMS and FAQ database, (2) keyword extraction in the question-answer pair and, (3) identifying the query codes from the query-answer pair. Each of the parameters is described.

### 3.8.1 Description of the FAQ database system

There are various ways that can be used to collect datasets for an experiment. For instance, the experiment performed by Jansen et al. [116] used log files where 74 terms were found to occur more frequently in their sample space of an average term of 100 using the Excite search engine. A collection of 1400 documents, from a United Nations database of 1988, were used in an experiment titled *using tf-idf to determine word relevance in document queries*. From the document, 86 queries were extracted to perform the experiment on information retrieval [194]. Burke et al. [41] used a total of 241 test

questions from a corpus drawn from system log files. A widely read and popular news medium, *The Times of India* blog, was used as the source of data. The blog has several datasets on topics like politics, sports, entertainment, cuisine, social evils [124].

From the FAQ query set, 20 questions were used for the analysis. These questions were to be translated to SMS shorthand by students at the University of the Western Cape. A set of 20 questions from 100 respondents yielded 2,000 SMS query formats used in our dataset; that is, each query has 100 respondents. A large collection of data was necessary in order to reduce bias in the SMS writing. *Appendix A* gives different samples collected for the research i.e. SMS translations and their corresponding English queries, that is, SMS query sentences written by English respondents. The samples are used as the basis for performance evaluation of the technique that was adopted for information retrieval efficiency in the various developed algorithms in Section 3.10. The efficiency for the three algorithms is then compared in terms of computational speed.

As shown in *Table 3.5*, the schema has three columns: (1) *Qcode*— a unique auto-incremental key that serves as the primary key (PK) for easy identification of the query and the answer pair; (2) *Query*— this attribute has a list of 350 FAQs within the domain of studies (medical); and (3) *Answer*— this attribute contains the answers to each query.

TABLE 3.5: MySQL description of the FAQ database table

Field	Type	Key	Default	Extra
Qcode	Int(255)	Primary	Null	Auto Increment
Query	Varchar(100)	-	-	-
Answer	Varchar(100)	-	-	-

The database structure for the FAQ information retrieval system has one table with 350 HIV/AIDS queries. *MySQL*, a relational database, was used to store FAQ and answers datasets for future data analysis. The primary objective of this evaluation is to compare the retrieval performance in the experiments using these algorithms: *naive query retrieval*, *tf-idf*, and *SMSql* (an algorithm the researcher has developed).

### 3.8.2 Stop-word lists

FAQ database structure, in *Figure 3.10*, is the collection of the stop words list. The stop words are a set of English words that repeat themselves within a corpus. From the linguistics analysis carried out by Tagg [217] on the SMS language, English words like *a*, *the*, *to*, *or* are rarely used in SMS texts. Mostly, single-letter words are made available to represent multiple-letter words—*d* for *the*, *n* for *and*, *r* for *are*, *u* for *you*—and when they are used they play an insignificant role in SMS-based information retrieval processes.

Stop words are very common words that appear frequently in text and carry little or no semantic meaning in an expression [77]. Leveling [136] investigated the effect of stop words at different stages of SMS-based FAQ retrieval using monolingual English language datasets. Using different experiments Leveling [136] concluded that a combination of retrieval without stop words and out-of-domain trained detection using SMART stop words yields the best results. The top twenty corrections in *Forum for Information Retrieval Evaluation* (FIRE) SMS preview data showed stop words as the most frequent error in SMS normalization—particularly the use of *d* instead of *the* [100]. At this stage, it is important to note that stop words are less important parts of the keyword phrases and are discarded. In the experiment, single character tokens are ignored during the normalization process, and they are likely to be stop words. Stop words are never considered to serve the role of the keywords.

The screenshot shows a database management tool interface. On the left is a sidebar with a tree view containing folders like 'acronyms\_and\_abbreviation', 'english\_and\_hiv', 'faq', 'frequently\_used\_smswords', 'homophone', 'preposition', 'punctuation', and 'test2'. A 'Create table' button is visible. The main area displays a SQL query: `SELECT * FROM 'preposition' LIMIT 0, 30`. Below the query are pagination controls: 'Page number: 1', 'Show: 30', and 'row(s) starting from row # 30'. A table titled '+ Options' is shown with columns 'id' and 'word'. The table contains 10 rows of prepositions, each with 'Edit', 'Inline Edit', 'Copy', and 'Delete' icons.

id	word
1	aboard
2	about
3	above
4	across
5	after
6	against
7	along
8	amid
9	among
10	anti

FIGURE 3.10: Punctuation/prepositions table

Stop words affect the retrieval effectiveness because they have high frequency and tend to diminish the impact of frequency differences among less common words, affecting the weighting process [2]. It is therefore recommended that a high frequency word  $n$ -gram that occurs in many words will need to be eliminated before computing the similarity coefficient. Weighting the remaining  $n$ -grams using an inverse frequency coefficient, that is assigning the highest values to least frequently appearing  $n$ -grams will ensure that matches between less frequent  $n$ -grams contribute more to word similarity than matches between frequent  $n$ -grams [199].

An approach of manually extracting the list of frequently used words or stop words from a *Brown corpus*, or adding missing inflectional forms to it, was described by Fox [89]. The final product was a published list of 421 stop words. This is the same approach applied in identifying the stop words in the query collection. Frequently used words were manually extracted from the FAQ query dataset of HIV/AIDS terms. Dolamic and Savoy [76] investigated the use of two sets of stop word lists and compared them

with a search approach (accounting for all word forms). Lower performance levels were recorded when using either short or long stop word lists, or no list at all, but these are usually not statistically significant.

### 3.8.3 Identifying the query codes from query-answer pairs

The translation of the SMS text to an English form, e.g. *when do you initiate antiretroviral therapy*, is used for this illustration. A new set of SMS queries is formed and this will be used to query the search engine. Query code is essential for easy identification and recognition of each query in the database. It serves the purpose of annotation. Logging data plays a significant role in the evaluation process of a quality search service with a search engine [151] in order to merge data effectively for further data analysis. In *Table 3.6*, for interaction purposes, the SMS code and query code represent the users and the information systems [73] in research communities. For easy identification, each question with its corresponding answers has a unique code. Isolation and identification of the keywords lead to the derivation of further idioms. The interpretation of the wordings is done individually and not collectively.

It is expected that the list of keyword phrase pairs extracted from the query will be randomly or statistically selected terms from the query database and must have been stored in the *MySQL* table. For example, *Table 3.6* could be considered for the generalization of the experiment.

TABLE 3.6: Keywords extraction from FAQ data files

SMS Code	Query Code	Keyword phrase extracted from the query
$Q_1$	A	$[a_1, \dots, a_n]$ list of keywords extracted from A
$Q_2$	B	$[b_1, \dots, b_n]$ list of keywords extracted from B
$Q_3$	C	$[c_1, \dots, c_n]$ list of keywords extracted from C
...	...	...

There is an average of seven words per question sentence for the FAQ query selected. For each query in the FAQ file there are two things happening: (1) a tag or code is assigned for easy identification, and (2) a list of keyword phrases for every query sentence is created. The underlined words in *Table 3.7* denote the keywords used as references for the query. The parsing rule used for this sample database allowed that keywords may appear in more than one query sentence.

The keyword is coded by assigning *Token\_id* in *Table 3.8* by considering the set of keywords  $K_1, K_2, \dots, K_m$ , acting as the list generated using the keyword extraction algorithm from the FAQ list in *Table 3.7*. A *token\_id* is assigned as a whole number from

TABLE 3.7: SMS codes, query and keyword extraction

SMS codes	Selected keyword phrase extracted from the query
$Q_1$	When do you <u>initiate antiretroviral therapy</u> ?
$Q_2$	Can <u>HIV</u> be <u>transmitted</u> through <u>breastfeeding</u> ?
$Q_3$	Explain <u>antiretroviral treatment</u> ?
$Q_4$	What are the <u>symptoms</u> of <u>HIV infection</u> ?
$Q_5$	Does <u>breastfeeding</u> pose any risk to the <u>HIV infected</u> mother?
$Q_6$	What are <u>antiretroviral drugs</u> ?
...	...
$Q_n$	...

the FAQ query set 1, 2, ... for each keyword. *Table 3.8* illustrates a sample of keywords and their corresponding *token\_id*.

TABLE 3.8: Assigning token\_id to the keyword

Token_id	Keywords
$K_1$	initiate
$K_2$	antiretroviral
$K_3$	therapy
$K_4$	drugs
$K_5$	transmitted
$K_6$	breastfeeding
$K_7$	symptom
...	...
$K_m$	...

Corresponding to the text in *Table 3.7* is the  $n \times m$  term dependent matrix shown in *Table 3.9*. The elements of this matrix are the frequencies with which a term occurs in the FAQ file. This is used for the scoring function. The scoring function is the addition of the weighting in each query column. The results are ranked to give a list of the query-answer pair. Using SMS codes  $Q_6$  in *Table 3.7*—*What are antiretroviral drugs*—for illustration, the contents of the seventh column (see *Table 3.9*) in the term-document matrix, *antiretroviral* and *drugs*, all occur once. A value of 1 is assigned to the term if it is available, otherwise 0. The *token\_ids* of *antiretroviral* and *drugs* are  $K_2$  and  $K_4$  respectively as shown in *Table 3.8*.

The query set,  $Q$ , is represented as  $Q_1, Q_2, \dots, Q_n$ , in *Table 3.9*. The term-document matrix table is used to calculate the frequency of keyword  $K_1, K_2, \dots, K_m$ , in the query sentence  $Q$ . The corresponding values of  $K$  in  $Q$  may be Boolean values, depending on whether it is present or not in the query sentence.

TABLE 3.9: (n x m) term-document matrix corresponding to the FAQ sentences

Token.ids	$Q_1$	$Q_2$	$Q_3$	$Q_4$	$Q_5$	$Q_6$	...	$Q_n$
$K_1$	0	0	0	0	0	0	...	0
$K_2$	0	0	0	0	0	1	...	0
$K_3$	0	0	0	0	0	0	...	0
$K_4$	0	0	0	0	0	1	...	0
...	...	...	...	...	...	...	...	...
$K_m$	...	...	...	...	...	...	...	...

### 3.9 Experimental methodology on FAQ information access using SMS

The efficiency of the retrieval mechanism is determined by its performance. The best retrieval strategy may depend greatly on the length and specificity of the query, because a complex data-driven retrieval strategy may have little success with short queries and limited amounts of information [234]. Users of search engines have been accustomed to using short queries with keyword combinations due to the interface restrictions and inner mechanism of the search engine [234]. However, the detail that they provide may be vital to obtain good results for longer, more precisely defined queries where little vocabulary is shared by relevant documents, so that the system may be required to have some language understanding capability in order to discover relevant answer documents [149].

As a result, retrieval efficiency can be calculated through *precision*, *recalls* and *f-measure*. The learning performance involves performing the same set of experiments with a pre-determined number of iterations on the same dataset a particular number of times. To conduct the evaluation, the following steps are taken:

1. A sample of twenty (20) SMS coded FAQ query sentences was taken. (*Mostly they are a set of queries that have greater representation in the data collected from the respondents. This has been determined statistically*)
2. Each query was designed to retrieve the five (5) best answers. The results will be verified by experienced users, using datasets applied (Section 3.3) at the beginning of the experiment, and their corresponding answers.
3. The retrieval efficiency can be measured using *precision*, *recalls*, and *f-measure*.

**Precision (P)** is the relative amount of correct content (FAQ query) retrieved. The value must be as high as possible for good parsing. Content is considered to be correct



if it matches that in the Gold Standard

$$\text{Precision} = \frac{\text{Number of relevant FAQ queries}}{\text{Number of retrieved FAQ entries}}$$

**Recall (R)** is the relative proportion of correct constituents compared to the gold standard parse. It shows how many relevant answers were actually retrieved out of the possible answers. The higher the recall value, the better the algorithm performance.

The two metrics, *precision* and *recall*, are inversely related and are computed using an unordered list of FAQ query sets [40]. They are based on the user's relevance assessments following the retrieval process [149]. Therefore the automatic handling of the various forms of user queries not only requires a large database of QA pairs, but also the technology to match the user query to the FAQ documents in the database [134]. It is imperative to link information seekers to information sources by matching the SMS query with the description of the content that is associated with the indexed information segments in the database.

The **F-measure (F)** is a measure of a test's accuracy and it is defined as a harmonic mean of *precision(P)* and *recall(R)*:

$$F = \frac{2PR}{P + R}$$

20 questions were selected and 10 different SMS text users were asked to query the search engine. The set of sampled questions is in *Appendix A*. The user's query is matched with the FAQ repository to bring out the corresponding answer. Information retrieval efficiency will never be effective unless the SMS query is translated into the natural language in which the FAQs are structured. The FAQ dataset comprises English words and HIV/AIDS terminologies. The choice of the query is a result of the evaluation carried out on the experimental corpus of Ipoletse [109], and using the evaluation metric measurement of *precision* and *recall* on the three algorithms, *tf-idf*, *Naive* and *SMSql*.

*Table 3.10* shows the relevance judgment scale needed to calculate retrieval efficiency. The judgment is based on the first 5 FAQ sets of queries that emerge from various ways in which SMS questions are sent into the search engine. This approach is similar to Mogadala et al. [166], where *cleaned* SMS was used as a query to match the 5 best documents containing FAQ questions, using the language model approach. It is important to map the position of the SMS query to the way the FAQ questions are presented in each of the algorithms compared. The mapping will assist in determining the best retrieval efficiency of the three algorithms. A maximum of 5 points is allotted to an SMS enquiry that exactly produces the intention of the SMS texter in terms of the FAQ data set. A value of 0 point may be considered for out-of-domain situations where

the result of the FAQ query has no bearing on the SMS enquiry. Some SMS queries will be out-of-domain and will not have any corresponding FAQ answer [100, 166].

TABLE 3.10: Relevance judgment value

Relevance judgment	Value
Excellent	5.0
Very Good	4.0
Good	3.0
Moderate	2.0
Poor	1.0

### 3.10 Algorithms for information retrieval experiments

The three algorithms used to determine the information retrieval precision and computational time for the retrieval are discussed. The search engine uses the 3 different algorithms, *tf-idf*, *naive* (brute-force string match) and *SMSql*. The efficiency of the retrieval results and the computational time for each query vis-à-vis the response time and accuracy of the FAQ question-answer pair returns are used as the basis for judging the most efficient algorithm. This has been used to solve the research question posed in Section 1.4.

#### A. *Tf-idf* algorithm

As applied in Section 3.7.4, the *tf-idf* algorithm is described below:

---



---

#### The *tf-idf* algorithm

---

##### Step 1 *Document pre-processing steps*

**Tokenization**—a document is treated as a string, or bag of words, and then partitioned into a list of tokens.

Frequently occurring or insignificant words, i.e., **stop words** are eliminated.

**Stemming word**—this step is the process of conflating tokens to their root form, e.g. *correct* for *correction*, *correcting*, *corrects*, *corrected*.

##### Step 2 *Document representation*

***n*-distinct words** from the SMS and FAQ corpora are statistically selected. The collections are represented as the *n*-dimensional vector term space.

---

---



---

The *tf-idf* algorithm continued

---

Step 3 *Computing Term weights*

Get term **frequency**(**tf**).

Find **inverse document frequency**(**idf**).

Compute the **tf-idf** weighting.

Step 4 *Measure similarity between two documents* (SMS query and FAQ dataset)

Calculate the **cosine similarity** by determining the cosine of the angle between two document vectors.

---

Using the *tf-idf* algorithm the ranking of the FAQ query for the set of SMS queries given by 10 SMS users were performed. This is ranked and represents relevance of the questions based on the SMS enquiries for this approach.

### **B. Naive (Brute-force string match) algorithm**

This problem involves searching for a pattern (substring) in a string of *text*. The result is either the index in the text of the first occurrence of the pattern, or indices of all occurrences. The first one is looked for. The algorithm is described:

---



---

The *Naive (Brute-force string match)* algorithm

---

Step 1 Align the pattern at beginning of the text.

Step 2 Moving from left to right, compare each character of the pattern to the corresponding character in the text until all characters are found to match (successful search); or a mismatch is detected.

Step 3 While pattern is not found and the text is not yet exhausted, realign the pattern one position to the right and repeat.

---

### **C. SMSql algorithm—The proposed algorithm**

This section describes the *SMSql* algorithm over the SMS-based FAQ search and retrieval system for mobile accessing of information. The translated keywords extracted from the SMS query are matched with keywords present in the FAQ corpus. One of the methods adopted in arriving at a ranked list is assigning weights to the relevant terms. This shows the degree of importance of the terms (tokens) in the documents. Weight difference is needed for the following reasons: (1) to measure the degree of similarity between the FAQ terms and SMS query terms. (2) to know the length and specificity of the query sentences, and the number of relevant of FAQ terms and the SMS query terms. A weight function/value of 2 is used to confirm the FAQ query sentence length. For as many keyword terms that are available in the FAQ sentence (and non-matching) are

assigned 2. This is important if there is a tie in the weight function between FAQ terms and SMS query. The FAQ query sentence with lower sum of non-matching is considered as the chosen FAQ query sentence.

An *SMSql* algorithm is described:

---



---

The *SMSql* algorithm

---

- Step 1 A weight function/value of 1 is assigned for equal matches of the two terms in the FAQ database and the English query term, otherwise it is set to 2 for other non-matching tokens.
  - Step 2 Sum the assigned values of matches in the FAQ query.
  - Step 3 Sum the assigned values of non-matching tokens in the FAQ query.
  - Step 4 Rank the weight function/value (in Step 2) in decreasing order.
  - Step 5 In case there is a tie in Step 2, select the FAQ query sentence with lowest sum non-matching tokens.
  - Step 6 Output the five best ranked query codes.
- 

The (SMSql) algorithm considered similarity in words between the SMS query and the FAQ database, the sentence length of the two sentences, as well as the order in which the words are placed. *Tf-idf* is a product of two weightings that does not consider differences in length of the text [150]. This is taken to be an advantage of the *SMSql* algorithm, because the length of query sentence is given priority. *SMSql* processes the input sentence word by word from left to right. When the first SMS word (the target word) is found, the context window is built. This window is formed by the words placed just before and after the target word present in the FAQ database. The window size used in this system was three (3), which included the target word and one word to its left and right, following the claim by Michelizzi [162] that words farther away from the target word are less likely to be related to words close to the target word.

When an FAQ file is chosen as the query is being issued, the system iterates through the Q&A pairs in the file, comparing each question against the user's question and computes a score based on the *weight function*. The *scoring function* is defined for assigning a score to each statistically selected keyword phrase in the FAQ corpus  $Q$ , where an SMS token  $s_i$  has been normalized to the English term  $t$  in the dictionary. Therefore, there is a similarity measure  $\varphi$ , between  $s_i$  and  $t$  such that  $\varphi(s_i, t) > 0$  and this is denoted in the equation as  $s_i \approx t$ .

Consider a query term  $q$  in an FAQ dataset  $Q$  as  $q \in Q$  in the particular query sentence. For each token SMS string  $s_i$ , the scoring function chooses the term from  $q$  having the maximum weight. Then the weights of the chosen terms are summed, giving the score.

$$\text{Score}(q) = \sum_{i=1}^n \max_{t \in Q \wedge s_i \approx t} (w(s_i, t))$$

The goal is efficiently to find the best matches to the query in the FAQ. The five selections with the highest scores found are returned to the user. Each question from the FAQ file is matched against the user's question and then scored. *Table 3.11* shows a scoring function for identifiable keyword matches when the *SMSql* algorithm is applied.

TABLE 3.11: Scoring function

SMS codes	Keyword phrase extracted from the query	Score function
$Q_1$	Initiate, Antiretroviral, therapy	3
$Q_2$	HIV, Transmitted, Breastfeeding	3
$Q_3$	Blood, Transfusions, Transmit, HIV	4
$Q_4$	Opportunistic, Diseases, Treated	3
$Q_5$	Antiretroviral, Drugs	2
$Q_6$	Sexually, Transmitted, Infections	3
$Q_7$	Opportunistic, Diseases	2
$Q_8$	Body, Fluid, Transmit, HIV	4
$Q_9$	Window, Period	2
$Q_{10}$	Receive, Counselling, Phone	3

### 3.10.1 Application of scoring functions to the query selection using the three algorithms

It is assumed that if a query such as *when do you initiate antiretroviral therapy?* is parsed, the keywords: *initiate*, *antiretroviral* and *therapy* (after excluding the stop words) will be used to compare all other question forms under the FAQ file, and then the queries that are likely to be selected are:

- Explain antiretroviral treatment?
- What are antiretroviral drugs?
- Are children also eligible for ARV therapy?
- Are children and women eligible for ARV therapy?
- When do you initiate antiretroviral therapy?

The scores are calculated and ranked according to the keyword/s represented in the SMS translation as shown in *Table 3.12*:

TABLE 3.12: Questions and scores

FAQ Questions	Scores
When do you <u>initiate antiretroviral</u> therapy?	3
Are children and women eligible for <u>ARV</u> therapy?	2
Are children also eligible for <u>ARV</u> therapy?	2
Explain <u>antiretroviral</u> treatment?	1
What are <u>antiretroviral</u> drugs?	1

The answer to the query will be given according to this ranking. At this stage, it should be noted that the actual parsing of the SMS query input is done sequentially, from left to right. The first word/phrase/letter is analysed and parsed through our architecture as described in Section 3.7.1. The process here is concerned with searching, sorting and matching a similar array of word/phrase/letter. When there is a tie (i.e. equal scores), the question length will be used to break the tie, as reflected in *Table 3.12*, 2nd and 3rd questions. When it is unsuccessfully parsed, that is, the word/phrase/letter cannot be normalized by SCORE, then the SMS query translation is not extracted from the string of arrays kept in the database. The SMS query is returned as an output without successful parsing, and another token is parsed and run through the process again. But if it is successfully parsed, the parsed phrase is extracted from the database in exchange for the SMS query and it becomes a new query phrase that will replace the SMS search query and the process is repeated again.

### 3.11 Statistical analysis

The method of statistical analysis used in the two research objectives of the experiment involves descriptive statistics and inferential statistics. The data analysis was carried out through the use of a computer program called Statistical Package for Social Sciences (SPSS) [179]. Data was analysed using descriptive statistics. This provides simple summaries about the sample and the measures [179, 229] used in this study. Inferential statistics involves reaching conclusions that go beyond the immediate data by comparing the dependence of two or more factors. The majority of inferential statistics findings come from a general family of statistical models known as the General Linear Model. This includes the *t-test*, *Analysis of Variance (ANOVA)*, *Analysis of Covariance (ANCOVA)*, *regression analysis*, and many of the multivariate methods such as *factor analysis*, *multidimensional scaling*, *cluster analysis*, *discriminant function analysis*, and so on [15, 179, 229].

### **A. Statistical analysis on SMS normalization**

Statistical significance tests between methods estimate the superiority of one method over another. Significance tests are used to compare the results of different methods and decide if any one produces measurably better results than another. The most common approach to apply is the *t-test* [179]. This test compares the magnitude of difference between methods to the variation among the difference. If the average difference is large compared to its standard error level, then the methods are significantly different. This is reported in Sections 4.2.6.2–4.2.6.5.

### **B. Statistical analysis on SMS-based information access**

The repeated measure Analysis of variance (ANOVA) was used because each method (algorithm) is considered in three dimensions (precision, timing and recall). ANOVA is a collection of statistical models used to analyze the differences between group means and their associated procedures [179]. There is a continuous scale on all three methods. By using *multivariable testing* the computational execution time for the three algorithms (Table 4.17) was considered in order to confirm the level of significance of the three methods. The results are given in Section 4.5.

## **3.12 Chapter summary**

In this Chapter the research approach was presented in relation to epistemological, theoretical, and methodological perspectives, and related methods. The various methods adopted in addressing the challenges identified in the thesis were discussed and explained. The chapter discussed the data structures and methodology involved in investigating the two research questions. The algorithms to achieve SMS normalization and the information retrieval mechanism using SMS text were both described. The SMS normalization algorithm bases its performance on three important parameters: (1) text entry error, (2) similarity measurement and (3) least character distance. These methods, together with a rule-based system interpreting the order of vowel precedence, play a decisive role in making the right choice of an English translation when there were ties, i.e. candidate words. SCORE is a character-based normalization technique that uses over 40,000 English words to support the process of normalization. The hypothesis that the length of SMS words is shorter than the parent words, while the order of characters is mostly the same as in the parent form, was considered in the development of the SMS normalization algorithm. It follows that many English words with similar character combinations to the SMS can be removed from consideration. The word with the lowest WER among these variants is chosen for the translation. Various experimental techniques to confirm the robustness of the algorithms developed were also discussed. The statistical analysis

to confirm the significance test and the dependence of the variables in each algorithm in the research objective was explained.

The chapter also discussed the typical behaviour of SMS queries in a search engine, and the need to improve on the retrieval mechanism of the SMS-based system. A new SMS-based information retrieval called SMS question locator (*SMSql*) was developed. The technique of getting the score function in order to rank the question-answer pair was considered. The keyword extraction technique as a way to improve the efficiency of FAQ in the IR system was also examined. A series of experiments was performed using the three (3) algorithms, *tf-idf*, *naive* and *SMSql*, to demonstrate the retrieval efficiency. The statistical analysis used to confirm the significance test and dependence of the variables in each algorithm in the research question was explained.

In summary, this chapter has provided insight into the tools used to produce the results to be presented in the next chapter. A detailed analysis of these results will be presented in Chapter 4 in relation to the two research objectives of the thesis.



# Chapter 4

## Results

### 4.1 Introduction

This chapter discusses the results of the implementation of (1) SMS normalization, and (2) SMS-based information access, with a view to determining whether the research objectives set out in Section 1.5 have been achieved. The objectives set out were (1) to design an algorithm for translating and normalizing of SMS text, and (2) to design and develop a system for secure information-accessing using SMS. Section 4.2 describes various results from experimentation on the SMS normalization algorithm (SCORE) developed in this research. The experimental methods include *vowel stripping*, *clipping positions*, *frequency or probability model*, *evaluation of the SCORE algorithm on the dataset of Caroline and Liu*, *annotators' translation experiments* and *cross validation*. The *cross validation* method described in section 3.6.6 was used to determine whether the implementation of the SMS algorithm meets the accuracy criteria and is better than the BLEU method. The statistical analysis for the first research objective is described in Sections 4.2.6.2 – 4.2.6.5. The evaluation of the second research objective is described in Section 4.3. The performance evaluation dealing with a comparison of the computational timing of the three algorithms is handled in Section 4.4. Section 4.5 describes the statistical analysis for the second research objective. Section 4.6 concludes the chapter.

### 4.2 Experimental results for SMS normalization

The proposed algorithm involves the use of edit distance or error percentage in solving SMS normalization. An algorithm to allow single and multiple (in this case up to three) character insertion and omission, or the input of wrong characters, was designed and has been described in Section 3.5. One can query whether an algorithm can be designed to

detect and count multiple errors reliably, including all combinations and permutations of errors that could be made. What began as a simple character-by-character comparison has grown in complexity. However, it is precisely this generalisation of experimental observations that is desired in empirical studies of this nature. The following are the various results obtained while using the SCORE normalization algorithm.

#### 4.2.1 Results obtained in Experiment 1—Vowel stripping

Using the technique of *vowel stripping*, the results obtained in Section 3.6.1 are grouped in three categories. The first category is words that have their vowels stripped out and the string generated is then submitted to the algorithmic process. The algorithm first strips the vowels out so that, for instance, *medicine* becomes *mdcn*, which is then processed by the SCORE algorithm. The result can be either a Success or a Failure. Successes are counted if the result is the same as the word that was not vowel-stripped i.e. the initial lexical item. A failure results when the word differs from the original input. For instance, stripping the vowel off *abt* may result in *abate*, *abet*, *about*, *abut*. The SCORE algorithm selects the word *about* because it has the least character distance (LCD). The other words will be *Failures*.

The results of the normalization obtained establish the robustness of the SCORE algorithm, which outperforms some of the existing methods even with a higher rate of unknown words, or a lower BLEU [181] score in raw text. The performance of the SCORE algorithm is represented in *Table 4.1*, where the success rate is calculated by the number of *vowel-stripped words* that return exactly the form in which they were before the *vowel stripping algorithm* was applied. The failure rate is the opposite of this action. *Passive* represents English words that do not have vowels (e.g. rhythm, hymn) but were part of the datasets.

TABLE 4.1: Results using vowel stripping method

SMS Query	Success	Failure	Passive
SCORE	82	13	5
BLEU	32	59	9

The results obtained from the BLEU method is the average result of 5 annotators that attempted to reverse the *vowel-stripped* word into its original form. The *success*, *failure* and *passive* results were calculated the same way as with the SCORE results. The observation was that there were many candidate words (that is, words that allow several forms of interpretation) obtained by the annotator, and this increased the failure rate.

The interpretation appeared to be accurate but the returned forms were mostly *false successes* or *false positives* because they were not the words intended for translation. The passive was 100% better than the SCORE outcome, as the annotators were able to identify words having no vowel; hence there was no need for translation.

#### 4.2.2 Results obtained in Experiment 2—Clipping positions

Using the technique of *clipping positions* in Section 3.6.2, the results captured in *Figure 4.1*, suggested that *initial clipping* suffers the least and that *vowel clipping*, *medial clipping*, *mixed clipping* and *end stripping*, do not differ significantly from each other. Normalization may not be achieved if the SMS word has its initial letter stripped off.

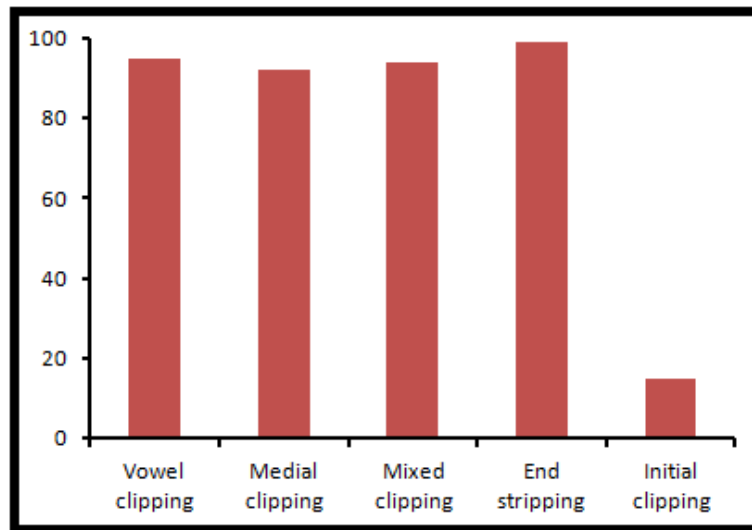


FIGURE 4.1: Normalization performances on 100 data sets using different clipping position

#### 4.2.3 Results obtained in Experiment 3—Frequency or probability model

Using the technique of *frequency* or the *probability model* discussed in Section 3.6.3, the results show in *Figure 4.2* that the highest frequencies, namely queries, Q2 and Q4, resulted from SMS messages that were either acronyms or abbreviations. The lowest frequency queries, Q1 and Q10, resulted from mixed clipping. The other categories are caused by vowel and character stripping from different positions of the word.

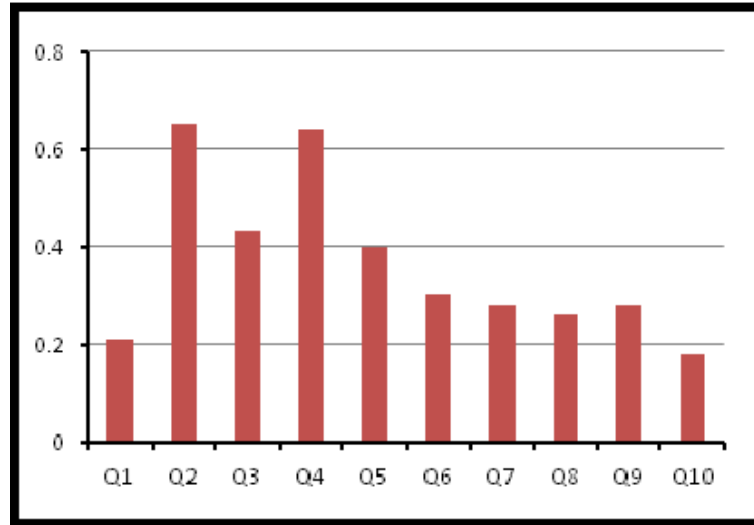


FIGURE 4.2: Relative frequency analysis of the 10 queries used for the experimentation

#### 4.2.4 Results obtained in Experiment 4—Evaluation of two set corpora using SCORE algorithm

Using the technique of *evaluation of the two set corpora of Liu and Tagg* in Section 3.6.4, *Table 4.2* shows an overall improvement in the normalized SMS text, based on the number of words in the corpus, the average number of words per message length, the average number of characters per message length and the average number of characters per word.

TABLE 4.2: The results of normalized SMS from the Tagg (2009) corpus after application of the SCORE algorithm

Features	Original text	Normalized SMS
No of messages in the corpus	11,036	11,036
No of words in the corpus	19,099	198,500
No of SMS words in the corpus	4,012	3,859
Average no of words per message	15.28	17.2
Average no of characters per message	18.24	24.5
Average character per word	4.65	5.7

While the number of messages in the corpus remains the same, there is a significant 5% increase in the number of words added to the corpus after the SCORE algorithm has translated some of the SMS words available in the corpus. This goes against the general rule observed from the number of tokens collected in English corpus research [188] because there are many more tokens in the original text than in the SMS. In trying to isolate SMS words used in the corpus, the SCORE algorithm succeeded in normalizing 86.18% of the SMS text. 13.82% of the text messages—alphanumeric, homophones and emoticons—were mostly outside of the scope of the dictionary used in the development

and implementation of the algorithm. It should be clear that the objective of this particular evaluation is not to confirm whether the translation is the right one or not, but to assess whether a *reasonable* English translation of the SMS has been achieved.

There are improvements of 13%, 34% and 25% in the average numbers of words per message, characters per message and characters per word respectively. A simplified setting of alignment (as a list of pairs), i.e. monotonic, between the source and target languages during the training and testing of the dataset was considered.

With the same number of messages in the corpus, *Table 4.3* shows that there is a significant 2.5% increase in the number of words added to the corpus after the SCORE algorithm has translated some of the SMS words available in the corpus.

TABLE 4.3: The results of normalized SMS from the Liu (2010) corpus after application of SCORE algorithm

Features	Original text	Normalized SMS
No of messages	20,036	20,036
No of words	85,866	87,012
Characters (no spaces)	216,968	245,325
Characters (with spaces)	301,837	312,587
No of SMS words in the corpus	14,012	12,011
Average no of word per message	3.5	5.6
Average no of characters per message	8.2	9.5
Average characters per word	1.8	3.1

In trying to isolate the SMS words that were used in the corpus, the SCORE algorithm gave results of 81% in the normalization process, with difficulties coming up in those areas that were outside the scope of the SCORE algorithm. There are improvements of 60%, 16% and 72% in the average numbers of words per message, characters per message and characters per word, respectively, in Liu’s corpus.

#### 4.2.5 Results obtained in Experiment 5—Annotator translations

The experiments are performed using three methods, as discussed in Section 3.6.5.

##### 4.2.5.1 Experiment 5a: Result obtained from English → SMS

Using the technique of *annotator translation* in Section 3.6.5, the percentage success rate for identifying English terms equivalent to their SMS counterparts in the data set is given in *Table 4.4*.

The interpretation shows that an 84% success rate was achieved by the annotators.

TABLE 4.4: Annotator results obtained from English  $\rightarrow$  SMS

Success	False success	Failure
84%	11%	5%

#### 4.2.5.2 Experiment 5b: Result obtained from SMS $\rightarrow$ English

Using the technique of *annotator translation* in Section 3.6.5, the results show the percentage of SMS terms identified as being equivalent to their English terms. The average results are shown in Table 4.5. A 24% success rate was achieved by the annotators.

TABLE 4.5: Annotator results obtained from SMS  $\rightarrow$  English

Success	False success	Failure
24%	61%	15%

In summary, the results in *Figure 4.3* demonstrate that *success* and *false success* are inversely proportional to each other when an operation of forward and backward selections is performed on a set of translation from English to SMS, and vice versa.

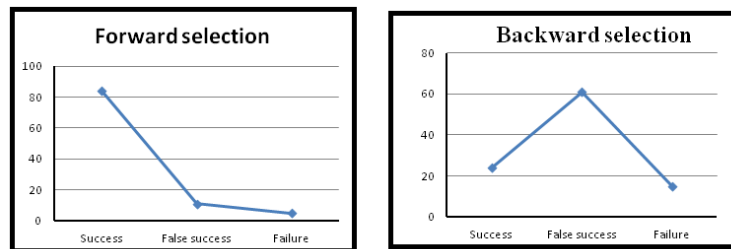


FIGURE 4.3: Annotators' forward and backward selection

#### 4.2.5.3 Experiment 5c: Annotator with/without prior knowledge

Using the technique of *annotator translation* in Section 3.6.5, the annotators had the twin privileges of (1) no knowledge of the datasets before the translation was done, and (2) knowledge of the datasets to be interpreted before they were asked to translate from SMS to English. This is an approach used by Gouws et al. [94]. The result, presented in *Table 4.6*, was poor.

The result shows the difficulty experienced in human translation of SMS words. Pre-knowledge of the SMS words gave an appreciable overall improvement of 3 times.

TABLE 4.6: Results for the annotators

Annotator#	Agreement without pre-knowledge (%)	Agreement with pre-knowledge (%)
1	28	74
2	32	87
3	17	69
Average	25.67	76.67

#### 4.2.6 Results obtained in Experiment 6—Cross validations

Using the technique of *cross validation*, in Section 3.6.6, the *N-best* approach was used as the measure of precision. There were areas of agreement and disagreement, especially with the annotators, but the average values were taken. Disagreement may arise over whether or not there is a difference in the translation of the query terms. A comparison was made between the results from the two methods, BLEU and SCORE, based on the efficiency and precision of outcomes using *N-best*. Both techniques were tested using the same experimental conditions. A measuring technique was adopted called *Mean Average Precision*, which is a single-value metric that serves as an overall figure for directly comparing different retrieval results. It is the total average of the outcome of retrieval for every document that is being considered in the experiment, where the mean of all these averages is calculated across all the test queries [157, 240].

$$\text{Mean average precision (MAP)} = \frac{1}{20} \sum_{n=1}^{20} ave_n$$

where *ave* is the average for each result, and *n*, the total number of query consider for the experiment.

Tables 4.7 and 4.8, Figures 4.4 and 4.5, show the average and mean average precision respectively for the 20 sample queries tested by 10 annotators using BLEU and SCORE techniques. The results are derived from the dataset sample Table 3.3. The maximum *N-best* result on the scale is 1.0.

In Table 4.7 the first and the last columns represent the randomly selected SMS queries used in the experiment and the average of all the annotators' scores per SMS query respectively. The other columns—A1 to A10—are the scores given by each individual annotator for each SMS query. The experiment was performed on 20 sets of SMS queries. This is represented in the first column, SMS Queries 1 to 20.

TABLE 4.7: Precision results for BLEU method

SMS Query	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Avg/ query
1	0.7	0.4	0.9	0.5	0.7	0.9	0.6	0.9	0.8	0.7	0.71
2	0.8	0.8	0.8	0.4	0.6	0.8	0.8	0.4	0.8	0.8	0.70
3	0.6	0.7	0.8	0.6	0.8	0.7	0.7	0.3	0.7	0.7	0.66
4	0.4	0.9	0.7	0.4	0.4	0.9	0.6	0.7	0.5	0.5	0.60
5	0.5	0.5	0.6	0.5	0.5	0.6	0.8	1.0	0.7	0.6	0.63
6	0.4	0.5	0.6	0.6	0.4	0.7	0.7	0.7	0.8	0.7	0.61
7	0.5	0.8	0.8	0.3	0.5	0.8	0.8	0.8	0.6	0.6	0.65
8	0.7	0.4	0.7	0.5	0.6	0.6	0.7	0.6	0.7	0.6	0.61
9	0.9	0.5	0.7	0.6	0.5	0.3	0.6	0.9	0.6	0.3	0.59
10	0.4	0.6	0.6	0.6	0.5	0.5	0.7	0.8	0.8	0.5	0.60
11	0.3	0.8	0.4	0.8	0.3	0.6	0.9	0.8	0.6	0.6	0.61
12	0.7	0.7	0.5	0.4	0.7	0.8	0.8	0.8	0.4	0.8	0.66
13	1.0	0.4	0.6	0.7	1.0	0.9	0.7	0.5	0.5	0.7	0.70
14	0.8	0.8	0.8	0.6	0.8	0.4	0.5	0.6	0.6	0.4	0.63
15	0.7	0.6	0.7	0.5	0.6	0.7	0.6	0.7	0.6	0.7	0.64
16	0.7	0.4	0.6	0.7	0.7	0.8	0.7	0.4	0.7	0.8	0.65
17	0.6	0.5	0.5	0.7	0.6	0.6	0.8	0.8	0.6	0.6	0.63
18	0.4	0.4	0.4	0.6	0.4	0.9	0.6	0.7	0.4	0.7	0.55
19	0.5	0.3	0.7	0.4	0.5	0.8	0.3	0.9	0.8	0.8	0.60
20	0.6	0.7	0.8	0.5	0.6	0.8	0.5	0.8	0.7	0.6	0.66
Total	12.2	11.7	13.2	10.9	11.7	14.1	13.4	14.1	12.9	12.7	–
Avg	0.61	0.59	0.66	0.55	0.59	0.71	0.67	0.71	0.65	0.64	–

$$MAP = \frac{0.61 + 0.59 + 0.66 + 0.55 + 0.59 + 0.71 + 0.67 + 0.71 + 0.65 + 0.64}{10} = 0.64$$

Similarly, *Table 4.8* represents the results of using SCORE to determine the precision. The SMS queries are the same as those used with BLEU in *Table 4.7*. There are twelve columns, the first and the last columns representing the randomly selected SMS query set used in the experiment, and the average score for all the annotators' scores per SMS query, respectively. The other columns—A1 to A10—are the scores of individual annotator for each SMS query set. The experiment was performed on twenty sets of SMS queries. This is represented in the first column, SMS Queries 1 to 20.

$$MAP = \frac{0.82 + 0.82 + 0.81 + 0.88 + 0.87 + 0.82 + 0.77 + 0.76 + 0.86 + 0.84}{10} = 0.83$$

$$\text{Percentage difference in MAP for the two methods} = \frac{0.83 - 0.64}{0.83} \times 100\% = 22.89\%$$



TABLE 4.8: Precision results for SCORE method

SMS Query	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Avg/ query
1	0.9	0.8	0.6	0.9	0.8	0.9	0.6	0.7	0.9	0.9	0.80
2	0.8	1.0	1.0	0.8	0.8	0.8	0.9	0.8	0.8	0.8	0.85
3	0.6	0.7	0.6	1.0	0.9	0.7	0.7	0.6	0.7	1.0	0.75
4	0.6	0.9	0.8	0.6	1.0	1.0	0.6	0.8	1.0	0.8	0.81
5	1.0	0.8	1.0	1.0	0.8	0.8	0.8	1.0	0.8	0.6	0.86
6	0.8	1.0	0.9	0.8	0.8	0.7	0.7	0.6	1.0	0.9	0.82
7	0.8	0.8	0.8	1.0	0.7	0.8	0.8	0.9	0.8	0.9	0.83
8	0.9	0.6	0.6	0.9	0.8	0.9	1.0	0.8	0.9	0.8	0.82
9	0.9	1.0	0.8	1.0	0.9	0.8	0.9	0.8	0.8	0.7	0.86
10	0.8	0.6	1.0	0.8	1.0	0.8	0.7	0.6	0.8	1.0	0.81
11	0.8	0.8	0.7	0.8	0.9	1.0	0.9	0.9	1.0	0.8	0.86
12	0.9	1.0	0.9	0.9	0.8	0.8	1.0	0.9	1.0	0.8	0.87
13	1.0	0.9	0.8	1.0	1.0	0.9	0.7	0.6	0.9	0.8	0.86
14	0.8	0.8	1.0	0.8	0.9	0.8	0.5	0.9	0.8	0.9	0.82
15	0.7	0.6	0.8	0.7	1.0	0.7	0.6	0.7	1.0	0.8	0.76
16	0.9	1.0	0.7	1.0	0.6	0.8	0.7	0.6	0.8	0.8	0.79
17	0.8	0.5	0.7	0.8	1.0	0.6	0.8	0.8	0.6	0.9	0.75
18	0.8	0.7	0.8	0.8	0.9	0.9	0.6	0.7	0.9	1.0	0.81
19	0.6	0.8	0.8	1.0	1.0	0.8	0.8	0.8	0.8	0.6	0.80
20	0.9	1.0	0.8	0.9	0.8	0.8	1.0	1.0	1.0	0.8	0.90
Total	16.3	16.3	16.1	17.5	17.4	16.3	15.3	15.2	17.1	16.8	–
Avg	0.82	0.82	0.81	0.88	0.87	0.82	0.77	0.76	0.86	0.84	–

*Figure 4.4* illustrates the comparison between the average results of BLEU and SCORE. It is confirmed that the average precision of SCORE is higher than BLEU by 23%. It is also observed that, unlike in BLEU, there are sudden surges in SCORE, especially where the query terms are in the medical domain. The SCORE system has been developed using a domain of medical terms.

The average precision in translating each query using the two methods is shown in *Table 4.9*. This table is a combination of the data in the twelfth columns of the *Tables 4.7* and *4.8*. It is the average precision of the two methods. When compare with BLEU, it is clear that SCORE attains higher relevant scores for every query.

#### 4.2.6.1 Experimental results

Similar test results for SMS queries are shown in Appendix D. The average precision for each of the 10 annotators is represented in Appendix D accordingly. It was observed that BLEU shows better results in some cases because the pool of English terms was not sufficient to cover the SMS query. The average precision of annotators for the BLEU and SCORE algorithms are shown in *Table 4.10*. The values are the average scores achieved by annotators using both methods for translating the SMS enquiries.

*Figure 4.4* is the graphical representation of *Table 4.10*. This shows the average precision of BLEU and SCORE algorithms for the 10 annotators.

TABLE 4.9: Average precision of BLEU and SCORE algorithms for each query sentence conducted for the experiment

SMS Query	BLEU score	SCORE algorithm
$Q_1$	0.71	0.80
$Q_2$	0.70	0.85
$Q_3$	0.66	0.75
$Q_4$	0.60	0.81
$Q_5$	0.63	0.86
$Q_6$	0.61	0.82
$Q_7$	0.65	0.83
$Q_8$	0.61	0.82
$Q_9$	0.59	0.86
$Q_{10}$	0.60	0.81
$Q_{11}$	0.61	0.86
$Q_{12}$	0.66	0.87
$Q_{13}$	0.70	0.86
$Q_{14}$	0.63	0.82
$Q_{15}$	0.64	0.76
$Q_{16}$	0.65	0.79
$Q_{17}$	0.63	0.75
$Q_{18}$	0.55	0.81
$Q_{19}$	0.60	0.80
$Q_{20}$	0.66	0.90

TABLE 4.10: Average precision of BLEU and SCORE algorithms for the annotators

10-fold cross validation	BLEU score	SCORE algorithm
$A_1$	0.61	0.82
$A_2$	0.59	0.82
$A_3$	0.66	0.81
$A_4$	0.55	0.88
$A_5$	0.59	0.87
$A_6$	0.71	0.82
$A_7$	0.67	0.77
$A_8$	0.71	0.76
$A_9$	0.65	0.86
$A_{10}$	0.64	0.84

The graphical representation of *Table 4.9* depicts the average precision of BLEU and SCORE algorithms for each query term conducted in the experiment as presented in *Figure 4.5*.

It is apparent that in general SCORE performs better than BLEU. A statistical analysis to determine the significance of the difference between the two methods is described in Sections 4.2.6.2 – 4.2.6.5.

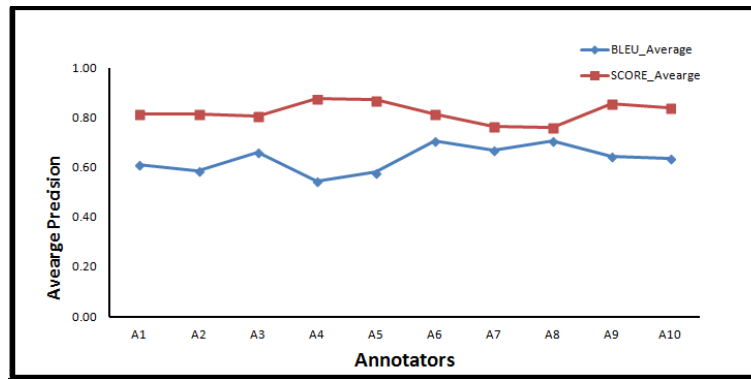


FIGURE 4.4: Average precision of all the annotators

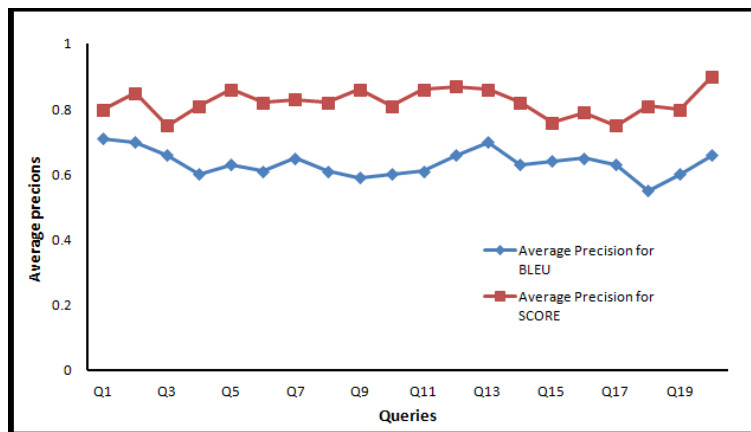


FIGURE 4.5: Average precision of all the queries

The final results, after the normalization process undertaken by the two methods, are reflected in *Table 4.11*. The SMS population was counted before and after the commencement of the experiment. The percentage average precision, by the annotators, using the two methods was also taken.

#### 4.2.6.2 Statistical analysis

Two statistical analyses were carried out, namely significance tests and correlations. The significance test measures the effectiveness between two methods, SCORE and BLEU. Another procedure was performed to determine the correlation between the corrected SMS translated words in the query collections and the resultant *n-best* result, using the two methods.

#### 4.2.6.3 Significance test

The aim of the test is to determine any improved performance using one method rather than the other. A significance test was adopted to reject the null hypothesis,  $H_0$ , that

TABLE 4.11: Summary of the SMS in each bin at the end of the normalization

Bin#	SMS Available (r)	Normalized by BLEU (s)	Normalized by SCORE (t)	% Average precision for BLEU normalization ( $\frac{s}{r} \times 100$ )	% Average precision for SCORE normalization ( $\frac{t}{r} \times 100$ )
1	148	98	118	66	80
2	79	60	72	76	91
3	108	85	94	79	87
4	98	67	86	68	88
5	124	80	110	65	89
6	103	84	90	82	87
7	138	102	114	74	83
8	112	80	108	71	96
9	92	78	85	85	92
10	107	67	97	63	91
Total	1109	801	974	72	88

there is no difference between the SCORE and BLEU methods. The idea is to show that, on the basis of results, the null hypothesis is indefensible, because it is associated with an implausibly low probability. Rejecting  $H_0$ , implies accepting the alternative hypothesis,  $H_1$ , that SCORE consistently outperforms method BLEU:

$H_0$  : average precision SCORE – average precision BLEU  $\leq 0$

$H_1$  : average precision SCORE  $>$  average precision BLEU

The hypothesis was tested by comparing average precision values across SMS queries in the two methods.

#### 4.2.6.4 T-tests

Two types of *t-tests* were used, (1) the *independent-samples t-test*, used when there is a need for a mean score comparison between two *different* groups or methods; and (2) the *paired-samples t-test*, used to compare the mean scores for the *same* group and the same condition or method on two different occasions, or when there are matched pairs [179]. In both cases one is comparing the values of some continuous variable for *two* groups, or on *two* occasions. A *paired-samples t-test* will tell if there is a statistically significant difference in the mean scores using the two methods.

The *paired-samples t-test* is applied

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$$

To test the null hypothesis, one needs to calculate the following values:  $\bar{x}_1$  and  $\bar{x}_2$ , are the means of the two samples;  $s_1^2$  and  $s_2^2$  are the variances of the two samples,  $n_1$  and  $n_2$  are the sample sizes of the two samples.

In *Table 4.11* the average precision for the two methods is presented, and the *t-test* will be used to confirm if there is significant difference in the two methods. The two variables, *categorical independent* and *continuous dependent variables*, are tests needed to confirm whether they are nominally distributed.

The results in *Table 4.12*, using Shapiro-Wilk, shows that they are nominally dependent, indeed that  $P_{value}$  of the BLEU method is ( $P_v = 0.699 > 0.05$ ) while for SCORE method is ( $P_v = 0.904 > 0.05$ ).

TABLE 4.12: Test of normality

	Shapiro-Wilk		
	Statistic	df	P_value
BLEU	0.953	10	0.699
SCORE	0.971	10	0.904

The boxplots in *Figure 4.6* also show that the average precisions of BLEU and SCORE methods are normally distributed.

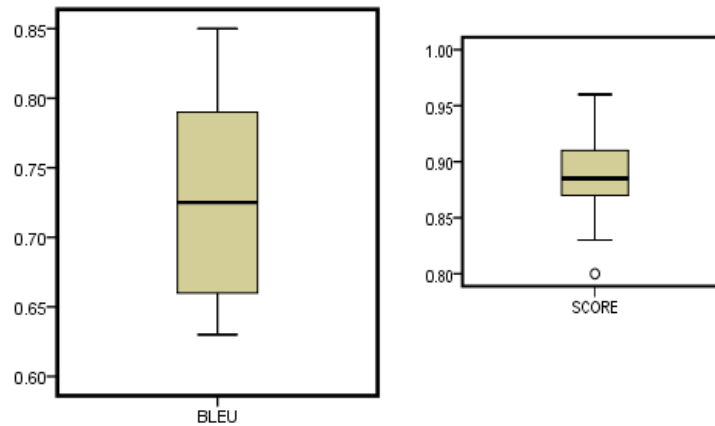


FIGURE 4.6: Boxplots

The  $P_{value}$  related to the paired *t-test* is less than 0.05 ( $P_{value} < 0.05$ ), which means that there is a statistically significant difference of mean precision between the BLEU method and the SCORE method. In *Table 4.13*, the mean precision of the SCORE method (mean = 0.8840, std= 0.04575) is higher than the mean precision of the BLEU method (mean = 0.7290, std=0.07549).

TABLE 4.13: Results of paired samples t-test

		N	Mean	Std. Deviation	Std. Error Mean
Average_Precision	BLEU	10	.7290	.07549	.02387
	SCORE	10	.8840	.04575	.01447
	Difference	10	-1.5500	.08317	.02630
	BLEU - SCORE				
95% Confidence Interval of the Difference : (-.21449, -.09551)					
T-value = -5.894, P-value = 0.00					

#### 4.2.6.5 Correlations

There is no linear correlation from the graph in *Figure 4.7*, and therefore the analysis could proceed to run a Pearson correlation coefficient test.

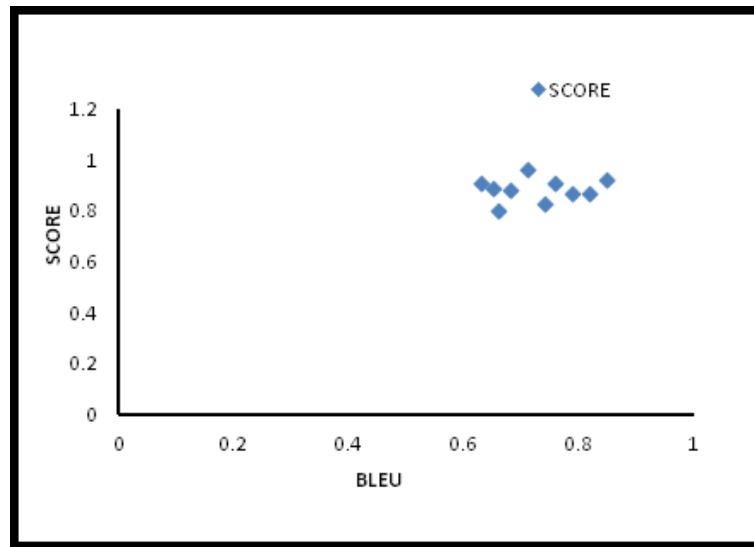


FIGURE 4.7: Scattered plot of SCORE vs BLEU

The mean precision of BLEU and SCORE are both normally distributed, as shown above; therefore the Pearson's Correlation Coefficient is applied. The  $P_{value}$  related to the Pearson correlation coefficient is greater than 0.05, therefore there is no statistically significant correlation between the average precision of the BLEU method and that of SCORE method; Pearson Correlation ( $N=10$ ,  $r=.127$ ,  $P_{value} = 0.727$ ). The value of the coefficient is also very small, confirming the absence of linear correlation.

### 4.3 Experimental results on information access using SMS

This section describes the results obtained in terms of *precision*, *recall* and *f-measure*, when three algorithms, *tf-idf*, *naive* and *SMSql*, are used to obtain results for various

queries used in the search engine, in order to ascertain the efficiency of the developed *SMSql* algorithm, described in Section 3.10C.

### 4.3.1 Results of *tf-idf* algorithm on information access using SMS

Table 4.14 presents the results in the form of *average precision*, *average recall* and *f-measure*. *Average precision* and *recall* are the *precision* and *recall* values obtained from, respectively, the set of top  $k$  ( $k$  is the size of the FAQ query document) existing in FAQ datasets after each relevant FAQ query is retrieved, and this value is then averaged over information needs. That is, the set of relevant FAQ documents required to satisfy a query is  $q_j \in Q$  is  $d_1, \dots, d_{m_j}$  and  $R_{jk}$  is the set of ranked retrieval results from the top results until the FAQ query document  $d_k$  is achieved. The *tf-idf* retrieval approach combines the frequency count of the word and the weight of each word in the document [213]. The responses to query (documents) are returned in a decreasing order of significance. At the top of the list is the highest sum of weight for the query. For instance, a query containing a higher weight  $w$  would be likely to receive an FAQ query  $q$  as a return value.

TABLE 4.14: Results of tf-idf algorithm

SMS (FAQ) Query	Average Precision	Average Recall	F-Measure
$Q_1$	2.92	1.86	2.27
$Q_2$	3.13	1.89	2.36
$Q_3$	4.94	0.89	1.51
$Q_4$	2.04	2.47	2.23
$Q_5$	3.87	0.86	1.41
$Q_6$	1.83	2.47	2.10
$Q_7$	1.12	2.80	1.60
$Q_8$	2.22	2.47	2.34
$Q_9$	3.80	1.86	2.50
$Q_{10}$	2.03	2.47	2.23
$Q_{11}$	2.13	2.45	2.28
$Q_{12}$	2.23	2.45	2.33
$Q_{13}$	2.81	1.85	2.23
$Q_{14}$	0.82	2.50	1.23
$Q_{15}$	2.90	1.84	2.25
$Q_{16}$	4.91	0.89	1.51
$Q_{17}$	4.91	0.87	1.48
$Q_{18}$	3.92	0.89	1.45
$Q_{19}$	3.23	1.86	2.36
$Q_{20}$	2.12	2.45	2.27
Total	57.88	38.09	39.946
Average	2.894	1.905	1.997

### 4.3.2 Results of the *naive* algorithm on information access using SMS

*Naive* retrieval is done by brute force whereby the list of queries is traversed to count the frequency of occurrences of a particular word [194]. The fault of this approach is that non-relevant documents appeared most often. The peak of the graph is where the most relevant query is retrieved. But before the peak the results of the query produced many irrelevant selections. The results are presented in *Table 4.15*.

TABLE 4.15: Results of naive algorithm

SMS (FAQ) Query	Average Precision	Average Recall	F-Measure
$Q_1$	2.76	3.83	3.21
$Q_2$	3.71	4.21	3.94
$Q_3$	3.92	4.04	3.98
$Q_4$	2.93	3.72	3.28
$Q_5$	4.33	4.24	4.28
$Q_6$	3.83	3.59	3.71
$Q_7$	3.24	3.84	3.51
$Q_8$	2.98	3.18	3.08
$Q_9$	3.83	2.79	3.23
$Q_{10}$	2.93	3.22	3.07
$Q_{11}$	2.94	2.83	2.88
$Q_{12}$	2.81	2.81	2.81
$Q_{13}$	3.54	2.90	3.19
$Q_{14}$	2.64	4.21	3.25
$Q_{15}$	2.98	2.80	2.89
$Q_{16}$	4.83	1.22	1.95
$Q_{17}$	4.91	1.63	2.45
$Q_{18}$	3.71	2.72	3.14
$Q_{19}$	2.82	2.61	2.71
$Q_{20}$	4.85	2.80	3.55
Total	70.49	63.19	64.097
Average	3.524	3.159	3.204

### 4.3.3 Results of *SMSql* algorithm on information access using SMS

There are two possible feature representations in *SMSql* results, *True* or *False*, indicating whether a particular feature exists in the answer or not. *SMSql* uses binary feature representation as it was found to produce the best generalization accuracy for information retrieval. *SMSql* represents a keyword  $e_i$  in the query as a vector of feature values, i.e.  $(e_i = f_1 f_2 \dots f_n, s)$  where  $f$  is a keywords in the FAQ files and  $s$  is the query sentence. Binary feature representation for similarity in the keyword features uses the existing algorithm, i.e. if the feature exists in the case  $f_1 = 1$  otherwise  $f_1 = 0$ . By summing all the values of  $f_1$ , that is,  $\Sigma_{f_1}$ , then the highest value stands as the selected or highest ranked query from the FAQ data files. All other values may fall into the category of False Positive (FP). FP are queries that were selected as being correct but



are not. To reduce the rate of FPs the  $k$ -NN algorithm is used:  $k$ -nearest neighbours have to consider the number and closeness of the keyword that matches the query and the data sets in the FAQ files. The query sentence length is considered as well. The results are presented in *Table 4.16*

TABLE 4.16: Results of SMSql algorithm

SMS (FAQ) Query	Average Precision	Average Recall	F-Measure
$Q_1$	2.79	1.95	2.30
$Q_2$	2.24	3.83	2.83
$Q_3$	4.93	1.52	2.32
$Q_4$	3.23	2.44	2.78
$Q_5$	3.84	2.92	3.32
$Q_6$	3.93	2.71	3.21
$Q_7$	2.29	3.64	2.81
$Q_8$	2.93	2.52	2.71
$Q_9$	4.89	1.41	2.19
$Q_{10}$	2.89	2.61	2.74
$Q_{11}$	3.94	2.82	3.29
$Q_{12}$	2.92	2.53	2.71
$Q_{13}$	2.78	3.27	3.01
$Q_{14}$	2.14	2.81	2.43
$Q_{15}$	3.92	2.64	3.16
$Q_{16}$	4.93	2.53	3.34
$Q_{17}$	4.92	2.85	3.61
$Q_{18}$	4.09	2.66	3.22
$Q_{19}$	3.13	2.52	2.79
$Q_{20}$	4.17	2.63	3.23
Total	70.90	52.81	57.986
Average	3.545	2.641	2.899

#### 4.4 Performance evaluation

This section is devoted to testing and evaluation of the developed system. The evaluation is carried out by computing the time taken for the retrieval of documents in the FAQ system. *Figures 4.8* and *4.9* depict the average precision and average recalls of the three (3) algorithms. The results are generated from the average precision and average recall results of *Tables 4.14*, *4.15* and *4.16*.

These metrics (average precision and recall) may not be sufficient to prove the algorithms. The system is further tested by employing a timing computation of the retrieval system. The comparison of the three results is confirmed by the execution time. The result is shown in *Table 4.17*. The three algorithms are compared in terms of computational speed. This is similar to the work described by Pudil et al. [189] using simple feature selection. The methods of Pudil et al. [189] show similar performance and differ only in computational efficiency. The objective of the comparison is directed

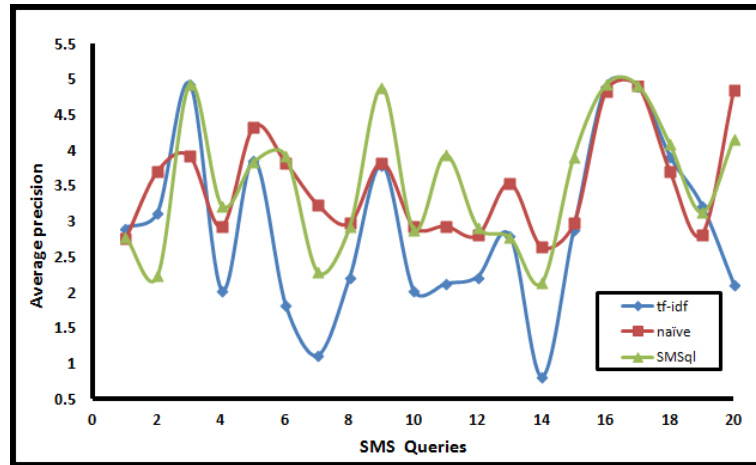


FIGURE 4.8: Average precision of the three algorithms

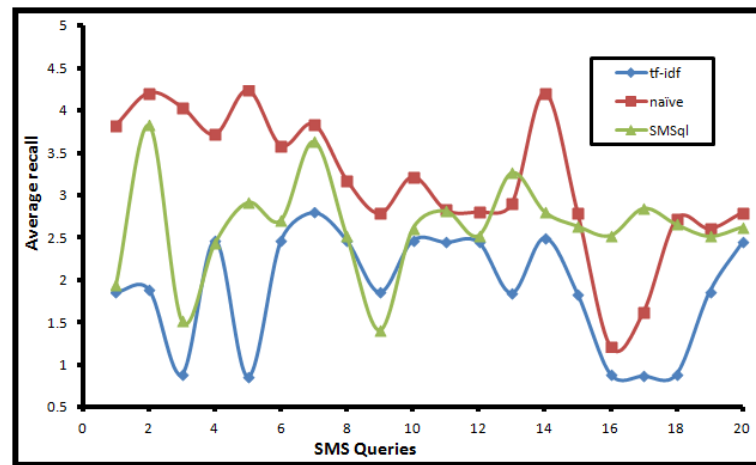


FIGURE 4.9: Average recall of the three algorithms

towards identifying sub-optimal search methods. This can be achieved by considering computational time and efficiency.

Percentage of improvement between *tf-idf* and *SMSql*

$$= \frac{0.073 - 0.070}{0.073} \times 100\% = 4.1\%$$

Percentage of improvement between *naive* and *SMSql*

$$= \frac{0.078 - 0.070}{0.078} \times 100\% = 10.3\%$$

The results show 4% and 10% improvement in the computational retrieval efficiency measured by computational speed between *SMSql* and other algorithms *tf-idf* and *naive* respectively.

TABLE 4.17: Time computation for the retrieval process of the SMS queries

Average time for each iteration of the SMS query	<i>Tf-idf</i>	<i>Naive</i>	<i>SMSql</i>
$Q_1 \rightarrow t_1$	0.085	0.099	0.097
$Q_2 \rightarrow t_2$	0.077	0.082	0.087
$Q_3 \rightarrow t_3$	0.086	0.076	0.076
$Q_4 \rightarrow t_4$	0.074	0.068	0.072
$Q_5 \rightarrow t_5$	0.085	0.078	0.085
$Q_6 \rightarrow t_6$	0.037	0.065	0.037
$Q_7 \rightarrow t_7$	0.069	0.075	0.069
$Q_8 \rightarrow t_8$	0.067	0.072	0.067
$Q_9 \rightarrow t_9$	0.077	0.080	0.072
$Q_{10} \rightarrow t_{10}$	0.068	0.072	0.068
$Q_{11} \rightarrow t_{11}$	0.074	0.074	0.074
$Q_{12} \rightarrow t_{12}$	0.088	0.090	0.078
$Q_{13} \rightarrow t_{13}$	0.067	0.077	0.057
$Q_{14} \rightarrow t_{14}$	0.075	0.079	0.075
$Q_{15} \rightarrow t_{15}$	0.082	0.089	0.062
$Q_{16} \rightarrow t_{16}$	0.067	0.077	0.057
$Q_{17} \rightarrow t_{17}$	0.078	0.082	0.078
$Q_{18} \rightarrow t_{18}$	0.068	0.078	0.062
$Q_{19} \rightarrow t_{19}$	0.059	0.069	0.059
$Q_{20} \rightarrow t_{20}$	0.074	0.080	0.064
Total	1.457	1.562	1.396
Average	0.073	0.078	0.070

In order to demonstrate clearly the effectiveness of each method, the selection of a feature set from data showing high statistical dependencies provides a more discriminating test [189]. The execution time to generate results was compared for the three algorithms. The system of calculating execution time can be constructed out of sequential programs but are typically built from concurrent programs called tasks [190].

From *Table 4.17*, the average time taken  $t_n$  for a query  $Q_n$  for each SMS request by the user is taken for each of the algorithms, and the results are presented. The score of each query sentence is calculated sequentially and then ordered to generate the result. The average time for each iteration of the SMS queries,  $Q_1 - Q_{20}$ , for each algorithm was taken. The average of the Time/sec was plotted against each algorithm and is presented in *Figure 4.10*. The results show the time spent in generating responses to requests made in this experiment. There is a 10.3% improvement when the computational speed of *SMSql* was compared with the *naive* algorithm, and 4.1% when compared with the *tf-idf* algorithm. *SMSql* was the fastest. Further analysis of the graph is carried out in Section 4.5.

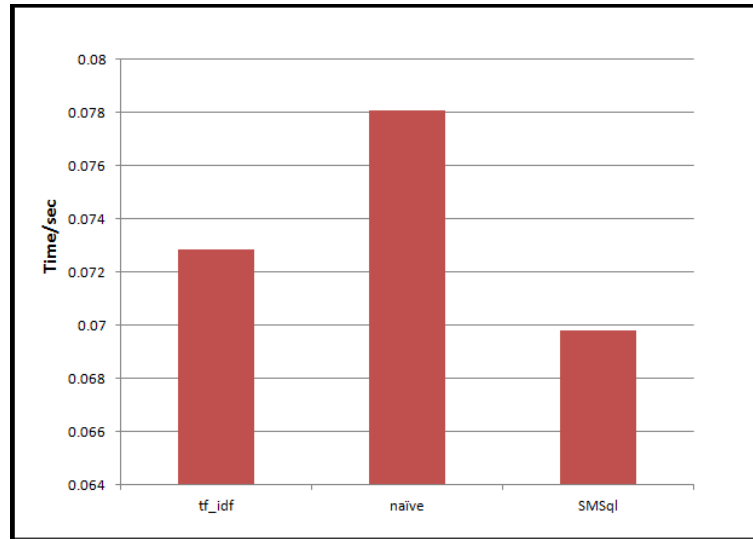


FIGURE 4.10: Comparison of the execution time of the three algorithms

## 4.5 Statistical analysis

The *one way* repeated measure ANOVA was used because each method (algorithm) is exposed to three conditions (*precision*, *recall* and *timing*) [179]. There is continuous scaling on the three methods.

Table 4.18 shows the descriptive analysis of the three methods using *average precision statistical analysis*, where 20 query samples (N) were used. The results of the *SMSql* gave the best Mean (3.55) and a moderated standard deviation (0.94).

The *one way* repeated measures ANOVA was conducted to compare the confidence interval of the three algorithms with the same set of queries. The mean and standard deviation are presented in Tables 4.18 and 4.19.

TABLE 4.18: Precision: descriptive analysis

Methods	Mean	Std. Deviation	N
Method 1: <i>Tf-idf</i>	2.90	1.202	20
Method 1: <i>Naive</i>	3.52	0.745	20
Method 1: <i>SMSql</i>	3.55	0.936	20

There was a significant effect in the result of *SMSql* algorithms (Wilks' Lambda = .59,  $F(2, 18) = 6.261, p < .005$ , multivariate partial eta squared = .41).

Table 4.20 shows the descriptive analysis of the three methods on *average timing statistical analysis*, where 20 sample queries (N) were used. The results of the *SMSql* gave the best Mean (0.07) and moderate standard deviation (0.013).

TABLE 4.19: Multivariate test for precision

Effect		Value	$F$	Hypothesis df	Error df	Sig.	Partial $\eta^2$
Precision	Pillai's Trace	.410	6.261	2.000	18.000	.009	.410
	Wilks' Lambda	.590	6.261	2.000	18.000	.009	.410
	Hotelling's Trace	.696	6.261	2.000	18.000	.009	.410
	Roy's Largest Root	.696	6.261	2.000	18.000	.009	.410

The *one way* repeated measure, ANOVA, was conducted to compare the confidence interval of the three algorithms with the same set of queries. The mean and standard deviations are presented in *Tables 4.20* and *4.21*.

TABLE 4.20: Timing: descriptive analysis

Methods	Mean	Std. Deviation	N
Method 1: <i>Tf-idf</i>	0.07285	0.011454	20
Method 1: <i>naive</i>	0.07810	0.007947	20
Method 1: <i>SMSql</i>	0.06980	0.012940	20

There was a significant effect in the result of SMSql algorithms (Wilks' Lambda = .58,  $F(2, 18) = 6.444, p < .005$ , multivariate partial eta squared = .42).

TABLE 4.21: Multivariate test for timing

Effect	Value	$F$	Hypothesis df	Error df	Sig.	Partial $\eta^2$	
Time	Pillai's Trace	.417	6.444	2.000	18.000	.008	.417
	Wilks' Lambda	.583	6.444	2.000	18.000	.008	.417
	Hotelling's Trace	.716	6.444	2.000	18.000	.008	.417
	Roy's Largest Root	.716	6.444	2.000	18.000	.008	.417

## 4.6 Chapter summary

This chapter has presented the results obtained in attempting to meet the research objectives and answer research questions. It will be recalled that the research objectives and questions set out at the beginning of the thesis focused on SMS normalization and information retrieval efficiency in different algorithms using SMS texts. It is therefore important to examine whether the issues have been addressed using the methodologies adopted.

Six experimental proceedings were presented in the first research objective. Remarkable among them is the sixth experiment, *cross validation*, where the developed algorithm was compared with the existing BLEU method, using annotators. In comparing the results obtained based on annotator judgement there is a 23% performance difference between the normalization carried out by the algorithm developed by the researcher and the BLEU method used by the annotator.

Regarding the second research objectives and questions, the typical behaviour of SMS queries in a search engine was discussed. The need to improve on the retrieval mechanism of the SMS-based system was given attention. A new SMS-based information retrieval called *SMS question locator (SMSql)* was proposed. The technique of getting the score functions in order to rank the question-answer pair was considered. The keyword extraction technique as a way to improve the efficiency of FAQ in the IR system was also discussed. A series of experiments was performed using the three algorithms, *tf-idf*, *naive* and *SMSql*, to assess the retrieval efficiency through computational speed. The results proved that the *SMSql* algorithm was 10.3% and 4.1% better than the *naive* and *tf-idf* approaches respectively, using the execution time as the metric.

Based on the research questions posed, it is therefore apparent that the results obtained have answered the questions raised in the research questions.

## Chapter 5

# Discussion and Conclusion

### 5.1 Overview

This study has focused on the problem of SMS normalization and information accessing through the use of SMS. The aim has been to solve the problem of how to use *unclean* text for information dissemination while retaining security of communication. The thesis presents a state-of-the-art investigation into SMS normalization techniques for correcting informal text-writing that can be applied to information access in FAQ systems. The main objective of this research work is two-fold: (1) normalising the SMS text in order to transform it into its original English form, and (2) using the normalized text as an input to an FAQ system, to generate an answer to user requests. The thesis is structured in two parts, and the completion of the first part leads on to the second. Once SMS is normalized, it can be used for accessing information. The security of SMS text communication between the receiver and sender (FAQ database server) was ensured using the secure shell (SSH) protocol.

Short messages can be sent to a central database server containing answers to FAQ, in this case on the issue of HIV/AIDS. As the HIV/AIDS infection rate continues to rise, in particular among young adults, cell phones have been identified as one of the tools that can be used to meet the challenge of information dissemination. Access to appropriate and timely information can prevent and manage the spread of this disease and many other chronic illnesses. Within the young adult age group, information access using text messages has become particularly appealing. In this regard, access to carefully screened information on HIV/AIDS within the context of an FAQ system was developed. However, merely automating SMS-based information search and retrieval poses significant challenges because of the noise inherent in SMS communications. In this thesis, a special corpus of SMS messages was collected on issues related to HIV/AIDS. The SMS

messages were then analysed, classified, and normalized with the support of English and medical dictionaries, all referenced to HIV/AIDS issues.

In the first part of the research, informal representation of text has called for its normalization in order to use the text for other natural language processing. This research reviewed some of the existing methods of SMS normalization, and then came up with a better approach to solving the lingering problems of SMS translation. The new approach, named **S**earch, **C**ompare and **R**eplace (SCORE), uses combined character-based, unsupervised and rule-based algorithms to develop a normalization process. The implementation of the SCORE algorithm involves language-model concepts such as  $n$ -gram. An appropriate language model is applied where there is a need to consider one or more words or characters in a sentence or word. Six experiments were performed based on the first research objective. The *cross validation* experiment was used to compare the SCORE and BLEU algorithms. The results show SCORE having a 23% better performance when compared with the BLEU score.

One of the problems encountered in the course of the evaluation of SMS normalization research is the generation of an SMS corpus. No existing corpus was adequate to use for the training because of the mobile phone users' privacy concerns. This challenge was resolved by developing an algorithm to generate SMS from existing English words. With this innovation, the SMS-English pairs and the variants of SMS for a particular English token were developed. Another algorithm was developed to translate the generated SMS into English in order to measure the accuracy of the SMS translation into English terms.

In Section 2.8.1, an assumption was made and defended relating to the *order of vowel precedence*. It is evident from the literature that the vowel *e* evidenced the highest proportion of usage. This finding was used to make a final decision in a situation where candidate words tied in appropriateness. Tying arises when there is an equal chance of selecting from two or more candidate English words as the translation for the SMS term.

In the second stage of the experiment, the normalized text message was used for information retrieval processing. This research work was complemented with additional use of the *vector space model* from the SMS query and the FAQ documents. Keyword matching or extraction was performed on the principle of deleting stop words from the query sentence. The terms that are left are referred to as collocation terms with multi-term adjacency, because of their closeness to each other. The two concepts—collocation and multi-term adjacency—are similar to the use of  $N$ -grams. A new algorithm, *SMSql*, was developed which was able to locate the normalized keywords. These sets of keywords are used to match the keywords in the FAQ database. A one-to-one mapping of the five best ranked lists on the FAQ is released as the result of the enquiry. The results of



SMS queries are intended to meet the needs of people with low literacy skills and diverse communication styles. This technique was measured and compared with the existing *naive* and *tf-idf* models. The computational time of the *SMSql* method in returning the best 5 results was 10% and 4% faster than *naive* and *tf-idf* respectively.

SMS security was reviewed in an FAQ system, and enciphering the messages through the use of the secure shell protocol (SSH) was proposed. A cryptographic key generation algorithm is run on the data, the password is set within the file systems, backup management is performed and user-level access control executed. The data is passed through the communication channel and the data decrypted on the user's mobile handset. The decrypting algorithm installed on the user's mobile handset will authenticate the secured message and the user will then be able to access the message. SMS encryption methods are therefore proposed to secure communication during transfer from the FAQ database server.

## 5.2 Summary of research contribution

A well-researched thesis contributes to the progress, advancement, and enhancement of human knowledge by adding unique and original ideas to the body of knowledge. This is achieved by assessing research problems critically and working on results that have been obtained in the past, with a view to improving on them. For new knowledge to be established, new theories must be formulated or existing ones manipulated. The development of a new theory is often arrived at from research questions which arise from a thorough analysis of gaps existing in the literature under review (see *Table 5.1*).

New research should indicate convincingly how it identifies and addresses specific gaps in the literature. The research contribution, in Section 1.8, made by this thesis arrives at an improved solution to the problem of normalizing SMS text into formal English and using this normalized SMS text for secure information access.

## 5.3 Chapters recap

In order to present a clear understanding of what has been discussed in the previous chapters, this section briefly recapitulates what has been presented in the thesis.

**Chapter 1** provides a statement and analysis of the problem. It presents the research problems as well as the research questions, research methodology, research aims and objectives.

TABLE 5.1: Summary of research contribution

Category	Typical Activity	Remarks
Problem identification	Identified research objectives i.e. (SMS normalization and SMS-based information access)	Achieved through experimentation (and review of conference and journal papers)
Design	Designed novel SMS normalization and SMS-based information access algorithms	Pilot tested and evaluated
Comparison of the existing methodology	Compared several theoretical models, system designs, algorithmic methodologies or implementations in a novel way	Achieved
Implementation	Implemented the two research objectives	Achieved
Empirical analysis of the algorithms	Studied the performance of an implemented system in a novel way by observing and comparing results with the stated research objectives and research questions.	Effective and efficient
Application of the research	It has significance and numerous applications in health sector	HIV/AIDS case

**Chapter 2** describes the background to the two research objectives put forward regarding SMS normalization and SMS-based information access, in order (1) to solve the research problem set out in Chapter 1 and (2) to make decisions regarding key concepts implicit in the research objectives. The chapter also reviews the metrics and methods that are used to achieve the research objectives (Sections 2.5–2.8). The approach taken in selecting translations of SMS text is reviewed in Sections 2.7 and 2.8, and issues surrounding SMS security using SSH are also reviewed.

**Chapter 3** describes the design and development of the algorithms used to achieve the two research objectives, i.e. (1) SMS normalization and (2) information access using the normalized SMS text. Algorithms were developed to process the FAQ query in order to present the most suitable answer to the request of the SMS user.

**Chapter 4** presents results obtained through the evaluation of the algorithms developed for SMS normalization and SMS-based information access.

**Chapter 5** presents an overview, a summary of the research contribution and suggestions for further work.

## 5.4 Further work

A number of issues have been identified that may be able to take this research further. Four interesting directions opened up in the course of this research into SMS normalization and accessing mobile information through SMS. There may be a need to carry out research into the way indigenous (South African) text messages are written with a view of normalizing it. South Africa, a multilingual society, has nine different national languages apart from English and Afrikaans. SMS technology is welcomed by many people, and therefore its normalization is paramount if it is to be used as a medium of sharing and accessing information, especially in issues to do with HIV/AIDS education.

Normalization in terms of text-to-speech (TTS) is another promising research area to be looked into. The advantages for visually-impaired people are obvious. Professional safety can be improved, for instance in cases where, drivers can receive messages while driving, because the message is received aurally. The uptake of cellphone use is unprecedented among South African youth. In addition, fresh research could consider different ways of generating speech processing systems i.e., text-to-speech (TTS) in order to handle phonetic text created by the widespread use of phonetic spelling in the South African context. This might be achieved in collaboration with other research groups, like the South African Sign Language body (SASL).

Another issue has to do with curiosity to explore information access in SMS-based interfaces using multilingual and cross-lingual input, especially from South African languages. This area of investigation has the potential to bring the application to the grass roots, so that non-English speakers will benefit from the technology. The ability to query an FAQ database, in a language other than the one for which it was developed, is of great significance and practical value to the Southern African community. There will be a need for FAQ retrieval systems to be able to handle cross-lingual data such as the inherently noisy text of SMS queries.

The sensitivity of health-care information and its accessibility via the Internet and mobile technology systems is a cause for concern in these modern times. The privacy, integrity and confidentiality of a patient's data are key factors to be considered in the transmission of medical information for use by authorised health-care personnel. Mobile communication has enabled medical consultancy, treatment, drug administration and the provision of laboratory results to take place outside the hospital. With the implementation of electronic patient records and the Internet and Intranets, medical information sharing amongst relevant health-care providers was made possible. But the vital issue in this method of information sharing is security: the patient's privacy, as well as the confidentiality and integrity of the health-care information system, should

not be compromised. There is a need to examine various ways of ensuring the security and privacy of a patient's electronic medical information in order to ensure the integrity and confidentiality of the information.

This study of SMS normalization and SMS-based information access was carried out through experimentation using *PHP* and *MySQL* software in a computing environment focused on the HIV/AIDS scenario. The next stage will be to convert this research into real-life implementation. There is a strong belief that a real-life implementation of these research objectives will go a long way towards creating improved awareness, education and training to address the spread of HIV/AIDS in society at large.

# Bibliography

- [1] R. K. Abercrombie, F. T. Sheldon, K. R. Hauser, M. W. Lantz, and A. Mili. Failure impact analysis of key management in AMI using cybernomic situational assessment (CSA). In *Proceedings of the Eighth Annual Cyber Security and Information Intelligence Research Workshop*, page 19. ACM, 2013.
- [2] I. Abu El-Khair. Effects of stop words elimination for Arabic information retrieval: a comparative study. *International Journal of Computing and Information Sciences*, 4(3):119–133, 2006.
- [3] S. Acharyya, S. Negi, L. Subramaniam, and S. Roy. Unsupervised learning of multilingual short message service (SMS) dialect from noisy examples. In *Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data*, pages 67–74. ACM, 2008.
- [4] S. Acharyya, S. Negi, L. V. Subramaniam, and S. Roy. Language independent unsupervised learning of short message service dialect. *International Journal on Document Analysis and Recognition (IJDAR)*, 12(3):175–184, 2009.
- [5] M. Agoyi and D. Seral. SMS security: An asymmetric encryption approach. In *6th International Conference on Wireless and Mobile Communications (ICWMC)*, pages 448–452. IEEE, 2010.
- [6] S. Agyepong, M. F. Bosu, and J. B. Hayfron-Acquah. MSEARCH: A mobile search service. *Research Journal of Information Technology*, 3(2):108–112, 2011.
- [7] R. Ahmad, A. Sarlan, K. Maulod, E. M. Mazlan, and R. Kasbon. SMS-based final exam retrieval system on mobile phones. In *International Symposium in Information Technology*, volume 1, pages 1–5. IEEE, 2010.
- [8] F. Ahmed, E. De Luca, and A. Nürnberger. Revised N-gram based automatic spelling correction tool to improve retrieval effectiveness. *Polibits*, 40:39–48, 2009.
- [9] F. Alam, S. Habib, and M. Khan. Text normalization system for Bangla. Technical report, Center for Research on Bangla Language Processing (CRBLP), BRAC University, 2008.

- [10] Y. Ali and S. Smith. Flexible and scalable public key security for SSH. In *Public Key Infrastructure*, pages 43–56. Springer, 2004.
- [11] R. Alshammari, P. I. Lichodziejewski, M. Heywood, and A. N. Zincir-Heywood. Classifying SSH encrypted traffic with minimum packet header features using genetic programming. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, pages 2539–2546. ACM, 2009.
- [12] G. Anderson, S. Asare, Y. Ayalew, D. Garg, B. Gopolang, A. Masizana-Katongo, O. Mogotlhwane, D. Mpoeleng, and H. Nyongesa. Towards a bilingual SMS parser for HIV/AIDS information retrieval in Botswana. In *Proceedings of the Second IEEE/ACM International Conference of Information and Communication Technologies and Development (ICTD)*, pages 329–333, Bangalore, India, 2007.
- [13] G. Anderson, Y. Ayalew, P. Mokotedi, N. Motlogelwa, D. Mpoeleng, and E. Thuma. Healthcare FAQ information retrieval using a commercial database management system. In *Proceedings of the 2nd IASTED Africa Conference on Modelling and Simulation (AfricaMS 2008)*, pages 307–313, Gaborone, Botswana, 2010.
- [14] G. Andreou and I. Galantomos. Teaching idioms in a foreign language context: preliminary comments on factors determining Greek idiom instruction. *Metaphorik*, 15:7–26, 2008.
- [15] Anonymous. What is statistics? Technical report, Research and Computer Division, Department of Tourism, Government of Kerala, Thiruvananthapuram, 2010.
- [16] N. Anuar and A. B. M. Sultan. Validate conference paper using dice coefficient. *Computer and Information Science*, 3(3):139, 2010.
- [17] M. Attia. Arabic tokenization system. In *Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, pages 65–72. Association for Computational Linguistics, 2007.
- [18] R. A. Atun, S. R. Sittampalam, and A. Mohan. Uses and benefits of SMS in healthcare delivery. *Discussion paper. London: Imperial College.*, 2005.
- [19] A. Aw, M. Zhang, P. Yeo, Z. Fan, and J. Su. Input normalization for an English-to-Chinese SMS translation system. *MT Summit-2005*, 2005.
- [20] A. Aw, M. Zhang, J. Xiao, and J. Su. A phrase-based statistical model for SMS text normalization. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, pages 33–40. Association for Computational Linguistics, 2006.

- [21] M. Badawi, A. Mohamed, A. Hussein, and M. Gheith. Maintaining the search engine freshness using mobile agent. *Egyptian Informatics Journal*, 2012.
- [22] R. Baeza-Yates. Information retrieval in the web: beyond current search engines. *International Journal of Approximate Reasoning*, 34(2):97–104, 2003.
- [23] R. Barzilay and M. Elhadad. Using lexical chains for text summarization. In *Workshop on Intelligent Scalable Text Summarization*, pages 10–17, Madrid, Spain, 1997. Association for Computational Linguistics.
- [24] R. Beaufort and C. Fairon. Normalisation of noisy typewritten texts, 2013.
- [25] R. Beaufort, S. Roekhaut, L.-A. Cougnon, and C. Fairon. A hybrid rule/model-based finite-state framework for normalizing SMS messages. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguists*, pages 770–779, Uppsala, Sweden, 2010.
- [26] S. Bergvik and R. Wynn. The use of short message service (SMS) among hospitalized coronary patients. *General Hospital Psychiatry*, 34(4):390–397, 2012.
- [27] M. Bieswanger. abbrevi8 or not 2 abbrevi8: A contrastive analysis of different shortening strategies in English and German text messages. *SALSA XIV*, 2006.
- [28] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 92–100. ACM, 1998.
- [29] V. Bochkarev, A. Shevlyakova, and V. Solovyev. Average word length dynamics as indicator of cultural changes in society. *arXiv preprint arXiv:1208.6109*, 2012.
- [30] R. C. Bogdan and S. K. Biklen. *Qualitative research in education. An introduction to theory and methods*. ERIC, 1998.
- [31] D. Boneh, N. Modadugu, and M. Kim. Generating RSA keys on a handheld using an untrusted server. In *Progress in CryptologyINDOCRYPT 2000*, pages 271–282. Springer, 2000.
- [32] K. Born and D. Gustafson. Detecting dns tunnels using character frequency analysis. *arXiv preprint arXiv:1004.4358*, 2010.
- [33] T. Brants, A. Popat, P. Xu, F. Och, and J. Dean. Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Language Learning*, pages 858–867. Citeseer, 2007.





- [45] C. Callison-Burch, M. Osborne, and P. Koehn. Re-evaluation the role of Bleu in machine translation research. In *EACL*, volume 6, pages 249–256, 2006.
- [46] G. Carter, A. Clark, E. Dawson, and L. Nielsen. Analysis of DES double key mode. In: *Proceedings of the IFIP TC11 Eleventh International Conference on Information Security, Chapman and Hall, UK*, pages 113–127, 1995.
- [47] C. Cassell and G. Symon. Qualitative research in work contexts. *Qualitative methods in organizational research: A practical guide*, pages 1–13, 1994.
- [48] J. Catlett. Megainduction: A test flight. In *Proceedings of the Eighth International Workshop on Machine Learning*, pages 596–599, 1991.
- [49] A. Chabossou, C. Stork, M. Stork, and Z. Zahonogo. Mobile telephony access and usage in Africa. In *International Conference on Information and Communication Technologies and Development (ICTD)*, pages 392–405. IEEE, 2009.
- [50] J. Chen, L. Subramanian, and E. Brewer. SMS-based mobile web search for low-end phones. In *16th Annual International Conference on Mobile Computing and Networking, ACM*, pages 125–135, 2010.
- [51] K.-Y. Chen and K.-M. Chao. A fully compressed algorithm for computing the edit distance of run-length encoded strings. *Algorithmica*, 65(2):354–370, 2013.
- [52] X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the Sixth ACM international Conference on Web Search and Data Mining*, pages 193–202. ACM, 2013.
- [53] D. Chor. Unpacking sources of comparative advantage: A quantitative approach. *Journal of International Economics*, 82(2):152–167, 2010.
- [54] M. Choudhury, R. Saraf, V. Jain, A. Mukherjee, S. Sarkar, and A. Basu. Investigation and modeling of the structure of texting language. *International Journal on Document Analysis and Recognition*, 10(3):157–174, 2007.
- [55] K. W. Church. Char align: a program for aligning parallel texts at the character level. In *Proceedings of the 31st Annual Meeting on Association for Computational Linguistics*, pages 1–8. Association for Computational Linguistics, 1993.
- [56] E. Clark and K. Araki. Two database resources for processing social media English text. In *Proceedings of the Eight International Conference on Language Resources and Evaluation, LREC’12*, 2012.
- [57] D. Contractor, T. A. Faruquie, and L. V. Subramaniam. Unsupervised cleansing of noisy text. In *Proceedings of the 23rd International Conference on Computational*

- Linguistics: Posters*, pages 189–196. Association for Computational Linguistics, 2010.
- [58] W. Y. Conwell. *Methods and systems for content processing*, 2012.
- [59] P. Cook and S. Stevenson. An unsupervised model for text message normalization. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, pages 71–78, Boulder, Colorado, 2009. Association for Computational Linguistics.
- [60] G. Cormode and S. Muthukrishnan. The string edit distance matching problem with moves. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 667–676. Society for Industrial and Applied Mathematics, 2002.
- [61] G. Cormode and S. Muthukrishnan. The string edit distance matching problem with moves. *ACM Transactions on Algorithms (TALG)*, 3(1):2, 2007.
- [62] W. J. Corvey, S. Vieweg, T. Rood, and M. Palmer. Twitter in mass emergency: What NLP techniques can contribute. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, pages 23–24. Association for Computational Linguistics, 2010.
- [63] M. Costa-Jussá and R. Banchs. Automatic normalization of short texts by combining statistical and rule-based techniques. *Language Resources and Evaluation*, 47(1):179–193, 2013.
- [64] M. Crotty. *The Foundations of Social Research: Meaning and Perspective in the Research Process*. Sage, 1998.
- [65] F. Damerau. A technique for computer detection and correction of spelling errors. *Comm. ACM*, 7(3):171–176, 1964.
- [66] P. Darke, G. Shanks, and M. Broadbent. Successfully completing case study research: combining rigour, relevance and pragmatism. *Information systems journal*, 8(4):273–289, 1998.
- [67] E. D’Avanzo and M. Bernardo. A keyphrase-based approach to summarization: the LAKE system at DUC-2005. In *Document Understanding Conference, 2005*, 2005.
- [68] P. Deepak and V. Subramaniam. Correcting SMS text automatically. *CSI communications*, pages 9–11, 2012.

- [69] Y. Deng, J. Xu, and Y. Gao. Phrase table training for precision and recall: What makes a good phrase and a good phrase pair. *Proceedings of ACL/HLT 2008*, pages 81–88, 2008.
- [70] S. Deorowicz and M. G. Ciura. Correcting spelling errors by modelling their causes. *International Journal of Applied Mathematics and Computer Science*, 15(2):275, 2005.
- [71] K. DeRose. What is epistemology. *A brief introduction to the topic*, 20, 2002.
- [72] P. Dewan, J. Grudin, and E. Horvitz. Towards mixed-initiative access control. In *International Conference on Collaborative Computing: Networking, Applications and Worksharing. CollaborateCom 2007.*, pages 64–71, 2007.
- [73] G. M. Di Nunzio, J. Leveling, and T. Mandl. Multilingual log analysis: LogCLEF. In *Advances in Information Retrieval*, pages 675–678. Springer, 2011.
- [74] A. Dmitrienko, Z. Hadzic, H. Lhr, A.-R. Sadeghi, and M. Winandy. Securing the access to electronic health records on mobile phones. In *Biomedical Engineering Systems and Technologies*, volume 273, pages 365–379. Springer Berlin Heidelberg, 2013.
- [75] D. Doggett and L. G. Richards. A re-examination of the effect of word length on recognition thresholds. *The American Journal of Psychology*, 88(4):583–594, 1975.
- [76] L. Dolamic and J. Savoy. When stopword lists make the difference. *Journal of the American Society for Information Science and Technology*, 61(1):200–203, 2010.
- [77] E. Dragut, F. Fang, P. Sistla, C. Yu, and W. Meng. Stop word and related problems in web interface integration, 2009.
- [78] A. Drummond and J. Campling. *Research Methods for Therapists*. Nelson Thornes, 1996.
- [79] C. Dürscheid and E. Stark. sms4science. An international corpus-based texting project and the specific challenges for multilingual Switzerland. *Digital Discourse: Language in the New Media*, page 299, 2011.
- [80] J. Edosomwan and T. Edosomwan. Comparative analysis of some search engines. *South African Journal of Science*, 106(11/12):1–4, 2010.
- [81] L. Egghe. The distribution of n-grams. *Scientometrics*, 47(2):237–252, 2000.
- [82] Z. Elberrichi, A. Rahmoun, and M. A. Bentaallah. Using wordnet for text categorization. *Int. Arab J. Inf. Technol.*, 5(1):16–24, 2008.

- [83] M. A. Elmi and M. Evens. Spelling correction using context. In *Proceedings of the 17th International Conference on Computational Linguistics-Volume 1*, pages 360–364. Association for Computational Linguistics, 1998.
- [84] D. Embrey. Understanding human behaviour and error. *Human Reliability Associates*, 1, 2005.
- [85] C. Fairon and S. Paumier. A translated corpus of 30,000 French SMS. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, pages 351–354, Sweden, 2006.
- [86] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37, 1996.
- [87] O. Ferschke, I. Gurevych, and M. Rittberger. Flawfinder: A modular system for predicting quality flaws in wikipedia. In *CLEF (Online Working Notes/Labs/-Workshop)*, 2012.
- [88] T. Fischer, K. De Biswas, J. Ham, R. Naka, and W. Huang. A multi-agent expert system shell for shape grammars, 2012.
- [89] C. Fox. A stop list for general text. *SIGIR Forum ACM*, 24(1–2):19–21, 1989.
- [90] R. Franceschini and A. Mukherjee. Data compression using encrypted text. In *Proceedings of the Third Forum on Research and Technology Advances in Digital Libraries, ADL '96.*, pages 130–138. IEEE, 1996.
- [91] P. Gadde, R. Goutam, R. Shah, H. S. Bayyrapu, and L. Subramaniam. Experiments with artificially generated noise for cleansing noisy text. In *Proceedings of the 2011 Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data*, page 4. ACM, 2011.
- [92] J. Gill and P. Johnson. *Research Methods for Managers*. Sage, 2002.
- [93] S. Goel and S. Yadav. An overview of search engine evaluation strategies. *International Journal of Applied Information Systems*, 1:7–10, 2012.
- [94] S. Gouws, D. Hovy, and D. Metzler. Unsupervised mining of lexical variants from noisy text. In *Proceedings of the First workshop on Unsupervised Learning in NLP*, pages 82–90. Association for Computational Linguistics, 2011.
- [95] B. Han, P. Cook, and T. Baldwin. Lexical normalization for social media text. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(1):5, 2013.
- [96] S. Harding. Rethinking standpoint epistemology: What is strong objectivity? *Knowledge and inquiry: Readings in epistemology*, pages 352–384, 2002.

- [97] J. Hellström and P.-E. Tröften. *The innovative use of mobile applications in East Africa*. Swedish International Development cooperation Agency (SIDA), 2010.
- [98] M. L. Henson. Security mailbox, Oct. 9 2001. US Patent 6,299,061.
- [99] G. Hirst and A. Budanitsky. Correcting real-word spelling errors by restoring lexical cohesion. *Department of Computer Science Toronto, Ontario, Canada M5S 3G4 gh@cs.toronto.edu, abm@cs.toronto.edu*, 2003.
- [100] D. Hogan, J. Leveling, H. Wang, P. Ferguson, and C. Gurrin. DCU @ FIRE 2011: SMS-based FAQ retrieval. In *3rd Workshop of the Forum for Information Retrieval Evaluation, FIRE*, pages 2–4, 2011.
- [101] J. Hogg. *Web service security: Scenarios, patterns, and implementation guidance for Web Services Enhancements (WSE) 3.0*. O’Reilly Media, Inc., 2006.
- [102] E. Hoggan, S. A. Brewster, and J. Johnston. Investigating the effectiveness of tactile feedback for mobile touchscreens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1573–1582, Florence, Italy, 2008. ACM.
- [103] M. Holmqvist, S. Stymne, L. Ahrenberg, and M. Merkel. Alignment-based reordering for SMT. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*, pages 3436–3440, 2012.
- [104] A. S. Hornby. *Oxford Advanced Learner’s Dictionary of Current English*. Cambridge Univ. Press, 2006.
- [105] V. Hoste, S. Gillis, and W. Daclemans. A rule induction approach to modeling regional pronunciation variation. In *Proceedings of the 18th Conference on Computational Linguistics-Volume 1*, pages 327–333. Association for Computational Linguistics, 2000.
- [106] Y. How and M. Kan. Optimizing predictive text entry for short message service on mobile phones. In M. J. Smith and G. Salvendy, editors, *Proceedings of Human Computer Interfaces International*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc., 2005.
- [107] L. B. Huang, V. Balakrishnan, and R. G. Raj. Improving the relevancy of document search using the multi-term adjacency keyword-order model. *Malaysian Journal of Computer Science*, 25:1, 2012.
- [108] A. Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Conference on Empirical Methods in Natural Language Processing*, pages 216–223, Sapporo, Japan, 2003.

- [109] Ipoletse. Ipoletse training manual for call centre operators for the national call centre on HIV and AIDS, Gaborone, Botswana, 2002.
- [110] M. Iraba, I. Venter, and W. Tucker. Using Inexpensive Mobile Technologies to empower rural farmers. In *Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2010*, Spier Estate, Stellenbosch, South Africa, 2010.
- [111] A. Irvine, W. Jonathan, and C. Callison-Burch. Processing informal, romanized Pakistani text messages. *Center for Language and Speech Processing Johns Hopkins University*, 2012.
- [112] A. Islam and D. Inkpen. Real-word spelling correction using Google Web IT 3-grams. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, volume 3, pages 1241–1249. ACL, 2009.
- [113] S. Itzhaky, S. Gulwani, N. Immerman, and M. Sagiv. A simple inductive synthesis methodology and its applications. In *ACM Sigplan Notices*, volume 45, pages 36–46. ACM, 2010.
- [114] D. A. Jackson, K. M. Somers, and H. H. Harvey. Similarity coefficients: measures of co-occurrence and association or simply measures of occurrence? *American Naturalist*, pages 436–453, 1989.
- [115] M. Jain. N-gram driven SMS based FAQ retrieval system. Master’s thesis, Delhi College of Engineering Delhi University, 2012.
- [116] B. J. Jansen, A. Spink, J. Bateman, and T. Saracevic. Real life information retrieval: a study of user queries on the web. In *ACM SIGIR Forum*, volume 32, pages 5–17. ACM, 1998.
- [117] K. Jasmeen and V. Gupta. Effective approaches for extraction of keywords. *IJCSI International Journal of Computer Science Issues*, 7(6), 2010.
- [118] R. Jernigan. A photographic view of cumulative distribution functions. *Journal of Statistics, Education Volume*, 16, 2008.
- [119] L. Jianan and P. Sävström. An intelligent FAQ answering system using a combination of statistic and semantic IR techniques. Master’s thesis, Dept. of Computer Science and Electronics at Mälardalen University, Sweden, 2006.
- [120] R. Jizba. Searching, part 4: Recall and precision: key concepts for database searchers, 2007.

- [121] A. Joshi. Improving accuracy of SMS based FAQ retrieval. *International Journal of Emerging Technologies in Computational and Applied Sciences*, pages 362–366, 2012.
- [122] Z. Junliang, Z. Xuefang, and Z. Guang. Designing an automated FAQ answering system for farmers based on hybrid strategies. *Chinese Journal of Library and Information Science*, 5(4):1–36, 2012.
- [123] M. Kaufmann and J. Kalita. Syntactic normalization of Twitter messages. In *Proceedings of the International Conference on Natural Language Processing, Kharagpur, India*, 2010.
- [124] B. Kaur, A. Saxena, and S. Singh. Web opinion mining for social networking sites. In *Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology*, pages 598–605. ACM, 2012.
- [125] J. Kaur and V. Gupta. Effective approaches for extraction of keywords. *Journal of Computer Science*, 7(6):144–148, 2010.
- [126] H. S. Knewton and R. W. Sias. Why susie owns starbucks: The name letter effect in security selection. *Journal of Business Research*, 63(12):1324–1327, 2010.
- [127] C. Kobus, F. Yvon, and G. Damnati. Normalizing SMS: are two metaphors better than one? In *Proceedings of the 22nd International Conferences on Computational Linguistics (COLING 2008)*, pages 441–448, Manchester, 2008.
- [128] P. Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, 2005.
- [129] P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 48–54, Edmonton, Canada, 2003. Association for Computational Linguistics.
- [130] L. M. Kohnfelder. *Towards a practical public-key cryptosystem*. PhD thesis, Massachusetts Institute of Technology, 1978.
- [131] G. Kondrak, D. Marcu, and K. Knight. Cognates can improve statistical translation models. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003)*, pages 46–48., Edmonton, 2003.
- [132] G. Kothari, S. Negi, T. A. Faruque, V. T. Chakaravarthy, and L. V. Subramaniam. SMS based interface for FAQ retrieval. In *Proceedings of the 47th Annual Meeting*

- of the ACL and the 4th IJCNLP of the AFNLP, pages 852–860, Suntec Singapore, 2009.
- [133] P. Kumar Jaiswal. *SMS Based Information Systems*. PhD thesis, University of Eastern Finland., 2011.
- [134] A. Langer, R. Banga, A. Mittal, and L. Subramaniam. Variant search and syntactic tree similarity based approach to retrieve matching questions for SMS queries, 2010.
- [135] L. Lee. Measures of distributional similarity. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, ACL '99, pages 25–32. Association for Computational Linguistics, 1999.
- [136] J. Leveling. On the effect of stopword removal for SMS-based FAQ retrieval. In G. Bouma, A. Ittoo, E. Métais, and H. Wortmann, editors, *Natural Language Processing and Information Systems*, volume 7337 of *Lecture Notes in Computer Science*, pages 128–139. Springer Berlin Heidelberg, 2012.
- [137] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. In *Soviet Physics Doklady*, volume 10, pages 707–710, 1966.
- [138] M. Li, C. Zong, and H. T. Ng. Automatic evaluation of Chinese translation output: word-level or character-level? In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 159–164. Association for Computational Linguistics, 2011.
- [139] W. Li, P. Miramontes, and G. Cocho. Fitting ranked linguistic data with two-parameter functions. *Entropy*, 12(7):1743–1764, 2010.
- [140] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 296–304, Madison, WI, 1998.
- [141] D. Lisonek and M. Drahansky. SMS encryption for mobile communication. In *International Conference on Security Technology SECTECH '08. IEEE computer society*, pages 198–201, 2008.
- [142] F. Liu, F. Weng, B. Wang, and Y. Liu. Insertion, deletion, or substitution? Normalizing text messages without pre-categorization nor supervision. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: shortpapers*, pages 71–76, Portland, Oregon, 2011.



- [143] F. Liu, F. Weng, and X. Jiang. A broad-coverage normalization system for social media language. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, pages 1035–1044, Jeju Island, Korea, 2012. Association for Computational Linguistics.
- [144] D. Y. Ma and H. B. Lai. Designing and development of the aerial surveying digital data management information system based on two-tier c/s structure model. *Advanced Materials Research*, 610:3702–3707, 2013.
- [145] I. S. MacKenzie and R. W. Soukoreff. Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, pages 754–755, Ft. Lauderdale, Florida, USA, 2003. ACM.
- [146] P. F. MacNeilage. Typing errors as clues to serial ordering mechanisms in language behavior. *Language and Speech*, 7:144–59, 1964.
- [147] N. Mahmud, J. Rodriguez, and J. Nesbit. A text message-based intervention to bridge the healthcare communication gap in the rural developing world. *Technology and Health Care*, 18(2):137–144, 2010.
- [148] G. Maiolini, A. Baiocchi, A. Rizzi, and C. Di Iollo. Statistical classification of services tunneled into SSH connections by a K-means based learning algorithm. In *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference IWCMC '10*, pages 742–746, New York, NY, USA, 2010. ACM.
- [149] S. Maleki-Dizaji. *Evolutionary Learning Multi-Agent Based Information Retrieval Systems*. PhD thesis, Sheffield Hallam University, 2003.
- [150] S. Mallat and A. Zouaghi. Proposal of a method of enriching queries by statistical analysis to search for information in Arabic, 2010.
- [151] T. Mandl, M. Agosti, G. M. Di Nunzio, A. Yeh, I. Mani, C. Doran, and J. M. Schulz. LogCLEF 2009: the CLEF 2009 multilingual logfile analysis track overview. In *Multilingual Information Access Evaluation I. Text Retrieval Experiments*, pages 508–517. Springer, 2010.
- [152] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*, volume 1. Cambridge University Press Cambridge, 2008.
- [153] A. Masizana-Katongo and T. Ama-Njoku. Example-based parsing solution for a HIV and AIDS FAQ system. *International Journal of Research and Reviews in Wireless Communications (IJRRWC)*, 1(3):59–65, 2011.

- [154] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(01):157–169, 2004.
- [155] M. S. Mayzner and M. E. Tresselt. Tables of single-letter and diagram frequency counts for various word-length and letter-position combinations. *Psychonomic Monograph Supplements*, 1(2):13–32, 1965.
- [156] J. C. McCroskey and V. P. Richmond. Willingness to communicate: Differing cultural perspectives. *Southern Journal of Communication*, 56(1):72–77, 1990.
- [157] V. D. Mea and S. Mizzaro. Measuring retrieval effectiveness: a new proposal and a first experimental validation. *Journal of the American Society for Information Science and Technology*, 55(6):530–543, 2004.
- [158] J. Meier, A. Boehm, E. Schreiber, S. Lippert, T. Neumuth, and S. Bohn. Development of a modular IT-Framework supporting the oncological patient treatment in ENT Surgery. *Biomed Tech*, 57:1, 2012.
- [159] I. D. Melamed, R. Green, and J. P. Turian. Precision and recall of machine translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume of the Proceedings of HLT-NAACL 2003–short papers–Volume 2*, pages 61–63. Association for Computational Linguistics, 2003.
- [160] D. Metzler and O. Kurland. Experimental methods for information retrieval. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1185–1186. ACM, 2012.
- [161] R. S. Michalski and R. E. Stepp. Learning from observation: Conceptual clustering. *Machine Learning: An artificial intelligence approach*, 1:331–363, 1983.
- [162] J. Michelizzi. *Semantic Relatedness Applied to All Words Sense Disambiguation*. PhD thesis, University of Minnesota, 2005.
- [163] R. Mihalcea and P. Tarau. TextRank: Bringing order into texts. In *Proceedings of EMNLP*, volume 4, pages 404–411. Barcelona, Spain, 2004.
- [164] A. Mittal and A. Sengupta. Improvised layout of keypad entry system for mobile phones. *Computer Standards & Interfaces*, 31(4):693–698, 2009.
- [165] G. Moerdler, K. McKeown, and J. Ensor. Building natural language interfaces for rule-based expert systems. *IJCAI-87*, pages 682–687, 1987.

- [166] A. Mogadala, K. Rambhoopal, and V. Varma. Language modeling approach to retrieval for SMS and FAQ matching. In *Proceedings of The 18th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2012.
- [167] D. Moldovan, S. Harabagiu, M. Pasca, R. Mihalcea, R. Girju, R. Goodrum, and V. Rus. The structure and performance of an open-domain question answering system. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 563–570. Association for Computational Linguistics, 2000.
- [168] M. F. Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, 6(4):525–533, 1993.
- [169] A. Moreo, M. Romero, J. Castro, and J. Zurita. FAQtory: A framework to provide high-quality FAQ retrieval systems. *Expert Systems with Applications*, 2012a.
- [170] A. Moreo, M. Navarro, J. Castro, and J. Zurita. A high-performance FAQ retrieval method using minimal differentiator expressions. *Knowledge-Based Systems*, 2012b.
- [171] M. L. Mphahlele and K. Mashamaite. The impact of short message service (SMS) language on language proficiency of learner’s and the SMS dictionaries: A challenge for educators and lexicographers. *IADIS International Conference Mobile Learning*, 2005.
- [172] W. Murnane. *Improving Accuracy of Named Entity Recognition on Social Media Data*. PhD thesis, University of Maryland, 2010.
- [173] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, 2007.
- [174] J. Oliva, J. I. Serrano, and M. Dolores del Castillo. SMS normalization: combining phonetics, morphology and semantics. *CAEPIA*, pages 273–282, 2012a.
- [175] J. Oliva, J. I. Serrano, M. D. del Castillo, and Á. Iglesias. A SMS normalization system integrating multiple grammatical resources. *Natural Language Engineering*, 19(1):121–141, 2012b.
- [176] M. Olivier. *Information Technology Research: A Practical Guide for Computer Science and Informatics*. Van Schaik, South Africa, 2009.
- [177] V. Ostojic, B. Cvoriscec, S. Ostojic, D. Reznikoff, A. Stipic-Markovic, and Z. Tudjman. Improving asthma control through telemedicine: a study of short-message service. *Telemed J E Health*, 11(28–35), 2005.

- [178] P. Pakray, S. Pal, S. Poria, S. Bandyopadhyay, and A. Gelbukh. SMSFR: SMS-Based FAQ Retrieval system. In *Advances in Computational Intelligence*, pages 36–45. Springer, 2013.
- [179] J. Pallant. *SPSS survival manual: A step by step guide to data analysis using SPSS*. Open University Press, 2010.
- [180] K. Panchapagesan, P. Talukdar, N. Krishna, K. Bali, and A. Ramakrishnan. Hindi text normalization. In *Fifth International Conference on Knowledge Based Computer Systems (KBCS)*, pages 19–22. Citeseer, 2004.
- [181] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceeding ACL*, pages 311–318, 2002.
- [182] G. Pass, A. Chowdhury, and C. Torgeson. A picture of search, 2006.
- [183] D. Pennell and Y. Liu. Toward text message normalization: Modeling abbreviation generation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*,, pages 5364–5367. IEEE, 2011a.
- [184] D. Pennell and Y. Liu. A character-level machine translation approach for normalization of SMS abbreviations. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 974–982, 2011b.
- [185] D. L. Pennell. *Normalization of Informal Text for Text-to-speech*. PhD thesis, The University of Texas, 2011.
- [186] J. L. Peterson. Computer programs for detecting and correcting spelling errors. *Communication of the ACM*, 23:676–687, 1980.
- [187] C. Pfleeger. *Security in Computing*. Prentice Hall, 1997.
- [188] D. Pinto, D. Vilarino, Y. Aleman, H. Gomez, and N. Loya. The soundex phonetic algorithm revisited for SMS based information retrieval. Technical report, Faculty of Computer Science, Benemerita Universidad Autonoma de Puebla, Mexico, 2011.
- [189] P. Pudil, J. Novovicov, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15(11):1119–1125, 1994.
- [190] P. Puschner and C. Koza. Calculating the maximum execution time of real-time programs. *Real-Time Systems*, 1(2):159–176, 1989.
- [191] B. Qin, Y. Xia, S. Prabhakar, and Y. Tu. A rule-based classification algorithm for uncertain data. In *IEEE 25th International Conference on Data Engineering ICDE'09.*, pages 1633–1640. IEEE, 2009.

- 
- [192] J.-J. Quisquater and C. Couvreur. Fast decipherment algorithm for RSA public-key cryptosystem. *Electronics letters*, 18(21):905–907, 1982.
- [193] K. Raghunathan and S. Krawczyk. Investigating SMS text normalization using statistical machine translation, 2007.
- [194] J. Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the First Instructional Conference on Machine Learning*, 2003.
- [195] J. Ratsaby and S. S. Venkatesh. Learning from a mixture of labeled and unlabeled examples with parametric side information. In *Proceedings of the Eighth Annual Conference on Computational Learning Theory*, pages 412–417. ACM, 1995.
- [196] J. Reason. *Human Error*. Cambridge University Press, 1990.
- [197] S. Z. Riehemann. *A Constructional Approach to Idioms and Word Formation*. PhD thesis, Stanford University, 2001.
- [198] E. S. Ristad and P. N. Yianilos. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532, 1998.
- [199] A. Robertson and P. Willett. Applications of n-grams in textual information systems. *Journal of Documentation*, 54(1):48–67, 1998.
- [200] D. E. Rose and D. Levinson. Understanding user goals in web search. In *Proceedings of the 13th International Conference on World Wide Web*, pages 13–19. ACM, 2004.
- [201] S. Rose, D. Engel, N. Cramer, and W. Cowley. Automatic keyword extraction from individual documents. *Text Mining*, pages 1–20, 2010.
- [202] M. S. Ryan and G. R. Nudd. The Viterbi algorithm. Technical report, University of Warwick, 1993.
- [203] G. Salton, A. Wong, and C. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.
- [204] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys (CSUR)*, 34(1):1–47, 2002.
- [205] L. H. Shaffer and J. Hardwick. Typing performance as a function of text. *Quarterly Journal of Experimental Psychology*, 20:360–369, 1968.
- [206] L. H. Shaffer and J. Hardwick. Errors and error detection in typing. *Quarterly Journal of Experimental Psychology*, 21:209–213, 1969.

- [207] D. Silverman. *Doing Qualitative Research: A Practical Handbook*. SAGE Publications Limited, 2013.
- [208] M. Simard, G. Foster, and P. Isabelle. Using cognates to align sentences in bilingual corpora. In *Conference of the Centre for Advanced Studies on Collaborative research: Distributed Computing*, volume 2, pages 1071–82, 1993.
- [209] A. Singhal. Modern information retrieval: A brief overview. *IEEE Data Engineering Bulletin*, 24(4):35–43, 2001.
- [210] F. Smith. *Research Methods in Pharmacy Practice*. Pharmaceutical Press, 2002.
- [211] T. Sohn, K. Li, W. Griswold, and J. Hollan. A diary study of mobile information needs. In *Proceedings of the 26th Annual SIGCHI Conference on Human factors in Computing Systems*, 2008.
- [212] H. Soltau, F. Metze, C. Fugen, and A. Waibel. A one-pass decoder based on polymorphic linguistic context assignment. In *IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU'01*, pages 214–217. IEEE, 2001.
- [213] P. Soucy and G. W. Mineau. Beyond TF-IDF weighting for text categorization in the vector space model. In *International Joint Conference on Artificial Intelligence*, volume 19, page 1130. Lawrence Erlbaum Associates Ltd., 2005.
- [214] R. W. Soukoreff. *Quantifying Text Entry Performance*. PhD thesis, York University Toronto, 2010.
- [215] R. Sproat, A. W. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards. Normalization of non-standard words. *Computer Speech & Language*, 15(3):287–333, 2001.
- [216] A. G. K. Stevenson-Taylor and W. Mansell. Exploring the role of art-making in recovery, change, and self-understanding—an interpretative phenomenological analysis of interviews with everyday creative people. *International Journal of Psychological Studies*, 4(3):p104, 2012.
- [217] C. Tagg. *A Corpus Linguistics Study of SMS Text Messaging*. PhD thesis, School of English, Drama and American and Canadian Studies, 2009.
- [218] W. F. Tichy, P. Lukowicz, L. Prechelt, and E. A. Heinz. Experimental evaluation in computer science: A quantitative study. *Journal of Systems and Software*, 28(1):9–18, 1995.
- [219] M. Tomitsch, F. Sturm, M. Konzett, A. Bolin, I. Wagner, and T. Grechenig. Stories from the field: mobile phone usage and its impact on people’s lives in East Africa.

- In *Proceedings of the 4th ACM/IEEE International Conference on Information and Communication Technologies and Development*, page 49. ACM, 2010.
- [220] M. Toorani and A. Beheshti. SSMS-A secure SMS messaging protocol for the m-payment systems. In *IEEE Symposium on Computers and Communications (ISCC 2008)*, pages 700–705. IEEE, 2008.
- [221] K. Toutanova and R. Moore. Pronunciation modeling for improved spelling correction. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 144–151, Philadelphia, PA, 2002.
- [222] G. Vaitheeswaran, P. Ghosh, and T. Fatemi. Methodology providing high-speed shared memory access between database middle tier and database server, Feb. 3 2004. US Patent 6,687,702.
- [223] D. Vilariño, D. Pinto, B. Beltrán, S. León, E. Castillo, and M. Tovar. A machine-translation method for normalization of SMS. *LNCS*, pages 293–302, 2012.
- [224] L. Wang and D. Wang. Method for ranking and sorting electronic documents in a search result list based on relevance, 2007. US Patent 7,814,099.
- [225] L. S. Wang. Relevance weighting of multi-term queries for vector space model. In *IEEE Symposium on Computational Intelligence and Data Mining, CIDM'09*, pages 396–402. IEEE, 2009.
- [226] Q. Wang, D. Korokin, and Y. Shang. Efficient dominant point algorithms for the multiple longest common subsequence (MLCS) problem. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, volume 7, pages 1494–1499, Pasadena, America, 2009.
- [227] C. Wartena, R. Brussee, and R. Brussee. Keyword extraction using word co-occurrence. *Database and Expert Systems Applications (DEXA) 2010 Workshop*, pages 54–58, 2010.
- [228] S. Weiser, L.-A. Cougnon, and P. Watrin. Temporal expressions extraction in SMS messages. *Information Extraction and Knowledge Acquisition*, page 41, 2011.
- [229] N. A. Weiss and C. A. Weiss. *Introductory Statistics*. Pearson Education, 2012.
- [230] D. West. How mobile devices are transforming healthcare. *Issues in technology innovation*, 2012.
- [231] C. Whitelaw, B. Hutchinson, G. Chung, and G. Ellis. Using the web for language independent spell-checking and auto-correction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Association for Computational Linguistics*, volume 2, pages 890–899, 2009.

- [232] J. D. Williams and S. Balakrishnan. Estimating probability of correctness for ASR N-best lists. In *Proceedings of the SIGDIAL 2009 Conference: The 10th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 132–135. Association for Computational Linguistics, 2009.
- [233] J. O. Wobbrock and B. A. Myers. Analyzing the input stream for character-level errors in unconstrained text entry evaluations. *ACM Trans. Comput. Hum. Interact.*, 13(4):458–489, 2006.
- [234] D. Wu, Y. Zhang, S. Zhao, and T. Liu. Identification of web query intent based on query text and web knowledge. In *First International Conference on Pervasive Computing, Signal Processing and Applications*, pages 128–131, Harbin, China, 2010.
- [235] Z. Xue, D. Yin, and B. D. Davison. Normalizing microtext. In *Proceedings of the AAAI-11 Workshop on Analyzing Microtext: San Francisco, USA Department of Computer Science & Engineering, Lehigh University Bethlehem, PA 18015 USA*, pages 74–79, 2011.
- [236] D. Yang, Y.-c. Pan, and S. Furui. Automatic Chinese abbreviation generation using conditional random field. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 273–276. Association for Computational Linguistics, 2009.
- [237] J. Yang, Y. Xu, and Y. Shang. An efficient parallel algorithm for longest common subsequence problem on GPUs. In *Proceedings of the World Congress on Engineering 2010 WCE 2010*, volume 1, London, U.K., 2010.
- [238] E. Yannakoudakis and D. Fawthrop. An intelligent spelling error corrector. *Information Processing & Management*, 19(2):101–108, 1983.
- [239] T. Ylonen and C. Lonvick. The secure shell (SSH) authentication protocol, 2006. IETF RFC 4252.
- [240] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 271–278. ACM, 2007.
- [241] F. Yvon. Rewriting the orthography of SMS messages. *Natural language Engineering*, 16(2):133–159, 2010.



- 
- [242] C. Zhang, H. Wang, Y. Liu, D. Wu, Y. Liao, and B. Wang. Automatic keyword extraction from documents using conditional random fields. *Journal of Computational Information Systems*, 4(3):1169–1180, 2008.
- [243] Y. Zhang, S. Vogel, and A. Waibel. Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? In *LREC*, 2004.
- [244] W. X. Zhao, R. Chen, K. Fan, H. Yan, and X. Li. A novel burst-based text representation model for scalable event detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. ACL, 2012.

# Appendix A

## Data set for FAQ information retrieval system

### English FAQ Sample

1. What is HIV?
2. What is AIDS?
3. How is HIV passed on?
4. How does the HIV test work?
5. Where can I get tested?
6. What is the window period?
7. Can you treat HIV?
8. Is it still possible to have sex and relationships if I have HIV?
9. What are my responsibilities as an HIV infected person?
10. Is there a risk to my own health in having sex?
11. Are there risks to others?
12. Do you think I should tell people I am HIV positive?
13. What are sexually transmitted infections (STIs) ?
14. What should I know about STIs ?
15. Where can I ask for treatment information ?
16. How do I know if I am infected ?
17. How can I avoid STIs ?
18. How common are genital warts ?
19. What about genital warts ? Will they stay until I have the right medication ?
20. I have an abnormal or unusual discharge from my vagina. What could it be ?
21. I have an abnormal discharge from my penis. What is it ?
22. Why are these sore like things on my vagina ?
23. Tell me, can I get any sexual infection if I use a condom ?
24. How risky is it to have oral sex ?
25. How will I know if I caught anything last night since I had unprotected sex ?

**SMS FAQ Sample 1**

1. Wats hiv?
2. Wats aids?
3. Hws hiv pasd on?
4. Hw the hiv test wrk?
5. Whr cn i gt testd?
6. Wats d window period?
7. Cn u treat hiv?
8. Is it stil posibl 2have sex n relationships if i hv hiv?
- 9 Wat r my responsibilities as an hiv infctd prsn?
10. Is thr a risk2ma own health in havin sex?
11. R ther risks to others?
12. Do u thnk i shud tel ppl im hiv positiv?
13. Wat r sexually transmitd infections(sti's)?
14. Wat shud i knw bwt sti's?
15. Whr cn i ask 4treatmnt info?
16. Hw do i knw if im infctd?
17. Hw cn i avoid sti's?
18. Hw comon r genital warts?
19. Wat bwt genital warts? Wil thy stay until i hav d ryt meds?
20. I gt an abnorml or unusual discharge frm ma vagina. Wat cud it b?
21. I gt an abnrmal discharge frm ma penis. Wat is it?
22. Y r these sore like thngs on ma vagina?
23. Tel me, cn i gt any sexual infction if i use a cndm?
24. Hw risky is it 2hav oral sex?
- 25 Hw wil i knw if i caught nethin last nyt since i had unprotctd sex?

**SMS FAQ Sample 2**

1. Whats hiv,
2. whats aids,
3. hows hiv psd on,
4. hw dus de hiv tst wrk,
5. whr cn i gt tstd,
6. wat is de wndow priod,
7. cn u trt hiv,
8. is it stl psbl 2 hv sex nd rltshps if i hiv,
9. Wat r my rspnsblties as an hiv infctd prsn,
10. is de a risk 2 ma own hlth in hvng sex,

11. r der rsk 2 adars,
12. du u thnk i shud tel ppl im hiv pstve,
13. wat r sxually trnsmid infctons,
14. wat sud i no abt STIs,
15. whr cn i ask 4 trtmnt infrmton,
16. hw du i no if im infctd,
17. hw cn i avoid STIs,
18. hw cmon r genitls warts,
19. wat abt genital warts? wil dey sty until i hv de rite medction,
20. i hv an abnorml or unusual dschrge 4rom ma vagna. wat cud it be,
21. i hv an abrnmal dschrge 4rom ma penis. wat is it,
22. y r dis sore lyk thngs on ma vagina,
23. tl me cn i get any sxual infctons if i use a condm,
24. hw risky it is 2 hv oral sex,
25. hw wil i no if i caught anythng lst nyt snce i had unprtctd sex,

### SMS FAQ Sample 3

1. Wts HIV?
2. Wts 8s?
3. Hws HIV ssd on?
4. Hw daz HIV tst wk?
5. Whr cn I gt tstd?
6. Whts th windw period?
7. Cn u ts HIV?
8. Is t stl posbl 2 hv sex n relati0nshps if i hv HIV?
9. Wat r my respnsiblties as n HIV infctd prsn?
10. Is thr a rsk 2 my own helth in hvng sex?
11. R thr rsk 2 othrz?
12. D u thnk I shud tl ppl Im HIV poztv?
13. Wt r STIs? Wt shud I knw abt STIs?
14. Whr cn I ask 4 trtment info?
15. Hw d I knw f Im infctd?
16. Hw cn I avoid STIs?
17. Hw cmon r genital wats?
18. Wt abt genital wats?
19. Wl thy sty untl I hv th ryt medcati0n?
20. I hv n abn0mal dschag 4rm my vagina.Wt cud t b?
21. I hv abn0mal dschag 4rm my penis.Wt is t?

22. Y r thz sor lyk thngs on my vagina?
23. Tl m,cn I gt any sexual infcti0n f I uz a cndm?
24. Hw rsky is t 2 hv oral sex?
25. Hw wl I knw f I cot anythng lst nyt snc I hd unprtectd sex?

#### SMS FAQ Sample 4

1. Watz?HIV?
2. Watz AIDS?
3. Auz HIV pasd on?
4. Hw ds d HIV tst work?
5. Whr cn I gt testd?
6. Watz d window period?
7. Cn u trt HIV?
8. Itz stl psibl 2 hv sx n rltshps if i hv HIV?
9. Wt ar my resp as an HIV infctd pson?
10. Is dre a rsk 2 my own hlth in hvin sx?
11. Ar dre rsks 2 oda?
12. D u tnk I shd tel ppl im HIV+ve?
13. Wt ar sxually transnitd infectns(STIS0?
14. wt shld i knw abt STIs?
15. Wre cn I ask 4 trtmnt infmatn?
16. Au do I kno if im infctd?
17. Au cn I avoid STIs?
18. Au cmon ar gnital warts?
19. Wt abt gntal warts?Wil dey sty untl I hv d rit medctn?
20. Iv an abnmal or unusual discharg frm my vagna.wt cld it b?
21. Iv an abnml dschag frm my pnis?Wtz it?
22. wy ar ds sores lik tins on my vagna?
23. Tll me,cn I gt any sxual infctn if i us a cndm?
24. Au risky ist 2 hv oral sx?
25. Au wl I knw if I caut anytin lst nit since I hd unprotectd sx?

# Appendix B

## PHP code for SCORE and SMSql algorithms

---

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Function for datacollection, vowel extraction algorithm
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
<?php
ini_set('max_execution_time', 91200);
include_once './core/db.php';
$word = "example";
$extract = extractVowels($word);
dataStore($word, $extract);
echo "<p> $word => $extract";
dataCollection();
function dataCollection() {
    $arrayColonm = array('a', 'b', 'c', 'd', 'e', 'f', 'g',
                        'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p',
                        'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z');
    foreach ($arrayColonm as $col) {
        $qry = " select $col ";
        $qry .= " from english_and_hiv ";
        $qry .= " limit 1700 ";
        $result = @databaseConnection::getConn($qry);
        while ($row = @mysqli_fetch_row($result)) {
            $word = $row[0];
            $extract = extractVowels($word);
            dataStore($word, $extract);
        }
    }
    $qry = " delete FROM 'test1' WHERE 'org_word' like '' ";
    @databaseConnection::getConn($qry);
    echo "<p> I'm done my job </p>";
}

function extractVowels($word) {
    $extracted = "";
    $vowel_arr = array('a', 'e', 'i', 'o', 'u');
    $string = $word;
    $len = strlen($string);
    for ($i = 0; $i < $len; $i++) {
        if (in_array($string[$i], $vowel_arr)) {
            if ($i == 0)
                $extracted .= $string[$i];
        } else {

```

```

        $extracted .= $string[$i];
    }
}
return $extracted;
}

function dataStore($word, $extract) {
    $qry = " insert into test1 (org_word,ext_word) " .
        " values('$word','$extract') ";
    databaseConnection::getConn($qry);
}
?>

```

---

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Function for Abbreviation, homophone, Frequently_used_sms
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
<?php

function getArrayWord($word) {
    $arrReturn = array();
    $arrChar = str_split($word);
    $col = $arrChar[0];
    $element = "";
    foreach ($arrChar as $value) {
        $element .= $value . "%";
    }
    $qry = " select $col ";
    $qry .= " from english_and_hiv ";
    $qry .= " where $col like '$element' ";
    $result = @databaseConnection::getConn($qry);
    while ($row = @mysqli_fetch_row($result)) {
        $arrReturn[] = $row[0];
    }
    return $arrReturn;
}

function Abreviation($word) {
    $findWord = null;
    $qry = " select meaning ";
    $qry .= " from acronyms_and_abbreviations ";
    $qry .= " where word like '$word' ";
    $result = @databaseConnection::getConn($qry);
    while ($row = @mysqli_fetch_row($result)) {
        $findWord = $row[0];
        break;
    }
    return $findWord;
}

function isHomophone($word) {
    $boo = false;
    $parts = preg_split("/(,?\s+)|((?<=[a-z])(?=\d)|((?<=\d)(?=[a-z]))/i", $word);
    foreach ($parts as $element) {
        if (is_numeric($element)) {
            $boo = true;
        }
    }
    return $boo;
}

```

```

}
function isPreposition($word) {
    $boo = false;
    $qry = " select meaning ";
    $qry .= " from preposition ";
    $qry .= " where word like '$word' ";
    $result = @databaseConnection::getConn($qry);
    while ($row = @mysqli_fetch_row($result)) {
        $boo = true;
    }
    return $boo;
}

function Punctuation($word) {
    $arrWord = null;
    $qry = " select meaning ";
    $qry .= " from punctuation ";
    $qry .= " where word like '$word' ";
    $result = @databaseConnection::getConn($qry);
    while ($row = @mysqli_fetch_assoc($result)) {
        $arrWord = $row['meaning'];
    }
    return $arrWord;
}

function frequently_used_smswords($word) {
    $arrA = null;
    $qry = " select meaning ";
    $qry .= " from frequently_used_smswords ";
    $qry .= " where word like '$word' limit 1 ";
    $result = @databaseConnection::getConn($qry);
    while ($row = @mysqli_fetch_row($result)) {
        $arrA = $row[0];
    }
    return $arrA;
}

function Homophone($word) {
    $parts = preg_split("/(,?\s+)|(((?<=[a-z])(?=\d))
        |(((?<=\d)(?=[a-z]))/i", $word);
    $final = "";
    foreach ($parts as $element) {
        if (is_numeric($element)) {
            $findValue = getHomophonesSQL($element);
            $final .= $findValue;
        }
        else{
            $final .= $element;
        }
    }
    return $final;
}

function getHomophonesSQL($word) {
    $arrA = "";
    $qry = " select meaning ";
    $qry .= " from homophone ";
    $qry .= " where word like '$word' limit 1 ";

    $result = databaseConnection::getConn($qry);

```



```

    while ($row = mysqli_fetch_row($result)) {
        $arrA = $row[0];
    }
    return $arrA;
}
?>

```

---

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Repeated character deletion algorithm
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
<?php

```

```

function checkRepeatedLetter($string) {
    $new_string = '';
    $starting_char = 0;
    while (strlen($string) > 0 && $starting_char < strlen($string))
    {
        $blah = preg_match('/[a-zA-Z0-9]{1,}/', $string, $matches);
        $letter = $matches[0][$starting_char];
        $new_string .= $letter;
        $regex = '/' . $letter . '{3,}/';
        $string = preg_replace($regex, $letter, $string);
        $starting_char++;
    }
    echo $new_string;
    return $new_string;
}
?>

```

---

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Service connection session
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
<?php

```

```

include_once 'connection.php';
if (session_id() == null) {
    session_start();
}

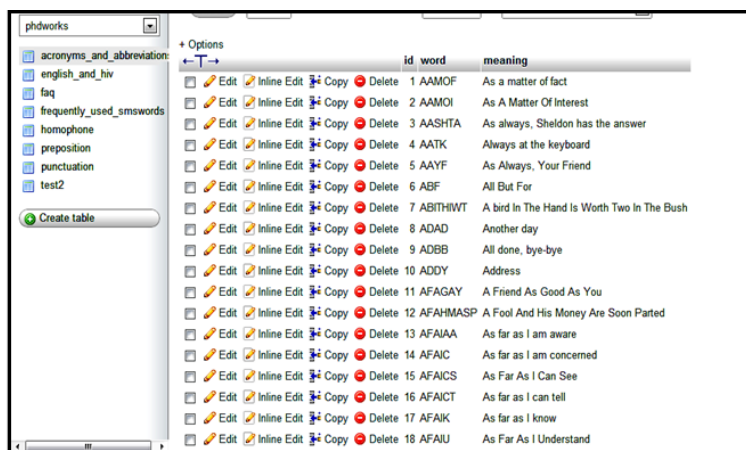
echo "<h2>--Seach Find (" . json_encode($_SESSION
    ['feedbackSize']). ") ---</h2>";
$qry = " select * from storesearch ";
$result = @databaseConnection::getConn($qry);
while ($row = @mysqli_fetch_assoc($result)) {
    echo "<br><br><a href='#'>".json_encode($row['words'])."</a>";
}
echo "<p>End of result </p>";
?>

```

---

# Appendix C

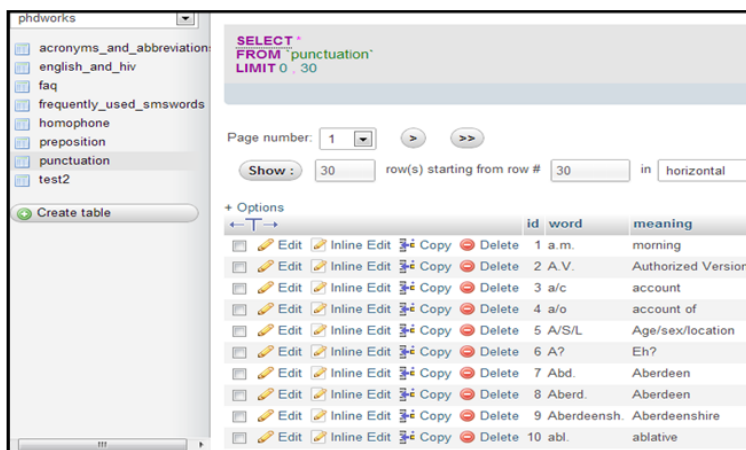
## Data structure of other modules



The screenshot shows a database interface with a table named 'acronyms\_and\_abbreviation'. The table has three columns: 'id', 'word', and 'meaning'. The data is as follows:

id	word	meaning
1	AAMOF	As a matter of fact
2	AAMOI	As A Matter Of Interest
3	AASHTA	As always, Sheldon has the answer
4	AATK	Always at the keyboard
5	AAYF	As Always, Your Friend
6	ABF	All But For
7	ABTHIWT	A bird In The Hand Is Worth Two In The Bush
8	ADAD	Another day
9	ADBB	All done, bye-bye
10	ADDY	Address
11	AFAGAY	A Friend As Good As You
12	AFAHMASP	A Fool And His Money Are Soon Parted
13	AFAIAA	As far as I am aware
14	AFAIC	As far as I am concerned
15	AFAICS	As Far As I Can See
16	AFACT	As far as I can tell
17	AFAIK	As far as I know
18	AFAIU	As Far As I Understand

FIGURE C.1: Acronyms/abbreviation



The screenshot shows a database interface with a table of punctuation and prepositions. The table has three columns: 'id', 'word', and 'meaning'. The data is as follows:

id	word	meaning
1	a.m.	morning
2	A.V.	Authorized Version
3	a/c	account
4	a/o	account of
5	A/S/L	Age/sex/location
6	A?	Eh?
7	Abd.	Aberdeen
8	Aberd.	Aberdeen
9	Aberdeensh.	Aberdeenshire
10	abl.	ablative

FIGURE C.2: Punctuation/preposition

SELECT \* FROM 'homophone' LIMIT 0, 30

Page number: 1

Show: 30 row(s) starting from row # 30 in

id	word	meaning
1	0	Zero
2	1	one
3	2	its
4	3	three
5	4	for
6	5	five
7	6	six
8	7	seven
9	8	ate
10	9	nine

FIGURE C.3: Homophone table

id	a	b	c	d	e	f	g	h	i	j	k	l
1	atack	bubble	cabal	dabble	eager	fabric	gastardne	haberdasher	illex	jabber	kaffan	labo
2	abacus	babe	cabaret	dace	eagle	fabricate	gabble	habit	ibis	jabru	kaiser	labi
3	abalone	babel	cabbage	dachshund	ear	fabulous	gable	habitable	ice	jacaranda	kak	labo
4	abandon	babes	caber	dad	earl	facade	gasl	habitat	ichtthyology	jack	Kalashnikov	labo
5	abase	baboon	cabin	daddy	early	face	gaffly	habitual	iccicle	jackal	kale	labo
6	abashed	baby	cabinet	dado	earn	facebook	gadget	habituate	icing	jackaroo	kaledoscope	labo
7	abate	baccarat	cable	daffodil	earnest	facet	Gael	hacienda	icon	jackass	kamikaze	labo
8	abattair	bacchanalia	caboodle	fiat	ears	facetious	gaff	hack	iconoclast	jackboot	kangaroo	labo
9	abbess	bach	cabriolet	dag	earth	facia	gaffe	hacker	idea	jackdaw	kasin	laby
10	abbey	bachelor	cacao	dagga	earnig	facial	gaffer	hackles	ideal	jacket	kappok	lace
11	abbot	bacillus	cache	dagger	esse	facie	gag	hackney	idem	jackknife	kappo's	lace
12	abbreviate	back	cachet	daguerstyp	easel	facie	gaga	hackneyed	identical	jackpot	kaput	lach

FIGURE C.4: English and medical words

# Appendix D

## Results of annotators

Annotator 1

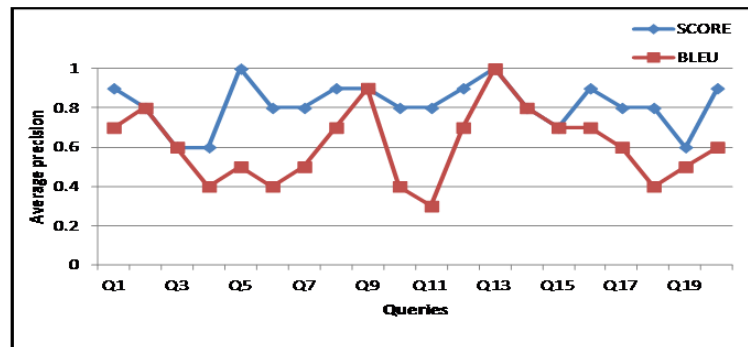


FIGURE D.1: Average precision for query in Bin 1: BLEU and SCORE

Annotator 2

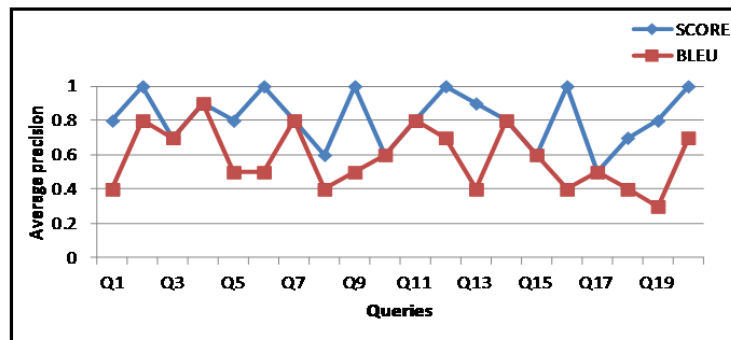


FIGURE D.2: Average precision for query in Bin 2: BLEU and SCORE

Annotator 3

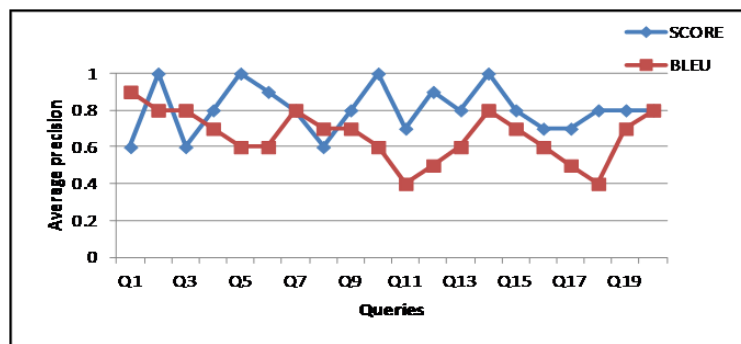


FIGURE D.3: Average precision for query in Bin 3: BLEU and SCORE

## Annotator 4

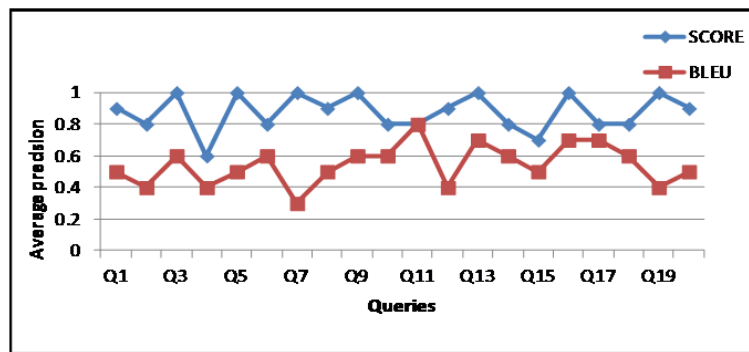


FIGURE D.4: Average precision for query in Bin 4: BLEU and SCORE

## Annotator 5

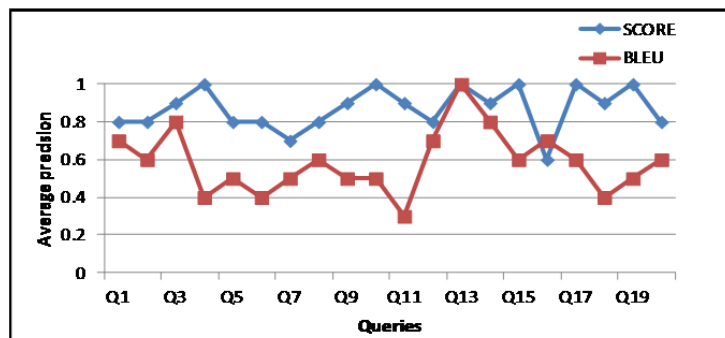


FIGURE D.5: Average precision for query in Bin 5: BLEU and SCORE

## Annotator 6

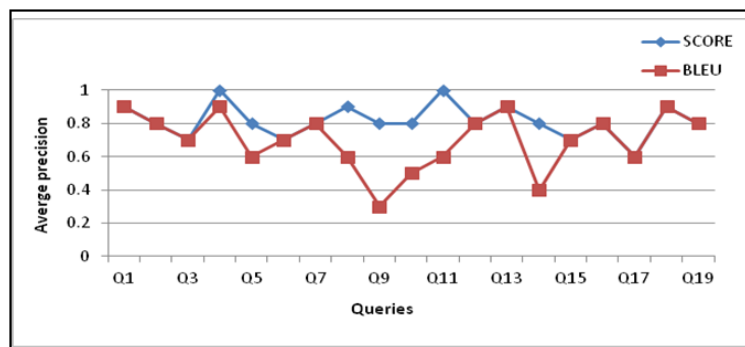


FIGURE D.6: Average precision for query in Bin 6: BLEU and SCORE

## Annotator 7

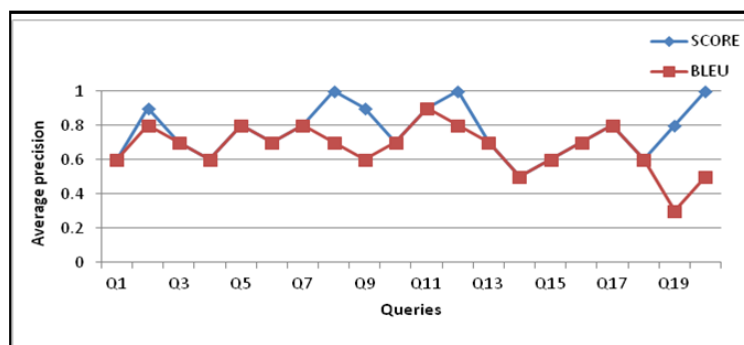


FIGURE D.7: Average precision for query in Bin 7: BLEU and SCORE

## Annotator 8

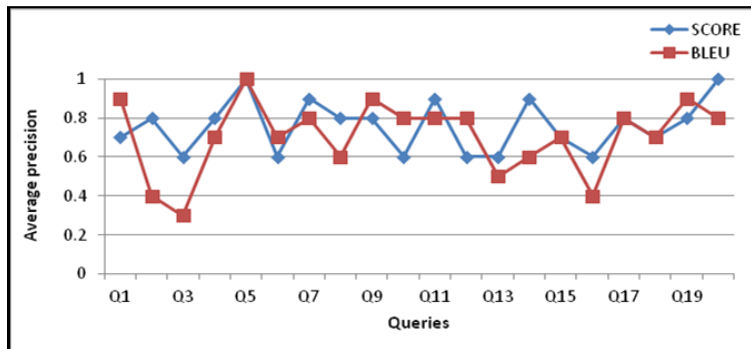


FIGURE D.8: Average precision for query in Bin 8: BLEU and SCORE

## Annotator 9

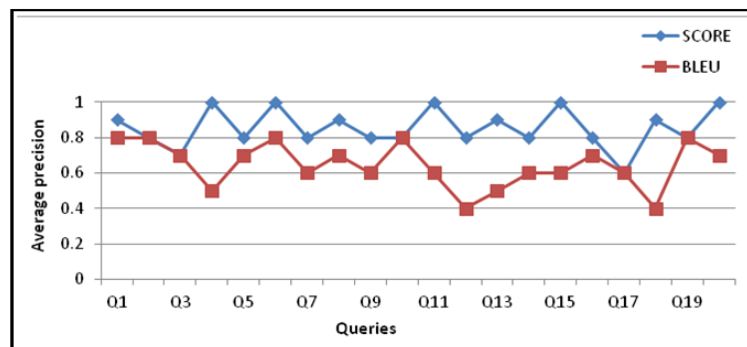


FIGURE D.9: Average precision for query in Bin 9: BLEU and SCORE

## Annotator 10

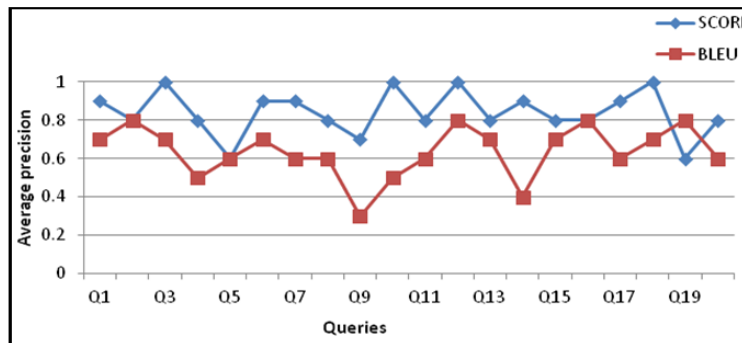


FIGURE D.10: Average precision for query in Bin 10: BLEU and SCORE

# Appendix E

## HIV/AIDS websites

FAQs on HIV/AIDS were collected from the following websites

1. <http://www.aids.gov/hiv-aids-basics/>
2. <https://actgnetwork.org/>
3. <http://www.webmd.com/hiv-aids/news/20101215/hiv-aids-cure-faq>
4. <http://www.symptomsofhiv.co.za/symptoms-of-hiv.php>
5. <http://www.info.gov.za/faq/aids.htm>
6. <http://www.halton.ca/cms/one.aspx?pageId=11097>
7. <http://aids.gov/frequently-asked-questions/>
8. <http://www.cdc.gov/hiv/resources/qa/index.htm>
9. <http://www.kingcounty.gov/healthservices/health/communicable/hiv/resources/testing.aspx>
10. <http://www.questioningaids.com/?page-id=11>
11. <http://www.cdcnpin.org/scripts/hiv/faq.asp>
12. <http://www.aids-india.org/faq.htm>
13. <http://www.medicinenet.com/script/main/art.asp?articlekey=123582>
14. <http://www.lifepositive.com/body/body-holistic/aids/aids-faq.asp>
15. <http://hivinsite.ucsf.edu/inSite?page=basics-00-00>
16. <http://www.who.int/tb/challenges/hiv/faq/en/>
17. <http://www.baangerda.org/en/FAQ.html>
18. <http://www.healthy.arkansas.gov/programsServices/infectiousDisease/hivStdHepatitisC/Pages/HIVAIDS.aspx>