Establishing the Reliability of Natural Language Processing Evaluation through Linear Regression Modelling

E.R. Eiselen

Thesis submitted for the degree Doctor of Philosophy in Linguistics and Literary Theory at the Potchefstroom campus of the North-West University

Supervisor: Prof G.B. van Huyssteen

Assistant Supervisor: L.T. Hickl

September 2013

ABSTRACT

Determining the quality of natural language applications is one of the most important aspects of technology development. There has, however, been very little work done on establishing how well the methods and measures represent the quality of the technology and how reliable the evaluation results presented in most research are. This study presents a new stepwise evaluation reliability methodology that provides a step-by-step framework for creating predictive models of evaluation metric reliability that take into account inherent evaluation variables. These models can then be used to predict how reliable a particular evaluation will be prior to doing an evaluation, based on the variables that are present in the evaluation data. This allows evaluators to predict the reliability of the evaluation prior to doing the evaluation and adjusting the evaluation data to ensure reliable results. Furthermore, this permits researchers to compare results when the same evaluation data is not available.

The new methodology is firstly applied to a well-defined technology, namely spelling checkers, with a detailed discussion of the evaluation techniques and statistical procedures required to accurately model an evaluation. The spelling checker evaluations are investigated in more detail to show how individual variables affect the evaluation results. Finally, a predictive regression model for each of the spelling checker evaluations is created and validated to verify the accuracy of its predictive capability.

After performing the in-depth analysis and application of the stepwise evaluation reliability methodology on spelling checkers, the methodology is applied to two more technologies, namely part of speech tagging and named entity recognition. These validation procedures are applied across multiple languages, specifically Dutch, English, Spanish and Iberian Portuguese. Performing these additional evaluations shows that the methodology is applicable to a broader set of technologies across multiple languages.

Keywords: Evaluation; Methodology; Natural language processing; Reliability; Regression modelling.

OPSOMMING

Een van die belangrikste aspekte van programmatuurontwikkeling vir natuurliketaalprosessering is die bepaling van die kwaliteit van sodanige programmatuur. Tot op hede is daar nog min navorsing gedoen oor hoe goed die onderskeie metodes en metrieke die kwaliteit van die tegnologie weerspieël en hoe betroubaar die evaluasieresultate van verskillende navorsers werklik is. Hierdie tesis ontwikkel 'n nuwe, stapsgewyse metodologie wat die betroubaarheid van evaluasies bepaal deur die ontwikkeling van statistiese modelle wat die betroubaarheid van ’n evaluasie beskryf op grond van die veranderlikes in die evalueringsproses. Hierdie modelle kan gebruik word om te voorspel hoe betroubaar 'n bepaalde evaluasie gaan wees alvorens die evaluasie uitgevoer word en laat die evalueerder toe om die evaluasiedata aan te pas sodat evaluasieresultate meer betroubaar sal wees. Die voorgestelde metodologie laat navorsers ook toe om resultate te vergelyk wanneer dieselfde evaluasiedata nie vir alle navorsers beskikbaar is nie.

Die nuwe metodologie word eers op 'n goedgedefinieërde tegnologie, naamlik speltoetsers, toegepas. Deel van hierdie toepassing is 'n gedetailleerde bespreking van die evaluasietegnieke en statistiese prosedures wat nodig is om die evaluasiemodellering te doen. Speltoetserevaluasies word in detail ondersoek ten einde die verhouding tussen individuele veranderlikes en bepaalde metrieke vas te stel, asook om die invloed van hierdie veranderlikes op die betroubaarheid van die metrieke te bepaal. Vervolgens word voorspellende regressiemodelle vir elk van die speltoetserevaluasiemetrieke daargestel en word die akkuraatheid van die modelle geverifieer om te bewys dat die modelle wel goeie voorstellings van die betroubaarheid van die metrieke is.

Na die deeglike toepassing en analise van die stapsgewyse metodologie vir evalusiebetroubaarheid op speltoetsers, word die metodologie op twee ander tegnologieë, te wete woordsoortetikettering en benoemde-entiteitherkenning, toegepas. Die metodologie word terselfdertyd ook op verskillende tale, naamlik Nederlands, Engels, Portugees en Spaans, toegepas, met die doel om te bewys dat hierdie metodologie toepaslik is op ’n wyer stel tale en tegnologieë.

Kernwoorde: Evaluasie; Metodologie; Natuurliketaalprosessering; Betroubaarheid; Regressiemodellering.

ACKNOWLEDGEMENTS

I would like to extend my sincere thanks and gratitude to the following people who helped me through the process of completing this work. I appreciate your patience and support during the times I worked on this.

Sansi

Sieg, Uca en Andreas

Gerhard van Huyssteen

The Senekal family

Martin Puttkammer

Suléne Pilon

Ulrike Janke

Lara Taylor-Hickl

Dermot McLoughlin

Vernon Southward

Kristina Toutanova

Centre for Text Technology

Research Unit for Languages and Literature in the South African Context

TABLE OF CONTENTS

1. Introduction 1

1.1 Contextualisation 1
1.2 Problem statement 6
1.3 Research questions 9
1.4 Research goals 9
1.5 Research methodology 10
1.5.1 Evaluation reliability 10
1.5.2 Case study: scoping a spelling checker evaluation 11
1.5.3 Case study: modelling evaluation reliability for spelling checkers 11
1.5.4 Validating the stepwise evaluation reliability methodology 12
1.6 Deployment 13

2. NLP evaluation and reliability 15

2.1 Introduction 15
2.2 NLP evaluation reliability 15
2.2.1 Statistical bounds of similarity 17
2.2.2 Reliability and quality of results 18
2.2.3 Methodology or metric inadequacy 18
2.3 Stepwise evaluation reliability methodology 19
2.3.1 Step 1: Define the purpose 20
2.3.2 Step 2: Define the type 21
2.3.3 Step 3: Define the method 24
2.3.3.1 Gold-standard evaluations 25
2.3.3.2 Post-computational judgement 26
2.3.4 Step 4: Collect the data 28
2.3.5 Step 5: Identify possible variables 35
2.3.6 Step 6: Identify relevant metrics 36
2.3.7 Step 7: Validate the metric 38
2.3.7.1 Theoretical validation of metrics 38
2.3.7.2 Empirical validation of evaluation metrics 40
2.3.8 Step 8: Select independent variables for the metric 43
2.3.9 Step 9: Select and verify reliability model 47
2.4 Conclusion 49

3. Case study: scoping a spelling checker evaluation 52

3.1 Introduction 52
3.2 Scoping a spelling checker evaluation 53
3.2.1 Step 1: Define the purpose 53
3.2.2 Step 2: Define the type 55
3.2.3 Step 3: Define the method 55
3.2.4 Step 4: Collect the data 57
3.2.5 Step 5: Identify possible variables 60
3.2.5.1 Methodological variables 62
3.2.5.1.1 Test suites vs. test corpora 62
3.2.5.1.2 Usage-based vs. automatically generated data 65
3.2.5.1.3 Types vs. tokens 68
3.2.5.2 Data-attribute variables 70

3.2.5.2.1 Modelling unit size ... 70
3.2.5.2.2 Type-to-token ratio ... 71
3.2.5.2.3 Heterogeneity ... 71
3.2.5.2.4 Target density ... 72
3.2.5.2.5 Error types ... 72
3.2.5.2.6 Quality of the technology ... 73
3.2.6 Step 6: Identify relevant metrics ... 74
3.3 Conclusion ... 78

4. Case study: modelling spelling checker evaluation reliability ... 80

4.1 Introduction ... 80
4.2 Modelling evaluation reliability: recall correct ... 81
4.2.1 Step 7: Validate recall correct ... 81
4.2.2 Step 8: Select independent variables for R_c ... 82
4.2.2.1 Hypotheses ... 84
4.2.2.2 Data-attribute experiments ... 85
4.2.2.3 Methodological variable experiments ... 91
4.2.3 Step 9: Select and verify reliability model for R_c ... 94
4.3 Modelling evaluation reliability: recall incorrect ... 96
4.3.1 Step 7: Validate recall incorrect ... 96
4.3.2 Step 8: Select independent variables for R_i ... 97
4.3.2.1 Hypotheses ... 99
4.3.2.2 Experiments ... 99
4.3.3 Step 9: Select and verify reliability model for R_i ... 104
4.4 Modelling evaluation reliability: precision incorrect ... 105
4.4.1 Step 7: Validate precision incorrect ... 106
4.4.2 Step 8: Select independent variables for P_i ... 106
4.4.2.1 Hypotheses ... 108
4.4.2.2 Experiments ... 109
4.4.3 Step 9: Select and verify reliability model for P_i ... 114
4.5 Modelling evaluation reliability: precision correct ... 116
4.5.1 Step 7: Validate precision correct ... 116
4.5.2 Step 8: Select independent variables for P_c ... 116
4.5.2.1 Hypotheses ... 118
4.5.2.2 Experiments ... 118
4.5.3 Step 9: Select and verify reliability model for P_c ... 119
4.6 Modelling evaluation reliability: suggestion adequacy ... 121
4.6.1 Step 7: Validate suggestions adequacy ... 121
4.6.1.1 Percentile SA ... 122
4.6.1.2 Scoring systems ... 123
4.6.2 Step 8: Select independent variables for SA ... 128
4.6.2.1 Hypotheses ... 129
4.6.2.2 Experiments ... 129
4.6.3 Step 9: Select and verify reliability model for SA ... 131
4.7 Modelling evaluation reliability: overall linguistic adequacy metrics ... 133
4.7.1 Step 7: Validate metrics ... 134
4.7.1.1 Predictive accuracy ... 134
4.7.1.2 F-measures ... 134
4.7.1.3 Mean time between failures ... 136
4.7.1.4 False instances per page ... 137
4.7.2 Step 8: Select independent variables for combinatory metrics ... 138

4.7.2.1 Predictive accuracy ... 138
4.7.2.2 fm_i ... 139
4.7.2.3 fm_c ... 140
4.7.2.4 FIPP ... 141
4.7.3 Step 9: Select and verify reliability models for combinatory metrics ... 142
4.7.3.1 Predictive accuracy ... 142
4.7.3.2 fm_i ... 143
4.7.3.3 fm_c ... 144
4.7.3.4 FIPP ... 145
4.8 Conclusion ... 147

5. Validating the stepwise evaluation reliability methodology ... 152

5.1 Introduction ... 152
5.2 Validation: multiple technologies and languages ... 153
5.3 Methodology validation: part of speech taggers ... 155
5.3.1 Step 4: Collect the data ... 155
5.3.2 Step 5: Identify possible variables ... 157
5.3.3 Step 6: Identify relevant metrics ... 160
5.3.4 Step 7: Validate the metrics ... 161
5.3.5 Step 8: Select independent variables ... 161
5.3.6 Step 9: Select and verify the reliability model ... 162
5.4 Methodology validation: Named entity recognisers ... 163
5.4.1 Step 4: Collect the data ... 164
5.4.2 Step 5: Identify possible variables ... 164
5.4.3 Step 6: Identify relevant metrics ... 166
5.4.4 Step 7: Validate the metrics ... 167
5.4.5 Step 8: Select independent variables ... 167
5.4.6 Step 9: Select and verify the reliability model ... 169
5.5 Conclusion ... 171

6. Conclusion ... 175

6.1 Summary ... 175
6.2 Contributions ... 180
6.3 Future work ... 181

Addendum A. Statistical approaches to data interpretation ... 182

Addendum B. Afrikaans error generation module ... 189

Reference list ... 192

TABLE OF FIGURES

Figure 2.3.1: Technology life cycle and evaluation (adapted from Hirschman & Mani, 2003:415) 23
Figure 2.3.2: Functional relation: $y = 2x$ 42
Figure 2.3.3: Statistical relation: precision and error percentage 42
Figure 3.2.1: A sample set of English words with valid words grouped together 75
Figure 3.2.2: Hypothetical response of a spelling checker on a sample set of English words 76
Figure 4.2.1: Visual representation of R_c 82
Figure 4.2.2: σ of Afrikaans spelling checkers on texts of differing sizes 86
Figure 4.2.3: TTR for Afrikaans stratified corpora 89
Figure 4.2.4: Relationship between TTR and R_c for Afrikaans corpora 89
Figure 4.3.1: Visual representation of R_i 97
Figure 4.4.1: P_i of SC B on 1,000 word texts with error percentages ranging from 1% to 10% 110
Figure 4.4.2: The effect of error percentage normalisation to 5% on P_i scores of SC B 113
Figure 4.4.3: The effect of error percentage normalisation to 1% on P_i scores for SC B 114
Figure 5.3.1 Dutch POS tagger precision on different modelling unit size 156
Figure 5.3.2: English POS tagger precision on different modelling unit sizes 157
Figure 5.3.3 Iberian Portuguese POS tagger precision on different modelling unit sizes 157
Figure 6.3.1: Probabilities as areas under a probability density curve with a normal distribution (Devore & Peck, 1986) 183
Figure 6.3.2: F curve and F critical value 187

LIST OF ACRONYMS

Acronyms	Meaning
ACE	Automatic content extraction
ALPAC	Automatic language processing advisory committee
BLEU	Bilingual evaluation understudy
BNC	British national corpus
CLEF	Cross-language evaluation forum
CoNLL	Conference on natural language learning
DiET	Diagnostic and evaluation tools for natural language processing applications
EAGLES	Expert advisory group on language evaluation systems
ELSE	Evaluation of language and speech engineering
FFPP	False flags per page
FIPP	False instances per page
FSRP	Forward stepwise regression procedure
IE	Information extraction
IR	Information retrieval
ISO	International standards organisation
LREC	Language resource and evaluation conference
MEDAR	Mediterranean Arabic language and speech technology
METEOR	Metric for evaluation of translation with explicit ordering
MSE	Error mean square
MSPR	Mean squared prediction error
MSR	Regression mean square
MT	Machine translation
MTBF	Mean time between failures
NIST	National institute of science and technology
NLP	Natural language processing
POS	Part of speech
PTB	Parameterisable test bed
SemEval	Semantic evaluation
SGML	Standard generalised mark-up language
SSS	Simplified scoring system
TEMAA	A testbed study of evaluation methodologies: authoring aids
TER	Translation edit rate
TREC	Text retrieval conference
TSNLP	Test suites for natural language processing
TTR	Type-to-token ratio
XML	Extended mark-up language

Chapter 1

1. INTRODUCTION

1.1 CONTEXTUALISATION

In the field of natural language processing (NLP), resources are often limited to the research and development of new technologies, with the testing and evaluation of the technologies often coming as an afterthought (Hirschman & Mani 2003; Manzi *et al.*, 1996; Paggio & Underwood 1997; Palmer & Finin 1990). Only in the last two decades has there been a concerted effort by the NLP community to actively pursue the creation of structured methods and measurement instruments to test the functionality of new language technologies. The increased attention to testing and evaluation is most noticeable in the establishment of publications such as the Language Resources and Evaluation Journal, as well as conferences such as:

- Message understanding conference (MUC, 1987-1997)[1];
- Machine translation summit (MT Summit, 1987-Current)[2];
- Text retrieval conference (TREC, 1992-Current)[3];
- Language resource and evaluation conference (LREC, 1998-Current); and
- Semantic evaluation workshops (SemEval, 1998-Current).

There have also been various government-sponsored research initiatives that have focused almost exclusively on evaluation, such as:

- Expert advisory group for language engineering standards (EAGLES, 1995a, 1995b);
- A testbed study of evaluation methodologies: authoring aids (TEMAA, 1997);

---

[1] One of the primary aims of the MUC conferences was establishing an evaluation methodology for the evaluation of information extraction systems (Grishman & Sundheim, 1996; Sundheim, 1992).

[2] Although the MT Summit is concerned with all aspects of machine translation, one of the central themes for all summits has been improved methodologies for evaluating machine translation systems. Several developments in machine translation evaluation have originated at this conference (Turian *et al*, 2003).

[3] TREC is defined as "the information retrieval community's annual evaluation forum" and these conferences have individually placed focus on the evaluation procedures and methodologies applied to the information retrieval task (Voorhees & Harman, 2005).

- Diagnostic and evaluation tools for natural language processing applications (DiET, Netter *et al.*, 1998);
- Evaluation of language and speech engineering (ELSE, Lenci *et al.,* 1999); and
- Rouge (Lin, 2004).

Prior to the 1990s, only a small group of researchers and projects evaluated NLP software (ALPAC, 1966; Damerau, 1964; Van Rijsbergen, 1979), yet most of their evaluations reported only results, with few or no descriptions of the methodologies that were used (Galliers & Sparck-Jones, 1993:59). The heightened awareness of and emphasis on evaluation during the 1990s have led to some standardisation of both methods and measurement of some of the characteristic attributes associated with NLP technologies. Much of this work has focused on testing the attributes described in the International Standards Organization (ISO) 9126 standard of software engineering, and on applying these quality characteristics to the software developed for NLP (EAGLES, 1995a; King, 2005; TEMAA, 1997).

The evaluation of NLP technology is mainly concerned with two aspects, namely methodology and metrics (Paggio & Underwood, 1997:4). Methodology refers to a clear set of steps and procedures that constitute an evaluation. These procedures should be well defined in order to get reliable and repeatable evaluation results. Metrics refer to quantitative measures and descriptions of functionality by assigning numerical values to the attributes and sub-attributes of an NLP technology (EAGLES, 1995b:2; El-Emam, 2000:1). These metrics should be reliable and descriptive measurement instruments that make the comparative evaluation of different systems possible.

Problems with the standardisation and reliability of evaluation methodologies and measurement instruments used in NLP evaluations are commonplace (King, 1999; Koller *et al.*, 2009; Lavelli *et al.*, 2004; Madnani *et al.*, 2011; Netter *et al.*, 1998; Schwartz *et al.*, 2011; Turian *et al.,* 2003; Vanni & Reeder, 2000; Voorhees 2002; White & O'Connell, 1994). Because there are often inconsistencies in evaluation methods and results, there is an almost constant evolution of the evaluation strategies used to determine the quality of an application.

The earliest study of NLP technologies began with two fields of research, namely machine translation (MT) and information retrieval (IR). As early as 1933, French and Russian researchers independently developed storage devices that could find an equivalent word in another language (Hutchins, 1995), but it was not until a 1949 memo by Warren Weaver (cited by Hutchins, 1995; Lesk, 1996) that wider academic research into MT started. Similarly, the field of IR was essentially started with a paper by Vannevar Bush in 1945. Both research fields expanded during the 1950s and 1960s, with various universities developing solutions for MT and IR, and governments in the United States, Russia and Europe providing funds for the development of these systems. Although evaluations must have been done on these early systems, no record of any formal evaluations exists (Galliers & Sparck-Jones, 1993; Robertson, 2008).

The Automatic Language Processing Advisory Committee (ALPAC) report of 1966 (ALPAC, 1966) is widely considered the first major evaluation effort in the NLP community (Galliers & Sparck-Jones, 1993). The major government sponsors of MT in the United States studied the prospects of MT in 1964, with the establishment of ALPAC. The report highlights both the necessity of evaluation as a critical aspect of NLP research and development, and the effect an evaluation can have on a research field (Galliers & Sparck-Jones, 1993). In their widely criticised report, they concluded that MT was slower, less accurate and twice as expensive as human translations (Hutchins, 1995). Their conclusions prompted them to state "there is no immediate or predictable prospect of useful machine translation" (ALPAC, 1966:32). This directly caused a substantial decrease in NLP research undertaken in the United States for nearly a decade. The report also affected the establishment of evaluation procedures, because researchers were less forthcoming in explicitly measuring and reporting the quality of their technology for fear of losing more funding.

As Galliers & Sparck-Jones (1993) point out, even though IR and MT have been around for more than 60 years, during the first 40 years (until the late 1980s and early 1990s) very little research was done on evaluation methodologies. They add that although evaluations did take place, mention of the evaluation methods was negligible, and for the most part only results were reported. Prior to the early 1990s, evaluations were mainly carried out with specific interest groups in mind, and most of these evaluations were carried out confidentially with little or no data or results

being made public (Galliers & Sparck-Jones, 1993; Netter *et al.*, 1998:2). Even between fellow researchers, the exchange of evaluation results was limited. The evaluations during this period were often "tailor-made" for specific users/customers, and would therefore not be of much use to the wider NLP research community (Netter *et al.*, 1998:2).

The one exception to this lack of study in the field of evaluation was in IR, where complex and well-defined methods and metrics were described, tested and implemented on a wide scale before the 1990s. As far as evaluation is concerned, IR has been at the forefront of both the research into new methods and the proposal of standardised testing methodologies. During the late 1960s the IR community established the so-called "Cranfield paradigm" which, although evolving during the 1970s and 1980s, still forms the basis of many of the evaluations that are done in IR today (Cleverdon, 1967; Galliers & Sparck-Jones, 1993; Sasaki, 2007; Smrž, 2004; Voorhees, 2002). The paradigm sets out a simple methodology for evaluation that relies on a set of documents, a set of queries, and related relevance judgements and measurement based on Precision and Recall (Rasmussen, 2003). Work by Van Rijsbergen (1979) also helped to establish a wide-scale awareness of the importance of standardised methods and metrics for the accurate evaluation of developed technologies. The fact that these methodologies have been scrutinised and tested as much as they have, means that there is in-depth knowledge of the methods and metrics used, as well as the problems associated with their implementation. A number of the lessons learned in the development of IR methodologies can be applied to the wider field of NLP, and are discussed later in this study (cf. 3.2.1; Chapter 4).

Since the start of the 1990s, there has been a concerted effort to standardise evaluation methodologies and frameworks used in NLP. Most notable have been the establishment of various conferences geared specifically to evaluation or dedicating a substantial part to evaluation, as well as to publications focused exclusively on evaluation, such as the Language Resources and Evaluation Journal. The growing awareness of evaluation during the early 1990s led to the establishment of EAGLES in Europe, a group specifically tasked with proposing evaluation criteria and standards for NLP applications. New publications and conferences with a specific focus on evaluation were established, but much of the work remains in progress as

the field expands in both scope and complexity (Koller *et al.*, 2009; Liu & Ng, 2012; Schwartz *et al.*, 2011).

Another major problem for evaluators and evaluation groups is the fact that they usually compete for the same resources as other NLP projects, such as annotated data, funding, time, researchers and developers. Unfortunately, good evaluations are expensive (Hirschman & Mani, 2003; Netter *et al.*, 1998), but have very little monetary value after completion of the evaluation. NLP researchers would therefore rather spend money on new technologies than on developing good evaluation methodologies and strategies (Hirschman & Thompson, 1997:409; Netter *et al.*, 1998:4). This leads to evaluations being done as quickly and cheaply as possible. This situation has improved over the last couple of years, with wider use of crowdsourcing strategies (Ambati *et al.*, 2010; Zaidan & Callison-Burch, 2011), although the quality of the annotations from these sources is still under investigation, and for some tasks where expert knowledge is required, crowdsourcing does not reduce cost. However, with the standardisation efforts mentioned earlier, there has been an increasing tendency to be more open in sharing information, resources, and results of evaluations.

Over the last several years, most of the focus in the field of evaluation has been on establishing new evaluation metrics for more complex technologies and automatic evaluation, with very little or no human involvement (Lin, 2004; Liu & Ng, 2012; Murthy *et al.*, 2008, NIST, 2002; Papineni *et al.*, 2002; Rodrigo *et al.*, 2010; Snover *et al.*, 2006). Even though most of the newer metrics were proposed in the early to mid-2000s, the process of establishing these metrics as adequate and reliable measures of the functionality of these technologies is an on-going process (Amigó, 2011; Cohen *et al.*, 2012; Koller *et al.*, 2009; Volokh & Neumann, 2012). Many of these metrics are still being validated, both theoretically and empirically, to establish their validity, and during this process new variants of the metrics are proposed and must go through the same validation process.

Another focal point of evaluation methodology over the last couple of years is the focus on annotations through crowdsourcing and having the crowd directly evaluate the functionality of particular technologies, such as MT (Callison-Burch, 2009; Zaidan and Callison-Burch, 2010), error detection (Madnani *et al.*, 2011), word sense disambiguation (Akkaya et al., 2010), and named entity recognition (Finin et al.,

2010). Using crowds of untrained annotators to annotate complex data is a way to create evaluation data much more quickly and cheaply than traditional data set creation by experts. With this approach, multiple untrained people annotate the same data simultaneously. The annotations from the different individuals are then combined, to find the most likely annotation according to the majority of the crowd.

In the context of this background on evaluation, I will show that there are still various problems with the reliability of reported evaluation results. This is true especially for evaluations that aim to compare results to existing research, where the same evaluation data is not available. The following section provides more details about some of the problems with evaluation reliability, as additional background to the research questions and goals.

1.2 PROBLEM STATEMENT

King's (1998:1) description of NLP evaluation explains that an evaluation is the process of determining the effectiveness and quality of a product by using different qualitative and quantitative criteria. These criteria need to be a reliable reflection of both the linguistic adequacy and the usability of a product (EAGLES, 1995a; TEMAA, 1997). Furthermore the criteria should be used as an indication of any shortcomings in the product and a guide for future development activities (Netter *et al.,* 1998:1).

Even though most descriptions of evaluations stress the importance of being repeatable and reliable, relatively large discrepancies in results are often reported, sometimes even in the same article (Kim *et al.*, 2012; Madnani *et al.*, 2011), with little explanation for the source of the differences between evaluation results. This problem is exacerbated when evaluations are used to compare different systems or algorithmic approaches when the same evaluation data is not available.

Even with larger-scale efforts to establish evaluation methodologies and standards for NLP, such as those developed by the TEMAA (1997), ELSE (Lenci *et al.*, 1999) and the DiET projects (Netter *et al.*, 1998), evaluations still yield inconsistent results. This inconsistency can be attributed to a combination of three factors:

a) the methodology used in the evaluation;

b) the metrics that are used for evaluations; and

c) variables that are inherent in the evaluation procedure.

With regard to methodology, the lack of reference data (e.g. benchmarks and test suites) and the more or less arbitrary way in which evaluation material is collected for the evaluation of NLP applications, make the comparison between products almost impossible (Netter *et al.*, 1998:2). Although this situation has improved for languages that have large, richly annotated datasets, for many resource-scarce languages the acquisition of large-scale annotated data remains a major problem. By standardising the methods used to evaluate a specific application, the comparison of applications will be more reliable (ELSE, 1999).

There are a number of problems related to the evaluation methodologies that can result in evaluations that show variations in the results even when evaluating the same system using different methodological implementations, for instance the spelling checker results in Table 1. This is in part due to the lack of standardisation in the methods that are used, causing the methods to act as variables in the evaluation process and introducing volatility into the results. A better understanding and standardisation of methodological variables such as the following is central to creating reliable evaluations and are discussed at length in Chapters 3 and 4:

- using structured data in the form of test suites, automatically generated test data, or usage-based test corpora;
- the size of the test data;
- using word lists or running corpora; and
- the structure of the evaluation data.

The second aspect, metrics, relates to the quantitative measuring of attribute values to determine the adequacy of each attribute of a technology (TEMAA, 1997). EAGLES (1995b: 2) state that "…a metric is reliable inasmuch as it constantly provides the same result when applied to the same phenomena." This statement implies that for a metric to be accepted as a norm, it should return results that are always reflective of the actual functionality of the technology and can be validated by secondary testing under similar circumstances. Over the past four decades various new metrics for NLP evaluation have been proposed, with varying degrees of success and reliability (Galliers & Sparck-Jones, 1993; Lavie *et al.*, 2004; Paggio &

Underwood, 1997; Papineni *et al.*, 2002; Riley *et al.*, 2004; Sparck-Jones & Van Rijsbergen, 1976; Starlander & Popescu-Belis, 2002; TEMAA, 1997). As Reynaert (2006:146) points out, there are often discrepancies between different authors about what each of these metrics should measure, as well as the definition of each of the metrics. Even though reliability is a prerequisite, there is little or no research that has specifically focused on the testing of metric reliability, especially with regard to the influence of variables in the evaluation process.

As an example, Table 1 shows the results of the precision metric of an Afrikaans spelling checker on a number of different texts, where the number of words and the percentage of errors in the text differ. These are all variables that are not well defined in similar evaluations, but clearly have an influence on the results of this metric.

	Percentage of errors	**Precision**
Text 1	1%	18.60%
Text 2	5%	46.91%
Text 3	7%	63.64%
Text 4	2%	28.89%
Text 5	3%	43.64%
Text 6	11%	77.68%
Text 7	6%	45.83%

Table 1: Precision of a spelling checker on texts with different error percentages

Depending on which one of the evaluation results is reported, the precision result of the spelling checker can vary between 18.60% and 77.68%. This type of discrepancy is not acceptable if one is trying to determine the quality of the spelling checker or to compare two different spelling checkers. In addition, these results do not reflect a stable metric according to the definition given by EAGLES (1995a:2).

One of the major reasons for the variability in metric results is the fact that the evaluation does not consistently take the different variables of the evaluation into account. In the example represented in Table 1, the percentage of errors as well as the number of words in each of the texts differed, and this has a noticeable influence on the metric (as is shown more systematically in 4.4). Identifying and relating these variables to the metrics on which they have an influence is a central shortcoming of current evaluation procedures, and is one of the reasons that different evaluators often get differing results even when testing the same technology.

Even with this seemingly obvious shortcoming in evaluation methodology, almost no work has been done on determining the reliability of evaluations. One of the only reports on evaluation reliability is a technical note from the National Institute of Standards and Technology (NIST) on reporting the uncertainty of NIST measurement results (Taylor & Kuyatt, 1994).

Based on these observations, this study presents a generic stepwise methodology for identifying variables in NLP evaluation processes that are good predictors of evaluation reliability. These variables are used to create regression models that allow evaluators to predict the reliability of an evaluation by predicting the range of variability of 95% of future evaluations. This allows evaluators to determine whether their evaluation methods and metrics provide reliable evaluation results, and how to adjust their evaluations to ensure more reliable evaluations.

1.3 RESEARCH QUESTIONS

In order to address the identified evaluation problems, the following questions are posed:

a. What is NLP evaluation reliability?
b. How can evaluation reliability be modelled and predicted?
c. Which variables affect evaluation reliability and how are those variables identified?
d. How do variables affect evaluation reliability?
e. Does a linear regression modelling methodology for evaluation reliability prediction scale to multiple technologies and languages?

1.4 RESEARCH GOALS

From the research questions, the following objectives have been identified for this study:

a. To explore and define reliability for NLP evaluations.
b. To design a stepwise methodology for modelling evaluation reliability.
c. To provide a structured method for identifying variables that are predictors of evaluation reliability.

d. To provide a detailed investigation of how variables affect evaluation reliability.
e. To validate the stepwise evaluation reliability methodology by applying the methodology to multiple NLP technologies and languages.

In addition to these primary goals, the study also investigates how evaluations can be structured to minimise the associated costs. The focus here is on optimising evaluations, so that the smallest amount of data possible can be used to do reliable evaluations, thereby saving both time and money.

1.5 RESEARCH METHODOLOGY

The following methods describe the process followed to achieve the research goals.

1.5.1 EVALUATION RELIABILITY

The first part of this study focuses on determining exactly what evaluation reliability entails and how current best practices in NLP evaluation methodologies and metrics cause variability in the evaluation results. This discussion concludes with a working definition for evaluation reliability that is a combination of the current best practices in evaluation along with some new insights into the nature and sources of evaluation variability.

After establishing a definition for reliability, the rest of the first part of the study provides the outline of a stepwise methodology for ascertaining evaluation reliability. This outline provides the framework for the rest of the study and the evaluation procedure that allows linear regression modelling to be used in determining evaluation reliability. The stepwise methodology describes a two-phased evaluation approach of first scoping the evaluation and secondly modelling evaluation reliability. These two phases are the focus of the case study in the next two parts of the study.

1.5.2 CASE STUDY: SCOPING A SPELLING CHECKER EVALUATION

In the second part of the study, I apply the previously discussed stepwise evaluation reliability methodology in each of the two phases to a well-established and understood technology, namely spelling checkers. The scoping phase of the stepwise methodology entails:

1. defining the purpose of the evaluation;
2. defining the type of evaluation that needs to be performed;
3. explicating the methods that are used in the evaluation;
4. collecting and annotating the evaluation data;
5. identifying all of the possible variables that could affect evaluation reliability; and
6. identifying the metrics that are modelled in the procedure.

Each of these steps is described in detail along with extended examples of how the decisions correlate with existing evaluation best practices, while also allowing for the further implementation of regression procedures in the next part of the case study.

1.5.3 CASE STUDY: MODELLING EVALUATION RELIABILITY FOR SPELLING CHECKERS

The third part of this study implements the last three steps of the reliability methodology by performing evaluations on three different Afrikaans spelling checkers and using the results of these evaluations to create predictive regression models for each of the metrics identified in the last step of the scoping phase. In this phase of the stepwise methodology the following set of steps are applied to each metric:

1. validating the metrics;
2. selecting good variability predictor variables; and
3. selecting and verifying the predictive regression model.

In addition to the regression modelling and validation, additional experiments are carried out to provide a better understanding of how the evaluation variables that form part of the regression model for each metric affect the reliability of the evaluations.

1.5.4 VALIDATING THE STEPWISE EVALUATION RELIABILITY METHODOLOGY

In the final part of this study, the investigation and findings from the previous chapter are applied to a broader set of technologies and multiple languages to provide proof that the proposed methodology is a valid procedure for determining and predicting evaluation reliability. The validation of the methodology focuses on applying it to two different technologies, part of speech (POS) tagging and named entity (NE) recognition, as well as multiple languages, namely Dutch, English, Iberian Portuguese, and Spanish[4]. This extension provides a complete overview of the methodology by providing a step-by-step description of the methods and how they are applied to different technologies.

The described stepwise evaluation reliability methodology is then applied to POS taggers for Dutch, English, and Iberian Portuguese, in order to identify the evaluation variables and to create a regression model that can predict the variability of an evaluation. As before, this model goes through a verification step to prove that the model does accurately predict evaluation variability when applied to unseen evaluations.

Lastly, the same procedure is applied to three different English and Spanish named entity recognition systems to create and validate the predictive regression model. These two sets of experiments prove that the proposed methodology and modelling are valid procedures for predicting evaluation reliability for individual evaluations.

[4] The reason for selecting these languages for the different technologies is discussed in 5.2.

1.6 DEPLOYMENT

In Chapter 1 a brief overview has been provided of the current state of evaluation to determine the focus and area of investigation of this study by providing NLP evaluation background and identifying problematic areas in the field of evaluation reliability. Based on the background, five central research questions were presented that determine the aims of the study. Finally, the methods that are used to accomplish these aims were briefly discussed.

Chapter 2 gives an overview of the issues around variability and reliability, culminating in a working definition of evaluation reliability. Based on the discussion and definition of reliability, section 2.3 describes the framework of a stepwise methodology for ensuring evaluation reliability. This methodology forms the basis for further discussion and experimentation for creating reliable evaluations that are performed in the following chapters.

Chapter 3 is the first part of a case study that implements the methodology described in the previous chapter. This chapter describes the application of the first phase, evaluation scoping, to spelling checkers. This application provides a good example of a well-understood technology that shows differences between different evaluation approaches as well as providing examples of validating evaluation metrics.

Chapter 4 is a continuation of the case study in which the second phase of the stepwise evaluation reliability methodology, reliability modelling, is applied to spelling checkers. This case study provides an in-depth description of the procedures for identifying evaluation variables, modelling reliability based on these variables, and validating the reliability model. This process is repeated for each evaluation metric relevant to spelling checker evaluation, and provides the initial proof of concept for the stepwise evaluation reliability methodology. In addition to the modelling, additional experiments are performed for each metric to provide additional understanding of the relationship between the variables and evaluation reliability.

In Chapter 5 the results from Chapter 4 are used to validate the new stepwise evaluation reliability methodology. This chapter applies the second part of the methodology to two different technologies, POS tagging and NE recognition, on multiple languages, in order to validate the proposed stepwise evaluation reliability methodology.

Chapter 6 provides a summary of the work in this study, with specific recommendations on how NLP evaluations should be designed to attain reliable results. The chapter gives an overview of the new stepwise evaluation reliability methodology and regression modelling techniques for applying the evaluation methodology to other technologies. Finally, areas of future research are described that allow further validation and implementation of the methodology to make this work applicable to a wider audience.

2. NLP EVALUATION AND RELIABILITY

2.1 INTRODUCTION

The evaluation of technology is central to the continued improvement of NLP technologies. Nearly every NLP publication has an evaluation section where results from the author's approach are compared to existing approaches or the state of the art. One of the major problems with evaluations is that they are often not performed on the same data or using the same methodology to calculate the metric results that are reported. This leads to uncertainty about the reliability of the evaluation results and comparisons that are presented in these publications.

The first part of this chapter provides an explanation of what is understood by the term "evaluation reliability" and how this pertains to evaluation metrics. I explain how reliability can be measured, and consider possible sources of evaluation variation.

Based on the overview of reliability, I propose a new stepwise methodology for establishing evaluation reliability that encapsulates the entire evaluation process. This methodology provides NLP researchers with a framework for ensuring reliable evaluation by creating a linear regression model that can predict the expected variability of an evaluation based on the underlying variables that are inherent in the evaluation process. The methodology is based on a combination of existing research and approaches, alongside statistical modelling techniques not previously applied to evaluation procedures. All the steps in the methodology are described in detail in section 2.3 and form the basis for the case study in Chapter 3 and Chapter 4.

2.2 NLP EVALUATION RELIABILITY

Since this study is primarily concerned with establishing whether a given evaluation result is a reliable reflection of intended functionality or not, it is important to provide a definition of what reliability is. With this in mind, the following short description of evaluation reliability gives a broad description of the aspects that are of interest in determining the reliability of an evaluation.

According to EAGLES (1995b: 2) "a metric is reliable inasmuch as it constantly provides the same result when applied to the same phenomena." Three requirements for reliable metrics are of importance to this study:

1. Descriptiveness: How well they describe the functionality of the attribute that is being measured.
2. Stability: The metric should give results that are always similar when measuring the same attribute under the same circumstances.
3. Reliability: The results from a metric should be a reliable reflection of a technology's functionality, which means that when a metric gives a specific result, this result should be an interpretable and trustworthy source on which correct decisions can be based (such as deciding which technology is the best acquisition for a company).

If a metric conforms to these three requirements, it is a reliable measurement of technology's functionality. King & Maegaard (1998:2), Koehn (2004:388), and Riley *et al.* (2004:1) point out that the metrics that are typically used in evaluations often do not give stable results, nor do they reflect the market-readiness of the product. This can be attributed to one of two factors – either that the metrics do not accurately describe the attribute they attempt to measure, or that the calculation of the metrics does not take all variable aspects of evaluations into account.

De Jong & Schellens (2000) hold that evaluation reliability pertains to the similarity of results when a given phenomenon is evaluated, while ensuring that the methodology is free from systematic bias. However, it is not exactly clear what this "similarity" refers to, or how to determine when an evaluation methodology or metric is systematically biased, which raises a couple of questions:

- What are the numerical or statistical bounds for similarity?
- Are results that are similar necessarily accurate reflections of functionality?
- If evaluation results are dissimilar, is this a reflection of an inadequate methodology or a deficient metric?

The following short discussion deals with a few possible evaluation issues regarding the reliability of metrics, and how inconsistencies caused by these issues can be

explained. This discussion aims to determine what the focus of the following experiments should be when trying to determine evaluation reliability.

2.2.1 STATISTICAL BOUNDS OF SIMILARITY

Since almost all NLP research reports evaluation results in the form of metrics that are numerical representations of functionality, one would expect that there is a well-established notion of statistical similarity for evaluation metrics, specifically, determining whether a particular reported result is significantly better than another result when evaluations are performed on different evaluation corpora. In certain resource-rich languages and with conference shared tasks, there has been a concerted effort to standardise evaluation data and make it available to the entire research community, for example:

- Automatic content extraction (ACE), 2003-2008;
- Cross-language evaluation forum (CLEF), 2004-2008;
- Conference on computational natural language learning (CoNLL), 2003, 2004;
- Morpho Challenge 2005-2010;
- MUC, 1987-1997;
- SemEval, 2000-2012; and
- TREC, 1992-2012.

In resource-scarce languages, however, data availability to perform comparative evaluations remains an issue. Furthermore, there are no clear distinctions for determining whether any of the reported results are statistically significantly better than previously reported results, even if it is on the same data set (Berg-Kirkpatrick *et al.*, 2012, Koehn, 2004).

Dietterich (1998) identifies five sources of variation in the validation of statistical methods for designing and evaluating statistical tests (machine learning algorithms in the case of the article). Of the five identified by him, only one is relevant to the evaluation of the statistical test, and that is the selection of evaluation data. However, when considering the evaluation data, Dietterich (1998) references only the size of

the data set, which although important, is not the only variable in the evaluation corpus. Later (cf. 4.2-4.6) I show that there are several other variables in the evaluation corpus that have an influence on the results produced by a particular evaluation.

Ideally, an evaluator should be able to identify the variables in the evaluation and make an estimation of how well the evaluation that is performed conveys the quality of the technology being evaluated. This can then be reported as part of the evaluation results to allow other evaluators to determine whether the evaluation is reliable and comparable to existing evaluations for which the variability is also known.

2.2.2 RELIABILITY AND QUALITY OF RESULTS

Apart from determining whether evaluation comparisons between different systems are significant, there is a secondary question regarding what the variability in evaluation results means when the same system is evaluated multiple times on different data, and how this variability affects the reliability of reported evaluation results. Specifically, if the same system is tested three times and reports metric results of 94.70%, 89.40% and 90.75%, which one of these results is an actual reflection of the quality of the technology? Is it determined by the evaluator who can select the evaluation that best suits the motivation of the evaluation, or can an additional variability value indicate the expected variability in the evaluation results that allows others to determine the validity of the evaluation?

This becomes more important when results are compared with existing evaluations where the same data is not available for direct comparison. Unless there is a reasonable way to determine the similarity between evaluation strategies by accounting for evaluation variables as discussed in the next chapter, all results that are reported should be suspect, and be subjected to additional scrutiny.

2.2.3 METHODOLOGY OR METRIC INADEQUACY

Lastly, the evaluation variability for the same technology using different evaluation approaches can be an indication of either a flawed evaluation methodology or a flawed metric that is used to report the results. Since so much of what are considered

improvements in the field of NLP rely on the reported metrics, it is important to take a critical look at both the methodologies that are used to do evaluations and the metrics that measure the quality. By reviewing the underlying principles of the methodology and the metrics, it is possible to better determine how and when variability in results occurs, and to account for these aspects of the evaluation at the outset of the evaluation procedure, to ensure that reported results are an accurate reflection of quality, and not a biased reflection of evaluator intention.

For the purpose of this study, evaluation reliability is defined as follows:

> An evaluation metric is reliable when 95% of future evaluations are within a ±1% interval from the reported result.

This proposed reliability definition is further explored and explained throughout the rest of this study, with specific reasoning for the proposed requirements.

2.3 STEPWISE EVALUATION RELIABILITY METHODOLOGY

Following from the discussion on metric reliability, it is necessary to provide a methodology that allows evaluators to actively ensure that the evaluation results they present are accurate representations of the quality of their technology. With this in mind, the following outline provides the structure and terminology for establishing an evaluation reliability methodology that can be used to model evaluation reliability in a structured way. This will allow multiple variables to be taken into account during the evaluation process, and should ensure more reliable reporting of evaluation results.

This outlining of the methodology provides a step-by-step guide for determining the relevant variables as well as creating the models that predict the variance of evaluations for a particular NLP technology. The method for creating the model only needs to be created once for a technology, and can then be used by the community of researchers for that technology as part of their result presentation by providing the confidence interval (CI) for their evaluation corpus given the technology model.

2.3.1 STEP 1: DEFINE THE PURPOSE

The first step in initiating an evaluation is determining the purpose of the evaluation. The purpose of the evaluation will ultimately determine what the correct approach to the evaluation will be, which methods and metrics should be used, and the type and origin of data that will be used in the evaluation. The purpose of the evaluation refers to the specific outcome of the intended evaluation, for instance to determine if the technical features of a spelling checker are of high enough quality for the university to buy and distribute the spelling checker (King, 1998). Although there may be additional requirements for determining if the spelling checker will be acquired, for the purposes of the evaluation, the technical aspects are the only concern.

In order to evaluate any product, it is necessary to identify the specific criteria and attributes (or features) that need to be tested, before one can determine what kind of evaluation needs to be done, and how the evaluation should be performed. In this regard, EAGLES (1995b) specifies seven quality characteristics (also referred to as attributes) for the evaluation of NLP applications that need to be addressed during the evaluation process. Six of these characteristics are based on the quality characteristics described in the definition of ISO 9126, which forms the basis of the work done in EAGLES. The seven characteristics identified by EAGLES (1995b) and King (1998) are:

1. Functionality
2. Reliability
3. Efficiency
4. Maintainability
5. Usability
6. Portability
7. Customisability.

These characteristics and examples of their use are described in detail in the spelling checker case study in 3.2.1.

The next step of the process, the desired outcome of the evaluation, will determine what type of evaluation will be done, and this is directly influenced by the purpose of the evaluation.

2.3.2 STEP 2: DEFINE THE TYPE

When establishing an evaluation, it is important to decide on the type of evaluation that will be performed, and to ensure that this aligns with the purpose of the evaluation. Voorhees (2002) distinguishes between two types of evaluations, namely system evaluations and user-based evaluations. User-based evaluations are evaluations that measure a user's satisfaction with the system, specifically how the user experiences the quality of the system in terms of the need it fulfils. System evaluations, on the other hand, determine how well the system performs its intended task. Bernsen & Dybkjær (2000:2) make a similar distinction, although they see user-evaluations as the final part of the system evaluation. They provide the following classification of evaluation types:

- technical evaluation of the system and its components;
- usability evaluation; and
- user evaluation.

They indicate that a technically excellent system is not always easily usable, while a technically inferior system may be more usable and therefore better suited to the end-user. The approach to evaluations should therefore take into account not only the technical ability of a system, but also the usability of the system.

Hirschman & Thompson (1997:410) state that NLP evaluations can be categorised into three different types, each with a specific function. These types of evaluation represent the different purposes of the evaluation, and the purpose of the evaluation determines which one of the following types of evaluation should be performed:

- *Adequacy evaluations* refer to the evaluation of the capability of a system to perform a given task. Does the system do what it is required to do? How well does it do it, and at what cost? The purpose of this type of evaluation is to determine how well the system fulfils the need of a user, by comparing the perceived need of a user with the actual functionality of the system. If a user wants a system to be able to recognise speech automatically, an adequacy evaluation would aim to determine how well the system recognises individual words and spoken sentences.
- *Diagnostic evaluations* are commonly used during the production of a system specifically to test the implementation of a particular component in the

system, for instance the morphological analysis or compound recognition component. The evaluation results are then used to determine whether the component improves the functionality of the system in relation to previous systems with a different version of the component. These evaluations are typically used when a component in an application is changed, and specific instances are evaluated to determine how the changes influence the functionality of the system.

- *Performance evaluations* (also referred to as *functional evaluations*) are measurements of a system's performance in one or more subareas of the system implementation, for instance word boundary recognition in compounds, or analysis of particular morphological forms. This type of evaluation is typically used in research and development environments to compare systems or successive generations of the same product. For example, in implementing different techniques for the recognition of word boundaries in compounds, such as longest string matching or different machine learning approaches, the effect of each technique on the overall functionality of the system is tested.

Another distinction closely linked to the mentioned evaluation types, is drawn between so-called "glass box" and "black box" evaluations. The difference here lies between evaluating particular component parts of the system and the overall performance of a system (Hirschman & Mani, 2003:420; Hirschman & Thompson, 1997:410; Lenci *et al.*, 1999:2). In a "glass box" evaluation, the evaluator knows what the underlying components in the system are, and can test each component and the influence the component may have on other components in the system as well as the effect on the complete system. "Glass box" evaluations are typically used by developers during the development stages of a system, to determine how a component either performs on its own (diagnostic evaluation) or how the component contributes to the system as a whole (performance evaluation).

"Black box" evaluations evaluate a final system, and typically consist of measures that assess a system's overall performance, without reference to or specific knowledge of the components underlying the output of the system. In these evaluations, there is less emphasis on why something is done correctly or incorrectly, but rather on how the system functions as a whole. These evaluations are only used on near market-

ready prototypes or final products, in order to determine how well a particular system fulfils the needs of the user.

Hirschman & Mani (2003:415) point out that there are different kinds of evaluations that should be done during the different stages of the product development cycle, as is represented in Figure 2.3-1 (where GB and BB refer to "glass box" and "black box" respectively). From this representation, it can be seen that both "glass box" and "black box" evaluations should be performed, although "black box" evaluations are preferable during the final stages of product development.

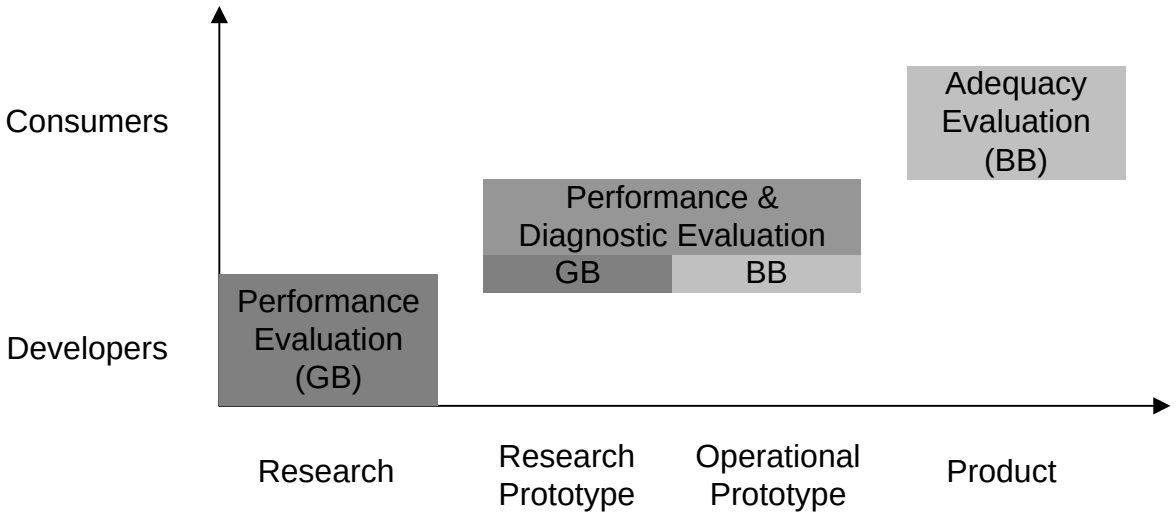

Figure 2.3-1: Technology life cycle and evaluation
(adapted from Hirschman & Mani, 2003:415)

Performance and diagnostic evaluations (i.e. glass box evaluations) are typically done during the development process, where different subcomponents and task-specific modules in the application are tested and evaluated (Hirschman & Thompson, 1997:412; Paggio & Underwood, 1997:277). As an example, spelling checkers' performance and diagnostic evaluations would focus on the behaviour of the spelling checker pertaining to specific lexical or morphological features of the language, such as compound analysis or case inflection coverage. Adequacy evaluations, or "black box" evaluations, aim to ascertain the competence of the entire system without consideration of specific functional characteristics (Hirschman & Thompson, 1997:410). For spelling checkers, for example, adequacy evaluations pertain to the overall recognition of correct words and flagging of incorrect words as experienced

by the end-user, and not to how a spelling checker performs on specific linguistic phenomena such as compounds.

This distinction in the type of evaluation is important because the methods and approaches used in the different types are vastly different. The conclusions that can be made based on the results of the different types of evaluations are also very different. For the purpose of this study, I will focus on determining the reliability of adequacy evaluations where the overall quality of the technology is measured against the expected behaviour.

2.3.3 STEP 3: DEFINE THE METHOD

Over the last two decades, considerable study of how to do evaluations and the effect of different evaluation implementations has been undertaken by various researchers and research institutions. For the most established technologies, these have yielded relatively established methods for the evaluation of the respective technologies. Having said that, there are still various questions and problems with these methods, and researchers continually find irregularities in the methods and the produced results (Callison-Burch *et al.*, 2008; Cimiano *et al.*, 2003, 2004; Soricut & Brill, 2004). The following gives an overview of the main methodological evaluation implementations in the different NLP fields.

Different definitions of what is considered a methodology are presented by different writers, each of whom highlights different aspects of evaluation methodologies. Bussmann *et al.* (2001:143) describe a methodology as a recipe for finding solutions for a specified set of problems, which should be specific enough to be applied to a suitable problem, yet leave enough room for creativity. According to Nance & Arthur (1988:221), a methodology is more than just a recipe; they refer to a methodology as a complementary set of methods with a set of rules on how to apply those methods. This definition implies that a methodology should explicitly state how and what should be done, and in which sequence, to obtain a solution for the defined problem.

Lenci *et al.* (1999:22) contend that the methodology should organise and structure a task, and subtasks, in order to reach global objectives. Before making decisions regarding the methods that are used to achieve any objectives, whether these are

global or local, it is imperative to identify the objectives of the methodology. There are a number of different ways in which to implement these methodologies, and the following short discussion highlights the most important methodological approaches used in NLP evaluation.

2.3.3.1 Gold-standard evaluations

The implementation of so-called "gold-standard" format evaluations (Hirschman & Mani, 2003) is the most widely used evaluation methodology today. This can readily be seen in NLP evaluations for technologies such as information extraction (Amigó *et al.*, 2011; Cimiano *et al.*, 2003, 2004; Galliers & Sparck Jones, 1993), POS taggers (Manning, 2011), named entity recognition (CoNLL, 2002; CoNLL, 2003; Kim *et al.*, 2012), chunking (Pate & Goldwater, 2011), parsing (Abney *et al.,* 1991; Carroll *et al.*, 2002, Schwartz *et al.*, 2011), and word sense tagging (Ruppenhofer *et al.*, 2010). These evaluations are based on a set of training and test corpora that have been annotated, mostly manually or semi-automatically, with some standard mark-up. These corpora sets represent the expected structured output from the technology, and usually take the form of some type of mark-up language such as the standard generalised mark-up language (SGML) or extended mark-up language (XML).

Hirschman & Mani (2003) provide the following stages of the gold-standard evaluation methodology:

- Definition of the evaluation task and the mark-up standards to be used in the creation of the corpora, including annotation guidelines, annotation tools, and comparison tools.
- Development of the annotated corpus for both the training and the test set.
- Providing the tools, task and mark-up definitions to developers to build the systems.
- Evaluation of the systems based on their processing of the evaluation corpus and comparing the output to the gold-standard corpus.

The corpora are then used to respectively train and test the technology, by partitioning the corpus into a training and a test set. Though there are some idiosyncrasies about the appropriate partitioning (Cimiano *et al.*, 2003, 2004), it

generally consists of an 80/20 division between the training and the test set. The system is developed and trained on 80% of the annotated data, and the remaining 20% is used for evaluation. After training the system on the modelling corpus, the remaining 20% of the corpus is processed by the system, and the output is compared with the original gold-standard mark-up. The resulting comparison is expressed as an evaluation metric, in most cases precision and recall, which is discussed in more detail in 2.3.6.

The use of gold-standard evaluation is not limited to machine trainable algorithms and implementations, and the approach of a set of "golden" expected results to compare with the output of the technology is very common in machine learning tasks. As Soricut & Brill (2004) point out, gold-standard evaluations are geared towards any NLP technology where the correct answer is usually unique (or low-level tasks, as they call them). This type of evaluation is the most typical evaluation of proofing tools, and a more in-depth discussion of this method follows later in this study (c.f. 3.2.5.1).

2.3.3.2 Post-computational judgement

For some technologies, what Hirschman & Mani (2003) refer to as output technologies and Soricut & Brill (2004) call high-level tasks, the evaluation makes use of methods that have users make judgements on the output of the system rather than comparing the output to predetermined expected outputs. This methodology is mostly used for the evaluation of technologies such as natural language generation (Koller *et al.*, 2009); question answering (Lin & Katz, 2006); and MT (Liu & Ng, 2012; NIST, 2002; Papineni *et al.*, 2002; Thompson, 1991; Turian *et al.*, 2003). These methodologies rely on evaluation through intrinsic and extrinsic measures. Although these technologies are the most complex, and although a substantial amount of research into the application of these evaluation methodologies has been undertaken, there are still significant gaps in both the methodologies that are implemented and the results that are reported (Callison-Burch *et al.*, 2008), especially with the introduction and wider use of crowd-sourcing strategies (Ambati *et al.*, 2010; Zaidan & Callison-Burch, 2011).

In the most general sense, the post-computational judgement methodologies rely on judging the output of a system against a predefined set of criteria to determine its quality in terms of output quality and informativeness of the output. Output quality generally refers to the linguistic quality of the output, that is to say, are the outputs coherent and well-formed instances of the language? In an application such as MT, for example, this can refer to the understandability of the translation if read by a native speaker.

Informativeness refers to how well the generated output reflects the content of the original input. For example, a well-written coherent summary, which does not contain the essential content of the original document, is not a good summary, while an incoherent or difficult to read summary containing all of the relevant content is also undesirable. In the 1966 ALPAC report, this was one of the major shortcomings according to the ALPAC evaluators since the additional work required to make the machine translations understandable was substantial, making the use of machine translations in the field unfeasible at that stage.

For technologies that require post-computational judgements during evaluations, more often than not there are several different "correct" possibilities, which make evaluation especially difficult (Soricut & Brill, 2004). For this reason, evaluation methodologies are not absolute in their assessment of correctness. In general, the criteria against which these technologies are judged are created by human evaluators who assign some score to the output on grading scales (Callison-Burch *et al.*, 2008; Doyon *et al.*, 1998; Hovy *et al.*, 2002; White & O'Connell, 1994). The immediately obvious shortcoming of this approach is that the judgements that are made by evaluators are subjective, and the inconsistency between evaluators is well documented (Callison-Burch *et al.*, 2008; Hovy *et al.*, 2002). To counter this subjectivity, most state of the art evaluations, such as those used at the MT Summit conferences, MUC, TREC and several other NLP competitions, require multiple evaluators to assess the same output and to normalise the results. For an already expensive evaluation undertaking, this hugely increases the amount of time and money required to do such evaluations.

As a counter for the discussed problems, there has been a wide-scale drive to find evaluation methods that automatically compare output to other output, and report relative measures of informativeness. These measures vary widely in their comparison

techniques, from sentences, clauses, phrases, to passages or even *n*-gram-based measures, such as the following well-known metrics developed over the last decade:

- Bilingual evaluation understudy **(**BLEU) (Papineni *et al.*, 2002);
- Metric for evaluation of translation with explicit ordering (METEOR) (Banerjee & Lavie, 2005);
- Translation edit rate (TER) (Snover *et al.*, 2006); and
- National institute of science and technology (NIST) score (NIST, 2002).

These automatic evaluations attempt to compute a correlation measure between the machine translation and a human translation of the same text. Although initially proposed by Thompson (1992), these measures were not widely accepted in the evaluation community until fairly recently (Hovy *et el.*, 2002). Over the last five to eight years, these measures have been implemented on a much wider scale, although there is still a lot of research around the validity of these measures and their improvement.

The search for automated evaluation methods in MT is indicative of a wider trend in NLP evaluation in the last 20 years. The search for reliable automatic evaluation methods has been one of the major focus points in evaluation research. This is not limited to MT, but has been the focus of all evaluations ever since the early 1990s. Automated evaluation procedures are also one of the main points of focus in this study, and are discussed in more detail later (cf. 3.2.5.1, 3.2.5.1.2).

2.3.4 STEP 4: COLLECT THE DATA

Once the method for evaluating the technology has been defined, the next step in the process is to find and annotate the evaluation corpus. It is important to keep both the type and method of evaluation in mind when collecting the evaluation data, since the data must conform to the evaluation type and method. The data collected in this step is used for both the next steps in the process – identifying the evaluation variables that are relevant for predicting the CI of the evaluation, and verifying that the model generated by this procedure accurately predicts the CI of evaluations when applied to previously unseen validation units.

The construction of a corpus for evaluation purposes has to take into account the objectives of the evaluation and must be representative of the functions that are evaluated in line with the best practises of corpus design. Van Zaanen & Van Huyssteen (2003:46) state that the corpus used during the evaluation is of utmost importance as it necessarily influences the results that are obtained from an evaluation. MacMullen (2003:12) also indicates that the standardisation of the corpus allows for a relative level of comparability between evaluations and products. This implies that if a standardised evaluation methodology and test material design criteria are in place, both intra- and inter-language comparative evaluations may be possible, that is to say between different versions of technology for the same language or between different versions of technologies in different languages.

Sinclair (2004) states that the design of a corpus should reflect the language of the users, and the designer should accept whatever is represented in the data and not be influenced to include items that are perceived as more important or interesting (Fillmore *et al.,* 1998; Galliers & Sparck-Jones, 1993; MacMullen, 2003). The corpus, as part of the evaluation procedure, should therefore reflect the nature of the texts that are generated by the prototypical user rather than the intuition of the evaluator as to what users would probably do.

In previous spelling checker evaluations, the types of test material that were used were limited in their scope of text genres. For example, the evaluations of the TEMAA and SCARRIE projects were done on newspaper articles, general interest magazine articles, and transcriptions of news wires (Paggio & Music, 1998:279; Paggio & Underwood, 1997:10; TEMAA, 1997). Although the content of the material would cover a large amount of varied data, the style of language that is used in these kinds of popular writing would not reliably reflect a spelling checker's performance on more technical texts, such as scientific and legal writing, which are environments in which it is reasonable to expect spelling checkers to be used.

Because the users of spelling checkers create a very diverse set of texts and it is almost impossible to determine what each individual user would expect from the product, the evaluation corpus that is used to do spelling checker evaluations must represent as many of the possible users as possible. This means that the text that is used should cover as many different domains as possible, while still remaining within a reasonable size (MacMullen, 2003:14). The evaluation corpus must be

representative of the language used by prototypical users, in order to evaluate how accurately the spelling checker would process actual users' language. Furthermore, the evaluation corpus must be constructed in such a way that it can be replicated recurrently, and in a similar fashion for different evaluations and languages, if necessary. The following section deals with the way an evaluation corpus should be structured in order to do accurate evaluations. This mainly focuses on the construction of a representative evaluation corpus, as the state of language users employ when creating texts in text editors.

One of the presuppositions of this study is that by constructing a representative evaluation corpus, it is possible to evaluate a spelling checker and represent as wide a variety of users as possible. Sinclair (2004) supports this view, but adds that the corpus should be collected without knowledge of or regard for the specific language phenomena in the selected texts; it should rather regard the communicative function of the text, which is closely linked to the genre of the specific text. Different communicative functions identified by Sinclair (2004) exist, such as:

- informative function (as in the case of newspapers);
- instructive function (as in the case of textbooks);
- research function (as in the case of scientific journals); and
- entertainment function (as with popular fiction).

The fact that different texts have different communicative functions means that if a text is categorised within a certain genre by its function, the fact that micro-linguistic phenomena (such as specialised terminology, infrequently used morphological constructions and the like) occur in the text, should not be taken into account when selecting the text. This further implies that a text should be selected only because of its communicative function (i.e. the kind of genre within which it is created), irrespective of the content. The selection of a text based on its communicative function is referred to as external criteria, and should be the only criteria for the design of representative corpora (Sinclair, 2004). Internal criteria, on the other hand, refers to texts which are selected because of specific phenomena that occur within a specific text, such as creating a text by selecting only exceptions on language rules to conduct research, and would lead to skewed evaluation results if the intention was to be representative. This also applies to an evaluation corpus that could lead to skewed

results if it is constructed by specifically choosing morphologically complex words and ignoring function words.

True representativeness, according to Biber (1993), is the extent to which a corpus sample includes the "full range of variability" in a given language. Unfortunately, this is only possible in very large corpora (Galliers & Sparck-Jones, 1993:126); for use in evaluations, such large corpora are not ideal, as they are difficult and expensive to construct and maintain. Chung (2003:226) points out that the representativeness of a corpus depends on the quality and composition of the corpus in line with the phenomena that are being investigated (or in this case, being evaluated). Because the evaluation of spelling checkers aims to evaluate the coverage of the language used in text editors, the corpus should represent as many authors who use the editors as possible.

In order to be representative of any language as a whole, it is necessary that a corpus include samples across a variety of texts that reflect the range of syntactic and semantic phenomena across that language (Sinclair, 2004). Balanced corpora are especially crucial for tasks such as lexicon building (Fillmore *et al.*, 1998:2) therefore when trying to evaluate a lexicon (which is what spelling checker evaluation in a certain sense amounts to) this balance is just as important. To this end, it is also important to have errors in the data, as this is one of the critical attributes of spelling checker evaluation, and edited texts, such as newspaper and scientific journal data, do not include many errors. The data should also not include too many errors, such as in the case of second language users, which could once again influence the evaluation. In addition, terminology occurs more frequently in specialised texts than in general-use texts. These terms are restricted to a particular topic area, and are reasonably frequent in that area but not anywhere else (Chung, 2003:228; Sinclair, 2004). For this reason, it is important to ensure that a wide variety of different texts should be used in the evaluation corpus. By doing this, micro-linguistic phenomena do not adversely affect the outcome of an evaluation, while still being represented in the corpus (Sinclair, 2004:8).

Sinclair (2004:11) indicates that there are six important steps in the creation of a representative corpus:

1. Determine the structural criteria of the corpus and create a framework for the corpus based on these criteria. This means that when the corpus is conceptualised, there should be some set of criteria for what kinds of text and writing should be included in the corpus. For spelling checker evaluations, some of the criteria for the compilation of the evaluation corpus would be that the material should be in electronic format, and in a particular encoding format that makes the automatic processing of the data possible. The data should also come from a wide variety of functions and genres, and should include both edited and unedited data if at all possible.
2. Create a thorough inventory of text genres and available material, using only external criteria (i.e. the function of the text). During this phase, the corpus should be designed to include as many of the different text genres created by users of text editors as possible. When constructing such a list, one has to try to get an exhaustive list of possibilities before excluding any of these sources. By including a number of different genres, it is possible to get samples of many different types of language use, and thereby include not only common words, but also more specialised terminology.
3. Place the texts in order of importance, based on their relative importance with regard to the kind of corpus that is being created. This is important when deciding how many words from each text to include.
4. Estimate a target size for each text type, taking into account:
    a. the overall target size of the corpus;
    b. the number of text types identified;
    c. the importance of each text type; and
    d. the practicality of gathering the data.
5. During compilation, compare the actual size of each text type with the original plan.
6. Document all steps in the process, so that future researchers and users of the corpus have reference points if they obtain unexpected results.

Since it is important to keep track of all of the different types of data collections used in the study, the following short descriptions and definitions are given as terminology and definition for each of the different types of data collection. The defined terminology will be used throughout the rest of this study.

Similarly to machine learning approaches to data, a high-level distinction will be made between the data used in creating the relevant regression models and the data that will be used to verify the generated models. In order for the models to be properly validated, 80% of the available data is used to create the regression models and the remaining 20% for validation purposes. The collection of all of the available annotated data will be referred to as the **evaluation corpus**.

The following sections provide an outline of the types of data collections used in the study, along with descriptions of the data, before providing an example of each of the collections.

Modelling Corpus: The modelling corpus is the full set of data that is available during the modelling stage, and typically consists of a set of documents with the relevant data in them. Ideally the individual documents in this corpus will have some attribute information associated with the data, such as the complexity, genre, and word count of the document. For gold-standard evaluations, the data must also be annotated with the expected output of the technology that will be evaluated. This set is used to select the modelling set.

Modelling Set: The modelling set is a selection of data items from the modelling corpus consisting of 10 or more modelling units, where each unit in the set has some shared attribute. This shared attribute is the number of words in most cases in this study, but can also be units with the same genre, type-to-token ratio (TTR), or any other shared attribute. Each unit in the set must be a unique, randomly selected datum from the modelling corpus, such that there is no overlap between the units.

Modelling Unit: The modelling unit is a collection of individual data items (words, sentences or documents depending on the technology being evaluated) against which the particular technology will be evaluated and for which metrics are calculated. Modelling units are always combined into modelling sets, and are randomly selected from the modelling corpus according to the shared attribute value defined for the modelling set. Once the individual units have been selected, unit level variables (such as TTR) must be established and annotated on the unit, after which the evaluation for each metric of interest for modelling should be performed. The results of the evaluations on the modelling units are then used to determine regression values for evaluation reliability.

Validation Corpus: The validation corpus has the same structure and annotations as the modelling corpus, but should be a quarter the size of the modelling corpus. The data in this corpus must never overlap with the modelling corpus, and should never be used during the model creation steps. This set is used to select the validation set.

Validation Unit: When validating the regression models, the validation process requires 10 or more evaluation units for which the predicted variability can be compared to the actual variance. These units are selected at random from the validation corpus such that there is no overlap between the different units, and that there is some variability in the distribution of values for the different variables that have been identified for the regression model.

As an example, for spelling checker variability regression modelling, the **evaluation corpus** consists of 400,000 words, where all spelling errors and their expected corrections have been annotated. This corpus is then split 80/20 into the **modelling corpus** and the **validation corpus**, consisting of 320,000 and 80,000 words respectively. The **modelling corpus** will then be used to create the required **modelling sets**, while the **validation corpus** will be split into 10 **validation units** where there is no control for any of the variables inherent in the evaluation.

Prior to selecting the **modelling sets**, the evaluator must decide what common attribute will be used to differentiate sets, such as number of words, and what the different set requirements will be. Once this has been established, random selections of **modelling units** must be gathered from the **modelling corpus,** so that there are 10 units for each attribute value. The spelling checker example requires modelling sets of 1,000, 5,000, 10,000, 15,000 and 20,000 words. The number of words selected for these experiments was such that there would be incrementally larger sets of data. This would allow the models to be uniform without requiring huge data sets by doubling the number of words for each increment.

By following the stated requirements the following sets are selected from the **modelling corpus**:

- 10 **modelling units** of 1,000 words;
- 10 **modelling units** of 5,000 words;
- 10 **modelling units** of 10,000 words;
- 10 **modelling units** of 15,000 words; and

- 10 **modelling units** of 20,000 words.

These **modelling units** are then processed by the technology, metrics calculated for each unit individually and combined to create the relevant regression model. Once all the data has been collected and annotated, the next step in the methodology is to statistically identify the set of variables that are good predictor variables.

2.3.5 STEP 5: IDENTIFY POSSIBLE VARIABLES

The first step in creating the evaluation regression model is to identify all the possible variables that are inherent in the technology and evaluation material. All variables that are identified in this step must be quantifiable in an established and reliable fashion, but can be either continuous or discrete.

Since each technology has a different target and function, it is almost certain that all technologies will have a different set of variables that could possibly influence the results and reliability of the metrics. It is not important at this point of the process to try to determine whether the set of identified variables are the ones that actually have an effect on the evaluation results, as that is explicitly calculated as part of the model creation in a later step of the process.

Identifying variables that may be relevant to the variability of evaluation results is not an exact science since there are many hidden factors that may contribute to the observed variability. The identification also relies on inherent knowledge about the technology and understanding the types of inherent factors that may influence the evaluation results. These variables will be directly attributable to the evaluation corpus. Variable identification is, for the most part, a process of analysing the data collected in the previous step and identifying those changeable aspects of the evaluation that possibly have an effect on the evaluation results. During this step of exploratory data analysis, it should also become apparent what the underlying relationship between the data and the evaluation results is. An extended set of example variables is available for spelling checkers in 3.2.4.

2.3.6 STEP 6: IDENTIFY RELEVANT METRICS

For as long as evaluations have been performed in the field of NLP, researchers have tried to establish quantitative measures of the linguistic quality of the technologies that they develop. Metrics are the most widely used method of reporting evaluation results, and mention of metrics can be found in almost every NLP publication. The first mention of these types of measures was with the definition of precision and recall in the 1960s (Hutchins, 1995). These metrics are still in use in most state of the art evaluations today.

Although metrics are widely used, the exact definition of what individual metrics are is not always easily available. According to El-Emam (2000:1), metrics are measures that quantitatively characterise some attribute of a software product. In addition to being quantifying characterisations, metrics should be accurate and reliable measurement instruments that reflect the quality of a software system. EAGLES (1995b) probably makes this the most explicit in its description of the function of metrics when they state "a metric is reliable inasmuch as it constantly provides the same result when applied to the same phenomena." Based on this, when metrics are reported for a particular technology, other researchers should be able to interpret the metrics results to assess the quality of a technology, purely based on the metrics provided. These metrics should therefore be good descriptors of the attributes they measure.

As was mentioned earlier, the precision and recall metrics are the oldest and most widely used metrics in the field of NLP. In its original implementation for IR, precision was defined as the proportion of selected items that the system got right, while recall was defined as the proportion of the target items that the system selected (Manning & Schütze, 2003). For IR, precision describes the number of expected documents that were returned as a proportion of all the documents that were returned. Recall, on the other hand, reports the percentage of documents that were expected to be returned and were in fact returned.

From these initial metrics, a number of variations of recall and precision metrics have been devised to give more accurate indications of the functionality of IR systems, based on the recall and precision metrics. The best-known of these are F-measure and average precision. F-measure is a combinatory metric that reports the harmonic mean between recall and precision measures. This is a weighted metric that can be

adapted to reflect the intention of the evaluation by assigning a larger weight to either recall or precision, depending on what the evaluator perceives to be more important. Recently, ranked retrieval has become an increasingly more important attribute of IR, where precision is measured at different recall points (such as recall at 10%, 20%, 30%, and so forth) and averaging the score. This increases the reliability of the precision metric as a descriptor of functionality (Singhal, 2001).

For most gold-standard evaluations, precision, recall, and F-measure are usually good descriptive metrics. For this reason, these measures have been used in most NLP evaluations at some point. However, for more complex technologies, additional metrics have been devised to:

a) describe the response of human evaluators; and

b) calculate overlap between machine output and human gold-standard output that has a positive correlation with human evaluations of the same output.

These metrics are most commonly used in the post-judgement type evaluations of MT, automatic summarisation, and information extraction systems.

The metrics used for a), such as fluency, understandability, and relevance, are dependent on a large number of human evaluators assigning scores on a scale to the output of each system, and averaging the scores of these evaluators to determine how fluent or understandable a particular output is. Relevance on the other hand, rates how relevant each of the documents retrieved by an IR system is to the topic that was used for the search (Singhal, 2001).

The second type of measure tries to determine how well the output of the system aligns with a human version of the same task. These types of metrics are most commonly used in MT and automatic summarisation systems, where overlap and consecutive overlap between textual elements such as unigrams, bigrams and phrases are calculated. The best-known of these measures is the BLEU metric (Papineni *et al.*, 2002) which calculates an adapted *n*-gram precision for a translation, then calculates the geometric mean of the precision scores and positive weighting of the scores. Papineni *et al.* (2002) then show that there is a positive correlation between the BLEU metric and human judgements, meaning that although the metric is not obviously descriptive in the value that is output, it does give an indication of quality

when compared to other systems. Similar metrics have been developed for automatic summarisation (Lin & Hovy, 2003).

2.3.7 STEP 7: VALIDATE THE METRIC

According to Rossi & Fernandez (2003:2), the validation of metrics is done on two levels, namely theoretical validation (also referred to as internal validation by El-Emam (2000:2)) and empirical validation (external validation according to El-Emam (2000:2)). The validation of metrics, whether theoretical or empirical, aims to determine if the metrics are accurate representations of the functionality of the technology that they aim to evaluate. As such, validation aims to do more than just determine whether the results of an evaluation are reliable and reproducible, but tries to determine if all of the methods used in the evaluation are correct, and the underlying linguistic phenomena have been taken into account when setting up the evaluation. When designing an evaluation, the validation of the metrics therefore also serves as validation of the methods that are used in the evaluation.

2.3.7.1 Theoretical validation of metrics

Theoretical validation is described as a theoretical endeavour to ensure that a metric is a correct quantitative representation of the attribute it intends to evaluate (Rossi & Fernandez, 2003:2). According to El-Emam (2000:2) this is a process whereby the software engineering community reaches consensus on what the properties and measures of the software attributes should be. The fact that theoretical validation is a process of discussion and testing within the NLP community implies that theoretical validation evolves over a period of time, and is only completed once a metric is widely accepted as being the most accurate measure of a given attribute (El-Emam, 2000; Rossi & Fernandez, 2003).

The process of theoretical validation takes into account a number of criteria through a number of phases. As an initial criterion, the purpose of the evaluation and the technology should be explicitly investigated to determine what the underlying set of attributes of the technology is. The attributes identified during this phase firstly establish what functionality should be measured by a particular metric, and also

provide indicators of the methodological challenges that need to be taken into account when eventually designing the evaluation.

Once the individual attributes inherent in the technology have been identified, the evaluator should determine what underlying factors are important in the description of the attribute. Many of the attributes identified during this phase also have sub-attributes that are more specific descriptors of functionality, and should also be determined at this point. The more complete the set of attributes is, the easier it is to determine what variable should be taken into account when designing and implementing the metrics for the evaluation.

After identifying the attributes and sub-attributes of the technology, an evaluator should establish how the functionality of each attribute can be described quantitatively through variables that can be used in a descriptive metric. The metric description must somehow describe the functionality of the technology in such a way that it is comparable with the way in which some user will experience the implementation of the technology. As an example, for information retrieval, a metric that merely indicates how many documents are returned for a particular query, is not of much use, since returning every available document would get a very high score, but would not indicate how useful the returned documents were. This metric would therefore fail theoretical validation, and time should not be spent implementing tests that measure this attribute. A metric that measures the percentage of all relevant documents returned is a much better descriptor of functionality, and would pass theoretical validation.

In most cases, no single metric can describe all of the attributes or functionality of a technology, and metrics should be used in some combination to accurately describe the full functionality of the technology. For this reason, most evaluations report results for more than one metric, since the different metrics report different quality characteristics, most of which are mutually exclusive. As an example, a general rule of thumb for precision and recall metrics in IR is that an increase in either precision or recall has an inverse effect on the other metric – as recall increases, some precision is usually lost, and vice versa. Depending on the target of the technology, systems may optimise for either precision or recall, or alternatively for a balance where neither precision nor recall is exceptionally high.

For more complex technologies, such as automatic summarisation, MT, and coreference resolution, the theoretical validation of metrics is an on-going process. As an example, the MT BLEU metric proposed in 2002 (Papineni *et al.*, 2002) shows that there is a positive correlation between the quality judgements of humans and the co-occurrence of n-grams in the translations. Determining how this relates to the translation as a measurement of a particular attribute has been more difficult to establish, and there have been various critiques (Banerjee & Lavie, 2005; Callison-Burch *et al.*, 2006; Lavie *et al.*, 2004) of the metric because of this theoretical conflict. For this reason, other metrics such as NIST score, METEOR, and TER have been developed, which have a better theoretical foundation (Banerjee & Lavie, 2005; Snover *et al.*, 2006).

With regard to spelling checkers, most of the metrics have been in use for a number of years (EAGLES, 1995a; Lenci *et al.,* 1999; Paggio & Underwood, 1997; Starlander & Popescu-Belis, 2002; TEMAA, 1997), and the evaluation community (i.e. researchers and evaluators) have, for the most part, come to an agreement on the theoretical validity of each of these metrics. For this reason, the focus of the experiments in the rest of this study is on empirical validation, rather than theoretical validation. In instances where a metric is not theoretically validated, a full description of the metric is provided with the specific intention of explaining how the metric describes a given attribute. These descriptions lay the groundwork for the theoretical validation required of the metrics, but must be tested and accepted by the wider evaluation community in order to be fully theoretically validated.

2.3.7.2 Empirical validation of evaluation metrics

The empirical validation of metrics entails the structured testing of metrics through experiments in an attempt to validate the metrics statistically (Kitchenham *et al.*, 1995:932). Although the empirical validation of evaluation metrics would seem like an obvious and critical part of evaluation methodology design, there have been relatively few validation efforts on the part of the NLP community. Although Entwisle & Powers (1998) point out that there are various problems with results that have been reported in NLP evaluations, there is still no proposed way to determine the reliability of an evaluation statistically. Empirical validation is one of the central tenets of this study, with specific focus on two important aspects of empirical

validation. I firstly outline a stepwise methodology for scoping the evaluation, including procedures for identifying the evaluation variables that affect evaluation metric results. Secondly, I propose an explicit statistically-based procedure for validating evaluation results, as well as being able to predict evaluation reliability based on the inherent characteristics of the evaluation. The following discussion of empirical validation draws mostly on statistical principles to provide the basis for doing metric validation in NLP. The discussion is based on Neter *et al.* (1996), Devore & Peck (1986) and Field (2005).

The process of metric validation from a statistical perspective is a process of determining which factors, or variables, in the evaluation process, have a statistically significant effect on the results reported by the metrics, and how each of these variables influences the results. By identifying the variables and their influence on the metrics, it is possible to either predict the behaviour of a metric given a particular design or control the set of variables in an evaluation in such a way that the variables are taken into account to ensure reliable metric results. The following discussion outlines a statistical approach to modelling linear regression models that describes the relationship between dependent and independent variables.

A statistical method for predicting the behaviour of a variable based on one or more variables is known as regression analysis. Regression analysis utilises the statistical relation between dependent (or response) variables and independent (explanatory or predictor) variables. Unlike a functional relation (Figure 2.3-2), the statistical relation between variables is not a perfect one, and observations do not fall perfectly on the relationship curve, but form a trend (Figure 2.3-3) where not all observed instances are perfectly described by the relational function.

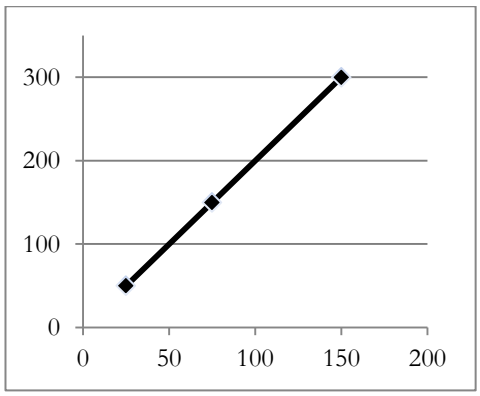

Figure 2.3-2: Functional relation: $y = 2x$

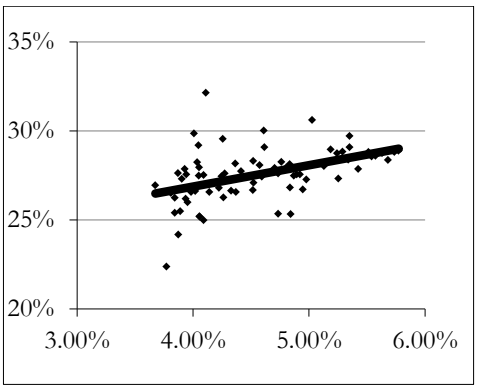

Figure 2.3-3: Statistical relation: precision and error percentage

The dependent variable is the effect or result of the process described in the statistical procedure, and is expected to change whenever one or more of the independent variables are changed. The independent variables are those attributes or features that influence the result in a meaningful way, and are used to predict the result of the dependent variable. As an example, when attempting to predict the likelihood of rain on a given day, the likelihood is the dependent variable, while any number of attributes can be independent variables. Some examples of independent variables in this case are:

- barometric pressure;
- wind direction;
- humidity;
- temperature;
- cloud cover;
- day of the week;
- date, etc.

One of the purposes of statistical analysis is to determine if and how independent variables influence the dependent variable and to describe the relationship between the independent and dependent variables. In the list of example independent variables, some of those variables may have a coincidental relationship with the dependent variable, such as day of the week, and the modeller should ensure that

there is a causative relationship between the independent and dependent variable prior to doing any modelling that includes the spuriously correlated variable.

Regression modelling is a formal means of expressing this relation when the dependent variable varies systematically with the independent variable to form a scattering of points around the curve of statistical relationship. The regression model serves three major purposes, namely:

a. description;

b. control; and

c. prediction.

Although not all three functions are necessarily addressed in all studies, in the case of determining evaluation metric reliability, at least two of these purposes are relevant. Firstly, regression modelling should be able to predict the variability in results for a particular metric based on a set of evaluation variables. By being able to predict the variability, the statistical relation can also be used to control results, by imposing requirements on evaluation design and methodology to ensure that results conform to the reliability principle put forth in 2.2. These functions and how they relate specifically to evaluations are extensively discussed in Chapter 4 and Chapter 5.

2.3.8 STEP 8: SELECT INDEPENDENT VARIABLES FOR THE METRIC

The next step in this methodology is the identification of evaluation variables that have a statistically significant effect on the evaluation results. This is done by doing a large number of evaluations with a static set of tools (technologies) on different modelling sets where each set has some variance in the set variables in the modelling corpus. Based on the collection of evaluations, the forward stepwise regression procedure (FSRP) is used to determine the best predictors of variability for each evaluation metric for the particular technology.

This procedure entails creating multiple disjoint modelling sets from the annotated data collected in the previous step. The modelling sets should be selected randomly from the larger modelling corpus by selecting the smallest unit that makes sense for the technology being evaluated. As an example, spelling checkers are only interested

in words, so one should select unigrams from the modelling corpus to create the modelling unit. On the other hand, POS taggers require contextual information, and the smallest valid unit would be a sentence. For other cases like information retrieval, this smallest unit would be a document, and other technologies may require some sub-unit of a document like a chapter or paragraph.

By selecting the data randomly, the distribution of data in each modelling sets is more likely to produce metric distributions that are Gaussian and ensure that the regression methods used for establishing a model are accurate. After identifying the subunit and creating a random selection algorithm, the data should be split into sets without controlling for any variable other than size. The one variable that is inherent in all evaluations is the size of the modelling and validation units, and although that may not necessarily be a good predictor of variance, there is no evaluation that does not have a size (in whichever the relevant unit is). Based on this, split the modelling set into at least 10^5 modelling units, each containing the same number of elements (x). After creating the 10 modelling units for size x, double the number of elements and create another set of 10 modelling units that contain 2*x elements. Continue doubling the number of elements until 10 modelling units of x*y are greater than the total number of elements in your initially annotated modelling corpus. For example, if you have 200,000 words in your modelling corpus, the largest group of disjoint modelling units that can be generated in this way is 20,000 word modelling units, since any larger number would create modelling sets containing overlapping data items.

Once the large modelling corpus has been split into the disjoint modelling sets, record the variable representation for each modelling set, namely the numeric representation of each variable within each modelling unit, including the number of elements in the modelling unit. After generating the modelling unit at each size interval and collecting the variable values, run evaluations on each modelling unit individually, and calculate the metrics that are relevant for the technology being evaluated. Calculate the mean and deviation of the evaluations for each metric for each modelling set. Combine the evaluation results with the numeric variable representations in a table that can be used in a statistical programme of your choice.

[5] Ten units are selected here to be in line with the widely accepted approach of 10-fold cross validation used in machine learning evaluations.

Using the recorded variables as independent variables and the deviation of each metric for the technology as the dependent variable, use a selection algorithm to identify the variables that are good predictors of variance that occur within the modelling units. Depending on the number of variables that are identified for the evaluation, either FSRP or the best subsets algorithm can be used to identify those variables that most accurately predict the variation in results for a particular metric. The fact that each metric targets a different functionality and therefore can have different independent variables that are relevant, means that this procedure should be completed for each metric individually. Once all the variables that are of interest are identified, the process of constructing the regression model that best predicts the variance is selected in the next step.

Although all the steps are required for regression analysis, of particular interest is the initial creation and refinement of the model. While identifying and collecting values for variables is essential to the process, the exploration of the variables that actually influence the results of evaluations is one of the central tenets of this study. This becomes more evident when considering the list of independent variables discussed in 3.2.1 and trying to identify whether all of the variables, some combination of variables, or even only one of the variables, has a significant influence on a particular metric. However, the more independent variables are used, the more variance there is in the regression model, thus worsening the model's predictive ability. It is therefore also important to identify only the variables that have a statistically significant relation with the dependent variable when constructing the linear regression models.

There are several techniques for identifying the variables that are the best predictors for a particular dependent variable. The choice of technique is usually based on the number of independent variables that are identified, but usually the different techniques lead to similar or identical sets of relevant independent variables. The main techniques for variable identification are as follows (Field, 2005; Neter *et al.*, 1996):

- all-possible-regression procedure;
- "best" subsets algorithm;
- forward stepwise regression procedure;
- backward stepwise regression procedure;

- ❖ forward selection procedure; and
- ❖ backward elimination.

Of these procedures, the all-possible-regression procedure produces several possible good models, while all of the other procedures generate a single "best" model. The fact that there is only a single "best" model can sometimes hide the fact that there may be several good models, and it is always necessary to use diagnostics to determine the quality of the predicted model.

In this study, the process for finding the best independent variables is carried out using FSRP, which aims to identify a single regression model as "best" from the full potential set of regression models. Although other methods, such as best subset, would be adequate for identifying the relevant independent variables, FSRP is used because it scales better to larger variable sets while still being accurate for small variable sets. The following is an overview of the implementation of FSRP. The statistical terminology used here is described in detail in Addendum A.

The first step in doing FSRP modelling is to fit simple regression models for each of the potential X variables and to determine if the slope is zero by calculating the F^* statistic[6]:

$$F_k^* = \frac{MSR(X_k)}{MSE(X_k)} \qquad (2.3\text{-}1)$$

The X variable with the highest F^* value, say X_4, is then the first candidate for addition, and is paired with all other X variables, to calculate the partial F test statistic. This statistic tests if $\beta_k = 0$ when X_4 and X_k are the variables in the model.

$$\frac{MSR(X_k|X_4)}{MSE(X_4,X_k)} = \left(\frac{b_k}{s\{b_k\}}\right)^2 \qquad (2.3\text{-}2)$$

If the F^* value exceeds a predetermined level, it is added to the model, otherwise the variable is dropped. The predetermined minimum F to enter value does not have precise probabilistic meaning, but is usually based on the F score given the degrees of freedom at .95 (with 0.05 significance), e.g. $F(0.95, 1, 50) = 4.03$ so the minimum F to enter is set to 4. If, for example, variable X_7 is added (i.e. F^* statistic is higher than the predetermined level), the technique then tests if any of the variables can be

[6] Please see Addendum A for complete descriptions of the statistical properties and functions used in this section.

dropped from the model, by calculating the partial *F* test statistic. If this *F** value falls below a predetermined value, it is dropped from the model, or else it is retained.

$$F_4^* = \frac{MSR(X_4|X_7)}{MSE(X_7,X_4)} \qquad (2.3\text{-}3)$$

If X_7 is retained, FSRP examines the next independent variable as a candidate for inclusion in the model, and then examines whether any of the existing variables can be dropped. This continues iteratively until no further X variables can either be added or deleted, and thus the "best" linear regression model is obtained.

In the context of empirical validation of metrics, the deviation of the metric is defined as the dependent variable, while the variables described in 3.2.1 are the independent variables. FSRP is used extensively later in this study (cf. 4.2-4.6 and Chapter 5) to determine which of the discussed variables should be included in the different regression models created, for evaluation result validation. The following section explores a comprehensive set of the independent variables inherent in spelling checker evaluations, in preparation for the statistical analysis and modelling performed in the following chapter.

2.3.9 STEP 9: SELECT AND VERIFY RELIABILITY MODEL

In the previous step, the tests were performed that identified the variables that were strongly correlated with the variance in the results for each metric. This step selects the regression function that best fits the results for each metric and the variables identified by the previous step. Fitting the regression function calculates the value of the estimated regression coefficient, using the least squares method.

This selected regression function contains regression coefficients for each of the variables selected in the previous step, and provides the weight for each of the variables that are combined with the numerical value assigned in the second step of the process to predict the variance of an validation unit given the set of variables and the model generated by this step. Typically the model will follow the form of

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \varepsilon_i \qquad (2.3\text{-}4)$$

where:

- Y_i is the value of the dependent variable;
- β_0, β_1 and β_2 are parameters;
- X_{i1} and X_{i2} are the known values of the predictor variable for the *i*th trial; and
- ε_i is the standard error of the estimate.

In the case of the regression function calculation, the values for the beta parameters are calculated as the regression coefficients, and are inserted in the regression function. Given these parameters and the function for any new evaluation, it should be possible to accurately predict the amount of variance that can be expected for an evaluation within the domain and scope of the technology being evaluated for each variable. This means that at the outset of the evaluation process, the CI of a particular validation unit can be predicted given the values of the independent variables in the evaluation. This result makes it possible to easily determine how reliable an evaluation is, since the lower the deviation, the more reliable an evaluation is, because the CI of the results is proportional to this deviation.

Before the model can be accepted as an accurate predictor of variance in evaluations, the model must be tested on some additional validation units to verify the accuracy of the model. To do this, the evaluator must create another evaluation on some previously unseen data. Based on the variables inherent in this new evaluation, calculate the expected variance for the new evaluation, given the model. Calculate the metrics on the new evaluation and compare the result with the previously established mean of the larger validation corpus, and calculate the mean squared prediction error (MSPR). According to Neter *et al.* (1996:436), "if the MSPR is fairly close to the error mean square (MSE) based on the regression fit to the model, then the model is not seriously biased and gives an appropriate indication of the predictive quality of the model". This means that if the MSPR is close to the MSE, the model accurately predicts future behaviour.

If the model is a good predictor of future results, the model will be able to predict the variance of future evaluations based on the evaluation variables inherent in the particular evaluation. With this information, the evaluator can deduce whether a particular evaluation setup provides results that are reliable. The higher the deviation

of the evaluation, the less reliable the evaluation. The smaller the deviation, the higher the reliability. It is impossible to produce an evaluation that has no variance, but creating evaluations with very narrow variance bands give evaluators a more reliable way of ascertaining the reliability and results reported for a particular evaluation.

2.4 CONCLUSION

This chapter focused on defining the scope of the study by describing what is meant by evaluation reliability and how this relates to the evaluation metrics typically used in NLP evaluations. It further provided an overview of some of the central concepts related to evaluation reliability and the required background on the structure and requirements of evaluations based on existing research. The goal of this chapter was to answer the following two research questions:

a. What is NLP evaluation reliability?
b. How can evaluation reliability be modelled and predicted using linear regression procedures?

The first part of the chapter described the parameters for establishing NLP reliability and some of the causes of unreliable evaluation results. I discussed the existing parameters of evaluation reliability and how this reliability is manifested in evaluations. From this discussion I proposed a definition for evaluation reliability based on the expected future variability of evaluations given the current evaluation and the variables inherent in the evaluation.

The second part of the chapter focused on establishing a stepwise evaluation reliability methodology that outlines the nine different steps in the methodology and how each step relates to existing literature and evaluation methodologies. The methodology consists of the following steps:

1. Define the purpose.
2. Define the type.
3. Define the method.
4. Collect the data.

5. Identify possible variables.
6. Identify relevant metrics.
7. Validate the metric.
8. Select independent variables for each metric.
9. Select and verify the reliability model.

The evaluation methodology can be divided into two separate phases, consisting of the first six steps and the last three steps. The first phase of the methodology is the scoping phase that defines the shared function, goals, resources, and approaches of the evaluation. As part of the scoping phase I discussed the purpose, type, and methodologies of evaluations that are commonly defined in the literature and how each of these relate to conducting reliable evaluations. This was followed by an in-depth discussion of the data collection process, where it is important to collect a balanced and variable corpus that allows the reliability modelling to be as widely applicable as possible. The following types of evaluation data used in the methodology were also defined:

- evaluation corpus;
- modelling corpus;
- modelling set;
- modelling unit;
- validation corpus; and
- validation unit.

The last two steps of the scoping phase focused on the procedure for analysing data to identify possible independent variables and evaluation metrics.

Following the scoping phase is the modelling phase that focuses on modelling the variability of results for each metric by selecting independent variables and regression models that accurately predict variability in future evaluations. This phase starts with the validation of the metrics identified in step 6 and how theoretical and empirical validation of metrics should be done.

Following the validation discussion, I proposed the use of linear regression modelling as a way of determining the validity and reliability of evaluation metrics. Linear regression modelling provides a formal means of expressing the statistical relationship between two or more variables where independent variables are predictors of a dependent variable, through a regression function or model that has predictive capabilities in future evaluations for a particular technology. In this discussion it was shown how FSRP can be used to identify and model the variables that can be used to describe the statistical relationship. Lastly, the procedure for validating the statistical model based on additional validation data was discussed.

The methodology described in this chapter is applied to a well-established NLP technology, spelling checkers, in the following chapter, with the aim of validating the steps and procedures outlined in this chapter.

Chapter 3

3. CASE STUDY: SCOPING A SPELLING CHECKER EVALUATION

3.1 INTRODUCTION

The previous chapter provides the framework for an evaluation methodology that allows evaluators to determine the reliability of their evaluations and predict the reliability of future evaluations. This framework is based on the assumption that more variables than just the number of tokens in the evaluation have an effect on the result. In order to validate this assumption, the proposed evaluation methodology must be applied to various NLP technologies to determine:

a. whether the assumptions about regression modelling of evaluations are viable; and

b. if this modelling can accurately predict the reliability of future evaluations.

This is accomplished by firstly providing an extensive case study on a mature technology, namely spelling checkers, before further validating the methodology on two additional technologies in Chapter 5.

The stepwise evaluation reliability methodology can be split into two distinct phases, namely the scoping phase and the modelling phase. In this chapter, I focus on applying the scoping phase to a spelling checker evaluation. In the first part of the chapter, I look at the purpose of the evaluation and the types of evaluation that are relevant to the type of spelling checker evaluation I want to perform, namely black box adequacy evaluations. Based on this assumption, the methods that other evaluators have used in the past for spelling checker evaluations are discussed, and those methods that are appropriate for black box adequacy evaluations are identified. The methods used to collect and annotate the data are described, before focusing on the inherent evaluation variables that occur in the process.

A central prerequisite of linear regression modelling is that there are one or more independent variables that can act as predictors of some other dependent variable. In the case of evaluations, the dependent variable can be either the metric itself or the variability of the metric when several evaluations are run. The independent variables, on the other hand, are those attributes of the evaluation that change between

different methods and the data that is used in the evaluation. In the second part of this chapter, the two categories of evaluation variables, namely methodological and data-attribute variables, both of which are of interest when validating and establishing the reliability of evaluation metrics are discussed. For each of these categories, the specific variable types that are relevant to the evaluation of spelling checkers are identified. This discussion of each of these variables indicates how previous evaluators approached the variables in existing evaluations, and what the implications of some of these choices are.

In the last part of the chapter, I identify and discuss the metrics that are widely used to evaluate spelling checkers, namely precision, recall, predictive accuracy, and F-measures, and explain how each of these metrics attempts to describe a particular functionality of the spelling checker.

3.2 SCOPING A SPELLING CHECKER EVALUATION

3.2.1 STEP 1: DEFINE THE PURPOSE

The purpose of all software applications is to fulfil a specific user need (King, 1996). With regard to spelling checkers, the most basic need is related to finding and correcting incorrectly spelled words in a text-editing environment (King, 1996; Paggio & Underwood, 1997; TEMAA, 1997). A user expects a spelling checker to flag[7] all and only incorrectly spelled words, and provide corrections or suggestions for possible corrections for those misspelled words. All other functionalities are additional to these basic functionalities, and if a spelling checker does not fulfil this need, all other functionality is arbitrary, as failure to fulfil these basic functionalities would lead people to ignore the spelling checker. For the purpose of this study, these two needs are the central focus of the performed evaluations.

It is important to relate the six quality characteristics identified by King and EAGLES (2.3.1) to the evaluation of spelling checkers. The following is an overview of the characteristics that are of interest to spelling checker evaluations, according to King (1998:6), EAGLES (1995b), and ISO (9126, 1991):

[7] Flags refer to words that are not recognised by the spelling checker, and which are rejected as incorrectly spelled (such as the red underlining in *Microsoft® Office*).

1. Functionality requires that the technology satisfy the user's implied needs.
2. Reliability requires that the technology is stable, does not crash, and has no inherent bugs.
3. Efficiency requires the technology to use as few resources (physical or virtual) as possible.
4. Maintainability requires that the technology can be modified easily, whether to update, improve or adapt.
5. Usability requires the technology to be easily understandable by the implied set of users.
6. Portability requires the technology to be easily transferred from one environment to another
7. Customisability requires that the technology be customisable to the specific user needs of the implied user.

For this study, the evaluations focus exclusively on measuring the functionality attribute of spelling checkers. The evaluations focus on similar products, used in an identical environment, where aspects like efficiency, maintainability and portability, are similar, if not identical. This is because all the spelling checkers that are evaluated are used in the *Microsoft® Office 2010* environment, where the functionality is embedded in and/or limited by the environment. Even though this is the case, some aspects, such as reliability, maintainability, and portability, do differ between spelling checkers. This, however, is not the focus of this study, as these criteria are of more interest to a software design investigation than a linguistic investigation (EAGLES, 1995b).

By using a standardised environment, all identical features can be excluded from the evaluation, focusing more on the linguistic functionality than on the environment. This does not mean that these features are unimportant, or that they should not be taken into account when doing a comparative evaluation where more than one platform is used, such as comparing spelling checkers from *WordPerfect®* and *OpenOffice*.

Secondly, the aim of the evaluation is to accommodate as wide a community of users as possible, while retaining realistic monetary and human input. Furthermore, the

evaluations should be easily reproducible for any given language to get a clear indication of the functionality of the spelling checker in comparison to different spelling checkers of the same language, as well as spelling checkers of other languages.

3.2.2 STEP 2: DEFINE THE TYPE

In the case of the spelling checker evaluations that are performed in this study, the focus is exclusively on doing black box adequacy evaluations. These evaluations do not focus on determining the quality of the individual approaches and architectures of the different spelling checkers, but concentrate exclusively on how well these systems detect errors and do not flag correctly spelled words.

The purpose of an adequacy evaluation for spelling checkers is mainly to determine how well the spelling checker detects (or flags) incorrectly spelled words and provides corrections for the incorrectly spelled words detected, as well as recognising all the correct words of the language. The aim of the evaluations in this study is not on the evaluation of internal modules within the spelling checker, in other words not performance or diagnostic evaluations. Because the evaluation focuses only on a market-ready product, the focus in this study is on black box evaluations.

3.2.3 STEP 3: DEFINE THE METHOD

In order to do reliable evaluations of the attributes identified by EAGLES, a structured methodology has to be presented that is both accurate and widely applicable (King, 1999; TEMAA, 1997). In its evaluation, TEMAA (1997) outlines guidelines for the setup and implementation of evaluations for spelling checkers, but is not explicit in justifying its choice of testing material, methodologies used, or the stability (and therefore reliability) of the metrics that are used in its evaluations. For this reason, the TEMAA methodology is implemented to investigate and validate the accuracy and reported results.

TEMAA's (1997:15) definition of a methodology is more specific to the methods involved in doing evaluations and obtaining metrics than the definitions described earlier (cf. 2.3.3). They indicate that an evaluation methodology refers to the ways in

which values for different attributes are consistently obtained. The steps of the methodology should lend themselves to *repeatability* to obtain values that are *similar* when measuring the same event on the same system multiple times. In spelling checker evaluations, these global objectives relate to how different spelling checkers comply with the six ISO 9126 quality characteristics, namely *functionality, reliability, efficiency, maintainability, portability,* and *usability,* as well as the additional characteristic added by EAGLES (1995b), namely *customisability* (Paggio & Music, 1998:278; TEMAA, 1997:7). Each of these quality characteristics or attributes can then be divided into sub-attributes (TEMAA, 1997:9) that can be measured by the quantitative measures described in 2.3.6 and Chapter 4. The methodology should therefore focus on steps and methods that will realise these objectives.

TEMAA (1997) maintains that part of its evaluation philosophy is the automation of as many components as possible in order to optimise repeatability and minimise the time an evaluation takes. Automation also reduces the influence of human interference in the evaluation process. In its evaluation methodology, the development of automated processes for the entire evaluation process is represented by the parameterisable test bed (PTB) (Paggio & Underwood, 1997; TEMAA, 1997). The PTB creates an evaluation environment in which many of the evaluation and functional components are automated with variable test objective descriptions. The PTB's functionality is described as follows:

> "The PTB is a computer program that is fed with parameters describing objects of evaluation, and classes of users (in a broad sense) of these objects; consults a library of test methods; carries out the relevant tests; and produces the evaluation report." *(TEMAA, 1997:15)*

With TEMAA's methodology in mind, this study also investigates the possibility of implementing automated evaluation procedures and modules in the methodology proposed in the following chapters.

Although the TEMAA framework has been validated by follow-up evaluation studies (Paggio & Music, 1998; Paggio & Underwood, 1997), neither of these studies has critically evaluated how the TEMAA methodology and the underlying decisions, such as data selection, automatic error creation and so on, affect the results of these evaluations. Specifically, none of the variables that are inherent in evaluations are discussed or investigated in either of these studies, and preliminary results on

applying this methodology to Afrikaans have shown some possible flaws or oversights in the construction of their evaluation methodology. With this in mind, Chapter 4 critically evaluates the assumptions and implementations to re-evaluate the methodology and make proposals for adjustments and improvements that will ensure more reliable evaluations for spelling checkers and possibly for a wider range of NLP technologies.

3.2.4 STEP 4: COLLECT THE DATA

The collection of data for the evaluation of spelling checkers has two main focal points. The first is ensuring that the data is balanced across multiple domains and genres in order to get a good and large enough sample of data to attempt to accurately reflect the full written language. The second important consideration is the definition and consequent annotation of incorrect words. After collecting the evaluation corpus, the data is split into two corpora with an 80:20 ratio, based on the guidelines described in 2.3.4 to form the modelling and validation sets.

As a model for designing the spelling checker evaluation corpus, the construction of the British National Corpus (BNC) is used as a guide. The reason for choosing the BNC is that it is widely held to be one of the most comprehensive and representative corpora currently in existence (Bianchi, 2012; Fillmore *et al.*, 1998; Galliers & Sparck-Jones, 1993:127; Leech, 2006; Zanettin, 2011). Table 2 outlines the stratification of the corpus that is used in the collection of the evaluation corpus.

Style	Sub-divisions	Percentage
Student writing	Student essays (10%)	10%
E-mail	Personal correspondence (7.5%) Official correspondence (7.5%)	15%
Academic	Humanities (5%) Social sciences (5%) Natural sciences (5%) Technology (5%)	20%
Popular	Humanities (10%) Social sciences (10%) Natural sciences (10%) Technology (10%)	20%
Reportage	Newspaper articles (10%) Popular magazine articles (10%)	20%
Instructional	Administrative writing (10%) Skills/hobbies (10%)	15%
Creative	Novels (10%)	10%

Table 2: Distribution of a stratified corpus based on the BNC

Defining the correct and incorrect words in an evaluation corpus seems like a relatively simple task. There are, however, some instances where determining whether a word is correctly spelled poses difficulties. For instance, languages where a standard written variant has not been officially established can produce differences in classification by different language users. This problem is especially significant for languages where the written standards have been underdeveloped for political or socio-economic reasons, such as isiZulu, isiXhosa, Setswana, Sesotho sa Leboa, Igbo, Yoruba, Wolof, and Inuktitut. Because all languages are constantly evolving and new words are being accepted into the language while others are changed when reforms take place, some words and variants in standardised languages such as French, German and Dutch are also difficult to classify as correct or incorrect.

These problems aside, for purposes of evaluating spelling checkers, a correct word can be defined as follows:

> *A correctly spelled word is any sequence of characters that conform to the orthographical spelling standards and morphological structure of the language in question, irrespective of context or meaning.*

Incorrect words can be defined as follows:

> *An incorrectly spelled word is any sequence of characters that does not conform to the morphological and orthographical structure of the language, including cognitive, typographical, phonological errors, and foreign words not accepted as part of the standard written variant.*

There are a number of issues that should be noted when classifying words according to these definitions. Firstly, it must be established what standard will be used as the standard written variant, and all decisions should be based on this standard. The ideal with regard to this standardisation is to make use of the guidelines of a governing language body which has set the standards for the language. In cases where uncertainty about the spelling of a word exists, guidelines from these bodies should be used. As an example, the word *chakalaka/tjakalaka* ("spicy sauce") has been in use for a number of years in Afrikaans, yet no standard spelling had been accepted until August 2006. Before this, the words would have been marked as incorrectly spelled words, but now they should be classified as correctly spelled.

Secondly, it is important to note that spelling errors do not refer to context errors, and words should be classified based purely on whether the word exists in the language or not. In the sentence "*My shoes are* ***to*** *big*.", ***to*** should not be marked as a spelling error, as this word is a correctly spelled word, even though the context of the word is incorrect.

With regard to meaning, words which are semantically improbable but still correctly spelled according to the standard orthography, can also sometimes cause classification difficulties. As an example, the Afrikaans word **deurgans* (literally *door goose*), which is most probably a misspelling of *deurgaans* (*throughout*), could be a correct compound referring to a doorstop in the shape of a goose, and therefore be a correctly spelled word. These words should not be classified as spelling errors, but be accepted as correct words. On the other hand, words where no possible correct context can be found, such as **plantkarglashond* (*plant car glass dog*), where it is morphologically possible for all the elements to form a compound, but the meaning is improbable, these words should be marked as errors.

One issue which is not specifically referred to in the definitions is named entities (such as proper place and geographical names), because they can also cause classification difficulties. Users generally expect spelling checkers to recognise the most common proper names of the language, while flagging incorrectly spelled

versions of those names (e.g. *John* and **Jogn*). Unfortunately, deciding when a proper name is "common" can lead to considerable disagreement, and many "common" proper names have a number of different spellings, all of which are considered correct. As an example, the very common Afrikaans name *Johan*, can also be spelled as *Johann*, both of which should be recognised. Misspellings of these two names should however be flagged (e.g. **Jphan*). From the perspective of the user, these words should be correctly handled, as these words are also part of the texts that users create. With this in mind, it is suggested that all proper names should be included in the evaluation, no matter how uncommon a name may be in a language. Incorrectly spelled proper names should also be classified as such, and where there is uncertainty about the spelling of an uncommon name, it becomes the responsibility of the person who classifies the words for the evaluation, to determine what the correct spelling is.

3.2.5 STEP 5: IDENTIFY POSSIBLE VARIABLES

In order for statistical analysis to be used in modelling the reliability of evaluation metrics, it is important to identify as many of the variables in the evaluation procedure as possible. By identifying and isolating variables, the reliability of the evaluation metrics can be tested, by investigating which variables influence each metric. With a better understanding of the relationships between variables and metrics, it is possible to propose evaluation methods that give reliable evaluation results.

There are two main types of variables in NLP evaluations, methodological variables and data-attribute variables. Methodological variables are different procedures that are followed in constructing and performing evaluations that can directly or indirectly influence the results of an evaluation. This methodological variety is not always easily described with a set of statistical variables, but is crucial to how data variability is manifested in evaluations, since the procedures for creating an evaluation almost always have a direct effect on the characteristics of the data. Therefore, this methodological variance is described to gain a better understanding of how they affect the data that is used in the evaluations.

Data-attribute variability refers to the variance of inherent characteristics of the data used in evaluations. These characteristics are usually quantifiable attributes or properties of the data that vary between different modelling units. These attributes can be used as independent variables that influence the variability in metric results of NLP evaluations. Since methodological variety and attribute variability are intertwined, items from both types are discussed in the remainder of this chapter.

The process of identifying possible variables depends on a good understanding of the technology that is being evaluated, and establishing what the factors are that will affect the evaluation. Some of these variables are obvious, such as size and quality, while other variables may be obvious but are not easily measurable, such as text complexity. Still others require a deeper understanding of the target of the technology and the metric that is evaluated. In general, it is best to identify as many quantifiable variables as possible, and let the linear regression modelling process determine which of the variables are actually relevant to a particular metric's reliability.

A number of variables are inherent in spelling checker evaluation, and are mostly concerned with the test materials and methods used in the evaluation. The following list of variables contains most of the quantifiable variables associated with spelling checker evaluations. The nature of these variables, as well as the reasons for including them, is described in detail in the following section.

Methodological variables:

- Test suites vs. test corpora (3.2.5.1.1);
- Usage-based vs. automatically generated data (3.2.5.1.2); and
- Types vs. tokens (3.2.5.1.3).

Data-attribute variables:

- Modelling unit size (3.2.5.2.1);
- Type-to-token ratio (3.2.5.2.2);
- Heterogeneity (3.2.5.2.3);
- Target density (3.2.5.2.4);
- Error types (3.2.5.2.5); and

- ❖ Quality of the technology (3.2.5.2.6).

The following discussion of variables provides a concise description of each of the variables, as well as the way that each of these variables can influence the results of spelling checker evaluations. This initial discussion is used later as the basis for conducting evaluation experiments (cf. 4.2-4.6) and regression modelling, to determine the statistical influence each variable has on the outcome of an evaluation. In these experiments, each of the described variables is implemented in a number of different ways, in order to determine which approaches and methods yield the most reliable results.

3.2.5.1 Methodological variables

3.2.5.1.1 Test suites vs. test corpora

Test suites are automated testing environments where data is constructed to test different situations and linguistic phenomena at different levels of abstraction (Lehmann *et al.*, 1996:2). This refers to the features of a test suite where the data is deliberately divided into different kinds of language instances and phenomena, in order to track how these instances are handled by a system and by different subsystems within a software system (Chapter 2). According to Prasad & Sarkar (2000:2), a test suite is a constructed database of input-output pairs that are either correctly or incorrectly handled by a system. There is some overlap between test suites and automatically generated data (cf. 3.2.5.1.2), since many of the test procedures and data in test suites can be generated automatically (TEMAA, 1997). There is, however, no prerequisite for automatically generated data in test suites, and data is often manually generated to adhere to the principles of test suite design.

Test corpora, on the other hand, refers to the use of large amounts of naturally occurring language data and language phenomena as used by people in real-world situations (Balkan *et al.*, 1994: 4). Fillmore *et al.* (1998:3) define a corpus as "a collection of naturally occurring language text, chosen to characterize a state or variety of a language", while Starlander & Popescu-Belis (2002:269-270) consider that the use of an evaluation corpus is more representative of actual users than the use of test suites. They point out that by using test corpora, the frequency of different correct and incorrect words can be more accurately represented, which is one of the

shortcomings of the TEMAA framework, as identified by Paggio & Music (1998:279).

Paggio & Music (1998:279) further draw attention to the fact that by using a corpus of naturally occurring data, an evaluator can get a good variety of data. In addition, they indicate that because the data has not been generated, the evaluation more closely resembles the conditions in which the technology is used, and thus more accurately mimics the experience of a user. For these reasons, a corpus lends itself more to adequacy evaluations than do test suites, as it tends to more accurately test the overall performance of a system from a user's perspective, rather than evaluating specific linguistic phenomena.

Most of the research on evaluation methodologies, such as DiET (Netter *et al.*, 1998), ELSE (Lenci *et al.*, 1999), test suites for natural language processing (TSNLP) (Lehmann *et al.*, 1996), NIST (Martin & Przybocki, 2000; NIST, 2002), Mediterranean Arabic language and speech technology (MEDAR) (Olivier Hamon *et al.*, 2012) and specifically for spelling checkers, such as SCARRIE Lenci *et al.* (1999) and TEMAA (1997), has focused on the development of evaluation test suites. According to Lehmann *et al.* (1996:2) and Balkan *et al.* (1994), there are mainly four reasons for using test suites rather than test corpora:

- *Control over data*, which implies that the linguistic phenomena that occur in the data can be introduced to evaluate specific modules within a system. The test items are specifically chosen to test different linguistic phenomena in the language as well as the individual underlying models in the technology. As an example, a test suite selects specific valid and invalid compounds to test the compounding functionality, irrespective of their actual occurrence in usage-based data.
- *Systematicity* refers to depth of coverage of the test items in terms of well-formed and ill-formed instances. The ill-formed items are derived by introducing systematic variations of the well-formed instances. By doing this, both the coverage and over-generation of a system can be comprehensively evaluated. As an example, when evaluating a grammar checker, one can replace all instances of the word "a" in the evaluation data with the word "an", and evaluate these variations to ensure that all incorrect instances are flagged.

- *Inclusion of negative data* is a feature of test suites that refers to the use of incorrect data, which is used to evaluate specific aspects and possible shortcomings, such as over-generation. An example of this would entail including incorrectly formed inflections in the data when evaluating a spelling checker, for example runned/sitted/writed.
- *Exhaustivity* refers to the fact that a test suite consists of a limited number of specific instances, and can therefore never test anything that is not part of the constructed evaluation corpus.

Bernsen & Dybkjær (2000:4) indicate that test suites are most useful in the evaluation of sub-components of a system independently of the rest of the system. The implication of the test suite approach is that the data sets are kept separate, in order to track progress between different stages in the development process. The fact that test suites focus on the sub-components of a system, means that this type of evaluation data is more suited to performance or diagnostic evaluation than adequacy evaluation.

In a spelling checker test suite, for instance, incorrect words that are used during the evaluation can be categorised according to the nature of the error in the word. These categories can then be used to determine how reliably a given system recognises, or provides suggestions for, each type of incorrect word. Each of the phenomena represented in the test suite can then be scored as a very specific sub-attribute, as presented in the results of Paggio & Underwood (1997), where error detection is evaluated on a per error type basis. In this manner, each error type has an "error coverage" figure associated with it, as shown in Table 3. The figures in this table represent the percentage of incorrect words in each of the error type categories that are flagged by the respective spelling checkers.

Product	***r* errors**	**Suffixation**	**Silent letters**	**Substitution**	**Loan words**
A	100%	100%	99%	100%	100%
B	75%	100%	96%	96%	100%

Table 3: Recall incorrect results on different error types for two Danish spelling checkers taken from Paggio & Underwood (1997:14)

Such an evaluation would not just score recall incorrect as a single metric, but rather the recall incorrect of the system on specific phenomena represented in the data, such as incorrect morpheme use, incorrect compound acceptance, and so on.

The main shortcomings of using test suites is that one does not necessarily get an accurate indication of a system's capabilities when applied to unrestricted datasets because of the restricted and controlled environment of a test suite (Prasad & Sarkar, 2000:4; Galliers & Sparck-Jones, 1993:125).

Even though it would appear that the use of test corpora, rather than a test suite, is the most suitable approach for adequacy evaluations of spelling checkers, the use of test suites cannot be discounted from adequacy evaluation strategies without additional proof that the results are not reliable representations of functionality. For this reason, both test corpora and test suites are used in the evaluation experiments in the next chapter, in order to find more definitive proof that either of the approaches is more reliable in its reflection of linguistic adequacy.

3.2.5.1.2 Usage-based vs. automatically generated data

A second methodological distinction that can be made is the difference between using usage-based data and automatically generated data. This is different from the distinction between test suites and test corpora in that even in test suites, the origin of the examples is usage-based data; it is just that the items included in the data are controlled, and the sparsity that occurs naturally for some phenomena in user data is overrepresented in test suite data. Automatically generated data, on the other hand, is a systematic way of creating evaluation instances that aim to mimic the behaviour of the user, while not requiring explicit annotation of the data instances, which make the process of data gathering much cheaper. A good example of this is the ErrGen module developed by TEMAA (1997), which generates incorrectly spelled words based on a set of rules that mimics the types of errors which Danish people typically make.

As was mentioned in 2.3.3, one of the major priority shifts in the evaluation of NLP applications over the last two decades has been the focus on the user's experience of a system, especially within the EAGLES/TEMAA framework (King & Maegaard, 1998). Manzi *et al.*, (1996:3) state that the explicit focus of the TEMAA evaluation

methodology was to evaluate products with respect to their adequacy for particular users. User experience is an important aspect of the evaluation of market-ready products, because the adequacy of a system has to fulfil a specific user's needs. The focus on the automation of evaluation tools, workbenches and methodologies that are currently widely used, has been of interest in the NLP community for several years (Balkan *et al.*, 1994; Lehmann *et al.*, 1996; Lenci *et al.*, 1999; Lin & Hovy, 2003; Murthy *et al.*, 2008; Netter *et al.*, 1998; Paggio & Underwood, 1997; Papineni *et al.*, 2002; TEMAA, 1997).

Automating the evaluation process is beneficial to researchers on a number of different levels, especially by increasing the usability, reproducibility, and accuracy of evaluations, and specifically includes the automation of data generation wherever possible. These two approaches to evaluation data are methodological differences, and therefore the difference in results is the result of methodological variables. The evaluations in the following chapter that describe the variance between the approaches not only consider the results in a single approach to data collection, but also conduct variability investigations between the two approaches.

King & Maegaard (1998:228) hold that users are described in the same way as systems, namely in terms of features and values. These feature and value pairs are then used to determine whether a system's output corresponds to a user's needs. One of the TEMAA design prerequisites was to enable the evaluation system to be adaptable when a system is evaluated for different users (King & Maegaard, 1998). They implemented this as a specific changeable attribute in the system design, enabling the evaluator to change specifications with regard to the attributes and corpus design when evaluating a system for a specified user, such as a legal representative or a journalist.

Although the adaptability of the evaluation to a specific user's needs is a valuable design criterion, changing the attribute set for every evaluation means that a very large set of test materials would need to be developed, significantly increasing the cost and time requirements of maintaining such a system. As one of the secondary aims of this study is finding ways to minimise the cost and time requirements of evaluations, other methods of representing the user need to be investigated. However, in order to attain this aim, the user has to be explicitly defined, in order for him/her to be represented by the data that is used in the evaluation.

The focus of adequacy evaluations for spelling checkers is to represent as many users as possible in a text-editing environment such as *Microsoft® Office*. The users of text editors vary greatly in their use of spelling checkers and the types of texts that they create, but the following groups of people would typically use these text editors:

- home users – writing letters, e-mails, and journals;
- information workers – creating presentations, memoranda, e-mails, official letters, and reports;
- academics – writing reports, scientific articles, lecture notes, and presentations;
- students – writing essays and creating presentations; and
- writers – writing fictional and non-fictional narratives.

It is evident from this description of the users that there is a very wide variety of different users, with different writing styles, who make use of spelling checkers. For this reason, it is imperative that the testing material that is used for the evaluation of the spelling checkers should represent as many of these different users as possible. Section 2.3.4 presents a detailed description of the construction of test materials in order to attain this representativity.

Automatic data creation and evaluation, on the other hand, have a slightly different focus in the approach to data compilation for evaluation. TEMAA (1997) places a high value on the automation of as many of the evaluation procedures and methods as possible. Lenci *et al.* (1999), Netter *et al.* (1998), and TEMAA (1997) cite the following reasons for their focus on automation:

- *Systematicity*: Automated evaluations make the process of constructing and systematically testing every attribute of a tool consistently and repeatedly, much easier.
- *Large sample coverage*: Because little or no human input is necessary after the initial data collection and annotation, it is possible to evaluate language samples that are much larger than would otherwise be practical.
- *Speed*: Automated processes are for the most part quicker and more able to sustain a constant work speed, whereas human evaluators cannot maintain a constant high level of performance for sustained periods of time.

- *Objectivity and consistency*: Although an evaluation should always be objective, human interaction with evaluation data often tends to lead to changes in evaluation data that are not necessarily applied consistently to all evaluations (cf. 3.2.5.2.2). Automated evaluations make no distinction between one evaluation item and the next, and can therefore not be influenced to change evaluation criteria based on a particular entry, only where the change is consistently made to all entries.
- *Human error*: In accordance with the previous two criteria, human errors often hinder reliable evaluations. Whether deliberately or unintentionally, human evaluators can be inconsistent in their evaluations and their assignment of attribute scores or categorisations of evaluation data and system responses (Netter *et al.*, 1998; TEMAA, 1997). Automated procedures are much more systematic, and if inconsistencies exist, these can often be easily identified and rectified.
- *Cost*: Although the initial cost of developing automated testing and evaluation procedures is often resource-intensive, in the end, automated evaluations save both time and money, if they have been well designed and thoroughly investigated before implementation.

It is important to note that the two approaches to data that were discussed in this chapter are not necessarily mutually exclusive, and that a combination of usage-based and generated data could be implemented in order to get reliable results. Because the different sub-attributes of spelling checkers also require different approaches to evaluation, using both types of data in a single evaluation can be justified in order to attain reliable evaluation results. In the next chapter, the influence of using the different types of data on an evaluation is determined in the stability of the metrics that are produced by using each of the data types (cf. 4.3).

3.2.5.1.3 Types vs. tokens

Differentiation between types and tokens is a common one in the field of both corpus linguistics and NLP research in general (Jurafski & Martin, 2000:195). Tokens refers to the total number of words in a text, while types refers to the unique (or distinct) words in the text. In different evaluations in the past, both of these

approaches have been used in spelling checker evaluations (Paggio & Music, 1998; Paggio & Underwood, 1997; TEMAA, 1997), but the effect of using one approach over the other in terms of descriptiveness and reliability has not been investigated.

When evaluations are performed, the evaluation results can be reflected either as the number of unique words which are recognised by the spelling checker (i.e. the types), or as the total number of words that are recognised in a piece of text, taking into account the frequency of words (i.e. the tokens). Part of the investigation is concerned with how the use of types or tokens influences the different metrics, as well as the implications this has for the reflection of user experience.

An evaluation using only word types will reflect the coverage of the different words in a language, while a token evaluation reflects the way running text, as a user would produce it, is covered by the spelling checker. If a spelling checker therefore flags a correct word in a type evaluation, it will only be penalised once for not recognising the word, but it also only receives one mark for recognising a high frequency word. In the evaluation of this variable, the influence on metric stability using each of these approaches is tested, also keeping in mind that the evaluation should reflect the way a user sees the functionality of the spelling checker.

A second aspect of types and tokens is the ratio between the two values in a particular modelling or validation unit. TTR is a long-standing, though contentious, measure of linguistic diversity (Covington & McFall, 2010; Popescu *et al.*, 2011). This ratio relates the number of types to the number of tokens in a particular text, where the higher this value is, the more diversity there is in the text. This measure can be used to represent a notion of text variety in evaluations that may be able to provide an additional text characteristic that can predict evaluation variability.

The two aspects of type and token usage address different aspects of evaluation variability, where using either type or tokens as the unit of evaluation introduces methodological variability, while the TTR of a text is an aspect of data-attribute variability. Each of these two aspects is evaluated and investigated in the next chapter as possible predictors of evaluation variability (cf. 4.2.2.1).

3.2.5.2 Data-attribute variables

3.2.5.2.1 Modelling unit size

To date, there are no fixed rules for determining the overall size of any corpus for a particular purpose (Chung, 2003:223; Sinclair, 2004:13). Although Sinclair (2004:9) states that there is insufficient scientific evidence for selecting samples that are of similar size for comparative purposes, it has been asserted by other researchers, such as Chung (2003:227), that the size of the corpora should be similar when doing comparative analysis. By having corpora of similar sizes, the underlying principle of equal sample size, in order to minimise the standard error of the difference between the texts, can be followed (Chung, 2003:228).

As El-Emam (2000:21) points out, the empirical validation of metrics can optimally be done when the sample size of the validation set is large, as this increases the reliability of the statistical significance that is measured for each metric. From this it follows that a large number of texts have to be spellchecked in order to get large samples, in which the significance of differences in metric measurements can be accurately determined (El-Emam, 2000:21). However, large corpora are expensive to design, create, and maintain, especially for evaluations where the usefulness of a corpus is very often limited to a specific technology. For this reason, it is important to find an optimal corpus size that both yields reliable results and is relatively cheap to develop and maintain. Specifically, the present study aims to find a way of calculating the minimum number of elements that can be used to carry out a reliable evaluation.

Related to the number of words that form the modelling unit is the proportion of different linguistic phenomena that appear in the text, such as the ratio of correct and incorrect words, types to tokens, morphologically complex words, terminology, and so on. Although some of these proportions are captured by other measures that have already been discussed, such as TTR and indexes of synthesis and fusion, some others do not have such explicit ratios. It should still be possible to use some measure as a variable in the evaluation reliability testing. As many of these different linguistic ratios as possible are included and discussed in the remainder of this section.

3.2.5.2.2 Type-to-token ratio

The TTR of a text refers to the ratio of unique words to the total number of words in a text. In most cases, as the size of the text increases, so the ratio of new unique words that are introduced decreases, and the TTR of the text also decreases. This variable relates to both text complexity and language complexity, since the more unique words are contained in the text, the more likely it is that the text is complex, while simpler text usually re-uses the same or a limited vocabulary. Similarly, morphologically complex languages and conjunctively written languages tend to have a higher incidence of unique words, and therefore a higher TTR. Since each text will have a unique TTR score depending on the vocabulary used in the text, this may be a good independent variable for predicting evaluation variability.

3.2.5.2.3 Heterogeneity

Although this study aims to be as comprehensive as possible in the evaluation of metrics and the influence of variables, text difficulty is not given specific attention. The main reason for this is that there is little consensus outside of English for how to calculate text difficulty, or how well these measures represent the difficulty of the text. Furthermore, the existing measures, such as the Flesch reading ease score, Fogg index, and the Dale-Chall formula for English (Bailin & Grafstein, 2001:285), are not always applicable to other languages; the investigation of readability measures for different languages falls outside the scope of this study, but should be investigated in future research.

Instead of attempting to establish text difficulty, I do include a measure of unit heterogeneity, that is, the level of variety in a selected modelling or validation unit. With this in mind, a simple scalar representation of unit heterogeneity, with values between 1 and 10, is used. This scale is a simple representation of the number of different domains of texts used in the construction of the modelling unit. Modelling units that were selected from a single domain have a score of 1, while units selected from 5 domains will have a heterogeneity score of 5. When selecting data for the modelling and validation units, no unit was selected from more than 10 domains, limiting the maximum homogeneity score to 10.

3.2.5.2.4 Target density

Another corpus-inherent attribute is the density of the target items in the corpus. The target items for each metric will be slightly different, depending on the attribute being measured by a particular variable. Target density should always be expressed as a fraction between 0 and 1, where the number of targets is related to the size of the entire unit. An example of target density in spelling checker evaluations is the percentage of errors in the text. If a text has 1.5% of errors, the target density for recall incorrect and precision incorrect is 0.015. Inversely, the target density for recall correct and precision correct is 0.985. The density of the targets could potentially influence both the value and the reliability of the metrics, since higher density usually means that the technology has more opportunities to perform its function, and possibly produces results that are more reliable.

3.2.5.2.5 Error types

As one of the central aspects of spelling checker functionality, the incorrect words that are used in the evaluations are of critical importance. The errors used in the evaluation of spelling checkers take a number of different forms, and there are a number of different ways of obtaining these errors. Different types of errors, the origin of these errors, and the ratio in which different error types occur in the evaluation corpora, can all have an effect on how accurately the evaluations related to these variables perform.

The first distinction that can be made is based on the work of Damerau (1964), Kukich (1992) and Verberne (2002), which describes errors in terms of spelling errors (also referred to by Kukich as cognitive errors), a special class of cognitive error, namely phonetic errors, and typographical errors. Spelling errors, according to Kukich (1992:387), are errors caused by a lack of knowledge or a misconception about the spelling of a particular word (e.g. **recieve*, **beutiful*). Phonetic errors are errors where the user substitutes a phonetically correct but orthographically incorrect sequence of characters in a word (e.g. **scarie*, **skorn*). Typographical errors are instances where the spelling of a word is known, but a "motor coordination slip" by the user causes an error (Kukich, 1992:387). The most common of these types of error were originally identified by Damerau (1964) as single-error misspellings, and can be classified into one of the following four categories:

- insertion – where an additional letter is inserted into the word, e.g. *categories* → **cartegories;*
- deletion – where a letter is omitted from the correct spelling, e.g. *spelling* → **speling;*
- transposition – where two adjacent letters in a word are interchanged, e.g. *height* → **hieght;*
- substitution – where a character in the word is replaced by another character, e.g. *word* → **qord.*

Although errors with more than one of these types of error do occur, they are not nearly as frequent as single-error misspellings (Damerau, 1964; Peterson, 1986; Pollock & Zamora, 1984). The type of errors that occur in an evaluation text are of interest as a variable where implementations of generated data become a possibility (specifically with error flagging, suggesting correct words and automatic correction).

3.2.5.2.6 Quality of the technology

The last evaluation variable for spelling checkers that is considered in this study is the quality of the technology with regard to the metric that is being evaluated. Although the purpose of the evaluation is to determine the quality of the technology, the error rate of the technology also affects the variability that can be expected in the evaluation. It is impossible to know what the actual error rate of the technology is, and the only indicator of this rate is the result of the evaluation (Mitchell, 1997). The fact that this is an estimate of the actual quality is not a problem, since this is just one of several variables that are considered for predicting the variability. This is similar in principle to the error rate that is used as part of the expression of interval estimate for error (Mitchell, 1997: 132-133). In principle, the higher the quality of the technology in respect to a particular metric, the lower the variance in evaluation results for that metric.

Since the quality of a single spelling checker on each of the different metrics can be different, the quality of the spelling checker with regard to each metric must be considered separately when estimating the influence of the quality on variability in evaluation results.

Since each of the variables discussed in this section could potentially influence the various evaluation metrics, each of these variables will be considered as an indicator of variance during the modelling stages of the methodology. For each of the metrics, the variables that are good predictors of variability are identified and used in the predictive model for that metric.

3.2.6 STEP 6: IDENTIFY RELEVANT METRICS

The aim of spelling checker evaluation metrics is to distinguish between different systems, by calculating the linguistic functionality of the system quantitatively in terms of user requirement. The metrics will therefore be measures of attributes such as the coverage of the language, the recognition of incorrect words, the appropriateness of suggestions, and so forth.

According to King & Maegaard (1998:2), the functionality quality characteristic (or attribute) of spelling checkers can be categorised in the sub-attributes recall, precision, and suggestion adequacy. Each of these sub-attributes is evaluated by a quantitative measurement (i.e. metric) that reflects the performance of the product with regard to the measured attribute. These sub-attributes also form the basis of most spelling checker evaluations performed over the past two decades (e.g. EAGLES 1995a; Starlander & Popescu-Belis, 2002; TEMAA, 1997).

As premise for the calculation of the spelling checker evaluation metrics, Starlander & Popescu-Belis (2002:270) and (Reynaert, 2006) describe as follows the four possible classifications that a spelling checker makes:

- **True positives** (*Tp*) are those instances where the spelling checker accepts a correct word;
- **False negatives** (*Fn*) are all cases where the spelling checker flags a correct word as incorrect;
- **True negatives** (*Tn*) are those examples where an incorrect word is flagged as invalid by the spelling checker; and
- **False positives** (*Fp*) are instances where the spelling checker accepts an incorrectly spelled word as correct.

These four categories of classification form the basis for all the evaluation metrics that are described in Chapter 4. In the evaluation process, each word that is used as

an input word is categorised according to the classification made by the spelling checker.

As an example, the small set of English words in Figure 3.2-1 illustrates how words are classified according to the response of a spelling checker. The words inside the circle are correctly spelled English words, while the words outside the circle are incorrectly spelled words. The spelling checker should recognise all the words inside the circle, while flagging all the words outside the circle.

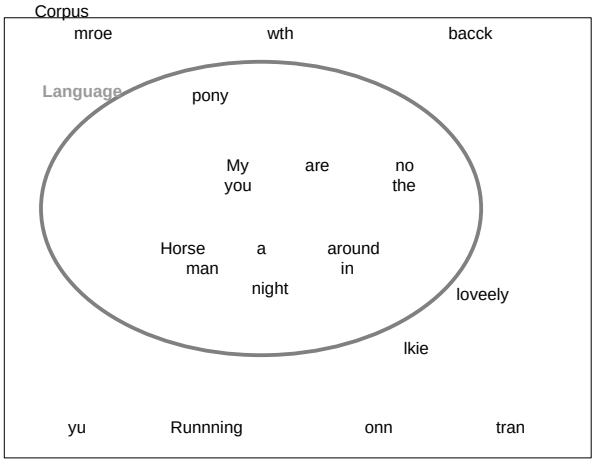

Figure 3.2-1: A sample set of English words with valid words grouped together[8]

The words in the darker circle in Figure 3.2-2 represent the words that a hypothetical spelling checker accepts as correctly spelled words. These include incorrectly spelled words (**lkie* and **loveely*), and excludes the correctly spelled word, *pony*. According to the classification given by Starlander & Popescu-Belis (2002), **lkie* and **loveely* are false positives, while *pony* is a false negative. The other words in the dark circle are all true positives, and words outside both circles are true negatives.

[8] Courtesy of Dermot McLoughlin

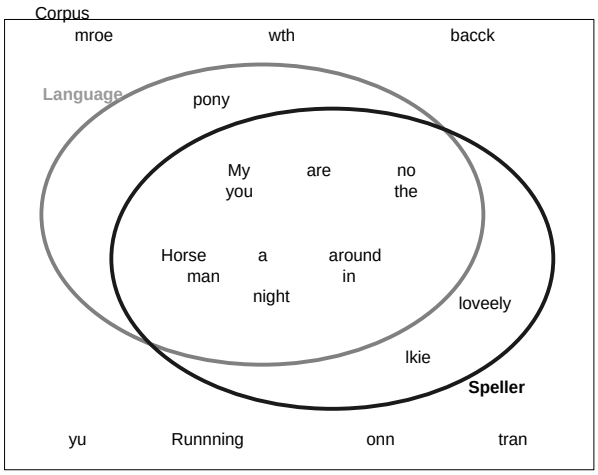

Figure 3.2-2: Hypothetical response of a spelling checker on a sample set of English words

During the evaluation process, words are classified into one of the four categories represented in the figures, and the aim of the evaluation is to determine how closely the spelling checker approximates the language in its classification of words. These approximations can be quantitatively measured by metrics such as recall and precision which are discussed in detail in 4.2-4.6.

Recall is widely accepted to represent a system's ability to handle instances of a given attribute correctly (TEMAA, 1997:31, Reynaert, 2006). For the evaluation of spelling checkers, recall is mostly used to refer to the acceptance of correct words as correct, and incorrect words as incorrect (Paggio & Underwood, 1997:14; Starlander & Popescu-Belis, 2002:271; TEMAA, 1997:31). These two recall measures are referred to as the recall correct (R_c) and recall incorrect (R_i) of a system (see 4.2 and 4.3).

Precision, according to TEMAA (1997:13), is "the degree to which a spelling checker rejects incorrect words", or as Starlander & Popescu-Belis (2002:271) explain, the accuracy of a system concerning the task it is set. For spelling checkers, this implies that only incorrect words should be flagged as incorrect, and no correct words should be flagged. This measure therefore determines how accurately the spelling checker flags words.

Starlander & Popescu-Belis (2002:271) give a refined view of precision, by dividing precision into precision incorrect (P_i) and precision correct (P_c), as opposed to using only P_i as TEMAA does. Even though TEMAA does not make use of this metric, it is included in the following discussions in 4.5 to determine its validity and descriptive usefulness.

Another core functionality of spelling checkers is providing corrections for detected errors. The suggestion adequacy metrics (Paggio & Underwood, 1997; TEMAA, 1997; Van Huyssteen *et al.*, 2004; Van Zaanen & Van Huyssteen, 2003) measure how often the spelling checker provides the correct intended word as a suggestion, and in which position the intended correction appears in the list of suggestions. There are different scoring systems for measuring this functionality, which is discussed in detail in 4.6.1.2.

Based on the basic metrics, Starlander & Popescu-Belis (2002:271) have also introduced three new metrics. These new metrics can be described as combinatory metrics that reflect the efficiency of the spelling checker on more than one sub-attribute. These metrics are:

- harmonic mean on correct words;
- harmonic mean on incorrect words; and
- predictive accuracy.

The metrics outlined by Starlander & Popescu-Belis (2002) are very specific with regard to each sub-attribute of the spelling checker, and give values to most of the different sub-attributes of the spelling checker process (excluding suggestion adequacy). Because of the specificity of the metrics and the inclusion of metrics for most of the different sub-attributes of spelling checking, the results that are obtained from these metrics are more descriptive of the linguistic functionality and adequacy of a spelling checker. All of the metrics used by Starlander & Popescu-Belis (2002) are included in the present study (cf. 4.2-4.6).

A final set of metrics that has received some attention in recent years is the development of a metric that reflects the overall linguistic adequacy of spelling checkers (Riley *et al.*, 2004; Starlander & Popescu-Belis, 2002; Van Huyssteen *et al.*, 2004). Although these three studies attempt to relate all of the attributes inherent in an evaluation into a single metric, there is still no clear indication as to which one of the metrics is the most reliable and descriptive one for evaluations. In the final part of Chapter 4, these metrics, as well as other possible overall metrics, are investigated to determine their relevance and reliability.

3.3 CONCLUSION

This chapter applied the first part of the evaluation reliability methodology, the scoping phase, to a spelling checker evaluation. The first part of the chapter focused on discussing the purpose and type of evaluation that is validated in this case, followed by a discussion on the different methods that are appropriate for the type of black box adequacy evaluations that are the focus of this study. In this regard, a mixture of usage-based and automatic procedures were proposed, based on the work previously done for the TEMAA (1997) evaluation project.

The next part of the chapter focused on the collection of data and identification of evaluation variables. The data collection process for the spelling checker evaluation focused on the process of finding and constructing balanced corpora that draw from as many different genres and writing styles as possible. After collecting the data, all incorrect words in the evaluation corpus were annotated with their expected corrections. After annotating the evaluation corpus, the data was split 80:20 into two distinct groups, namely the modelling and the validation corpus. The modelling corpus forms the basis of the data that will be used in steps 7 & 8 of the reliability methodology, while the validation corpus will be used in validating the proposed regression models for each metric.

Since the use of linear regression modelling is centrally dependent on the identification of relevant independent variables, the next part of the chapter identified the evaluation variables that are relevant to spelling checker evaluations. This discussion showed that there are two categories of evaluation variables in spelling checker evaluations, namely methodological variables and data-attribute variables. In each category, several variables specific to that category were identified and discussed, to provide the theoretical framework for the evaluation experiments that are performed in the next chapter.

In discussing each of these variables for spelling checker evaluations, those areas that are most likely to cause evaluation variability and therefore prevent reliability were identified. In the next chapter, these evaluation variables, both methodological and data-attribute, are used as the basis for hypotheses and experiments in the validation of spelling metric evaluation metrics.

The last part of the chapter identified and discussed the most widely used metrics that are used in the spelling checkers evaluations. The discussion mainly focused on the definition and background of the recall, precision, F-measure and predicative accuracy metrics. These metrics are well understood and defined, and have been used in various evaluation projects (EAGLES 1995a; Riley *et al.*, 2004, Starlander & Popescu-Belis, 2002; TEMAA, 1997) and form the basis of the modelling steps of the reliability methodology performed in the next chapter.

4. CASE STUDY: MODELLING SPELLING CHECKER EVALUATION RELIABILITY

4.1 INTRODUCTION

In Chapter 1 it was made clear that the aim of evaluations is to determine the quality of a product by measuring specific attributes (King, 1999:13). King & Maegaard (1998:4) point out that all attributes have some value, and that these values can vary between attributes that have numerical representation and those that are merely statements of fact (TEMAA, 1997:8). This study is concerned with determining the reliability of the quantitative metrics that measure the functionality quality characteristics as described in 3.2.1. This chapter aims to answer the following research questions:

a. How can evaluation reliability be modelled and predicted using linear regression procedures?
b. Which variables affect evaluation reliability, and how are those variables identified?
c. How do variables affect evaluation reliability?

As a continuation of the previous chapter, this chapter applies the second phase of the stepwise evaluation reliability methodology, the modelling phase, to three Afrikaans spelling checkers. In this phase, each of the evaluation metrics identified in step 6 of the process is first validated, and then the independent variables that affect the reliability of the metric are identified, using FSRP. The identification of relevant independent variables for each metric is performed by collecting a number of different evaluation corpora that have different inherent qualities, guided by the principals discussed in the previous chapter. The aim of this is to have a large, heterogeneous set of evaluations that represent the different kinds of evaluation corpora that could be available for spelling checker evaluations. The use of heterogeneous evaluation corpora is intended to produce results that can serve as evidence that particular variables have a statistically significant influence on the evaluation results in a number of different domains. Based on the previous step, each metric and the relevant variables are further tested to show how individual variables

affect reliability, in order to gain a better understanding of the relationship between the variable and metric reliability.

The final step in the evaluation reliability methodology is the selection of a linear regression model that includes each of the variables selected by FSRP. The purpose of this model is to be able to accurately predict the reliability of future evaluations where the values of the independent variables are known. Once a valid model has been selected, the model must be validated on additional validation corpus data, and if the model is shown to accurately predict the variability on these new validation sets, it can be used to predict future evaluation reliability.

4.2 MODELLING EVALUATION RELIABILITY: RECALL CORRECT

4.2.1 STEP 7: VALIDATE RECALL CORRECT

Recall correct (R_c – Starlander & Popescu-Belis, 2002:271) is defined as the number of valid words in the text that are accepted by the spelling checker as correct, true positives (*Tp*), in relation to the total number of correct words in the text, that is, the sum of *Tp* and words incorrectly identified as incorrect, false negatives (*Fn*). Recall correct is calculated as follows:

$$R_c = \frac{Tp}{Tp+Fn} \qquad (4.2\text{-}1)$$

This is a measure of the acceptance of correct words in relation to all the correct words in the evaluation, expressed as a proportional fraction between 0 and 1. Most of the metrics that describe proportional values in the rest of this study are expressed as percentile values, which is a way of expressing the proportion of a whole of 100. The higher the R_c score, the higher the spelling checker's coverage of the target language. Visually, this is represented by the central grey area in Figure 4.2-1. The greater the proportion of the language covered by the speller, the higher R_c is. This metric also represents the probability of any correct word being accepted as correct in future, based on the response of the spelling checker on the modelling unit.

In order to evaluate the R_c of the spelling checker, each correct word used as input to the spelling checker is classified as *Tp* or *Fn,* based on the response of the spelling checker. Since only correct words can be classified as *Tp* or *Fn* according to the

definition of these terms for spelling checker evaluations (Van Huyssteen *et al.*, 2004), only correct input words are used to evaluate the reliability of the R_c metric.

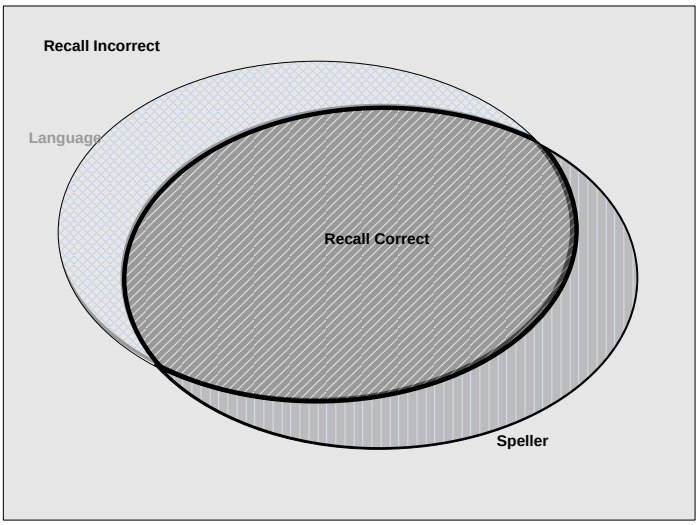

Figure 4.2-1: Visual representation of R_c

4.2.2 STEP 8: SELECT INDEPENDENT VARIABLES FOR R_C

As a first step in determining the reliability of the R_c metric, it is necessary to identify the variables that have a statistical relation to the variability of the metric and can be used to predict the CI of R_c. In statistical relations, the variables that are predictors of another variable are referred to as independent, explanatory or predictor variables, while the related variable, the deviation for R_c in this case, is known as the dependent or response variable. In the previous chapter, the set of evaluation variables that were relevant for spelling checker evaluations were discussed in detail, and from that list, the following set of variables is possibly relevant to the evaluation of R_c:

a. unit size;
b. TTR;
c. heterogeneity; and
d. quality.

The reason why only these variables are relevant to the evaluation of R_c is that this metric is only interested in the set of correct words in the modelling unit and how well these words are recognised by the spelling checker. For this reason, the error type and the incorrect-to-correct word ratios are not relevant to the R_c metric. Although it is possible to build a tool to generate all possible or theoretically correct forms in a language to calculate R_c, it seems illogical to generate data for a scenario

where there are large amounts of good user-generated data available. Generating the correct forms also introduces an additional element of variability to the process. For this reason, the effect of automatically generated data is not included as a possible predictor of variability for R_c.

The first step in the methodology is to determine which of the variables that are inherent in the evaluation of R_c are good predictors of variability when different evaluations are completed. The variable identification can be done by any one of the variable identification procedures described in the previous chapter, mostly depending on the number of variables that are identified for the particular metric.

With the aim of determining which of these variables have a statistically significant relationship with the variability of R_c, each variable is treated as an independent variable in the statistical analysis of evaluation results. With the combination of results and taking each of the variables in each experiment into account, it can be determined whether particular variables have a statistically significant influence on the results of R_c by using FSRP (cf. Chapter 3.2.2).

The modelling sets used for performing the regression procedure were created by randomly selecting documents from the 320,000 word modelling corpus, where each selected modelling unit had between 1,000 and 30,000 words without any control over the independent variables in the corpora. The modelling units selected in this step are also used to do the required modelling for the other evaluation metrics throughout the rest of this chapter. Each of these corpora was then run through the respective spelling checkers to calculate the R_c metric.

For this regression model, the minimum acceptable *F* limit for adding a variable to the model is 3.88, which is determined by $F(0.95; 1, 217)$. At each step in FSRP, the variable with the highest *F* over 3.88 is added to the regression model. In the first step, the TTR variable has the highest *F* statistic, 249.99, and is entered into the model. The values in Table 4 represent the following steps in FSRP, and add the quality and modelling unit size variables in steps 1 and 2 respectively. In the final step, the heterogeneity variable does not have a sufficiently high *F* statistic, and is not added to the model.

Candidate variables	*F* Statistic	Numerator degrees of freedom	Denominator degrees of freedom
Step 1			
Unit size	2.3314	1	216
Heterogeneity	0.2018	1	216
Quality	**481.6580**	1	216
Step 2			
Unit size	**7.4334**	1	215
Heterogeneity	2.0595	1	215
Step 3			
Heterogeneity	2.9091	1	214

Table 4: Steps 1-3 FSRP results for R_c.

The results of FSRP show that the variability in the results is best described by the following variables:

- unit size;
- TTR; and
- quality.

Given the regression results presented here, the following section tests some more specific hypotheses about the relationship between these variables and the R_c metric. Controlled experiments for each of the variables are carried out to calculate the regression model and deviation for evaluations given a particular variable. This provides a more in-depth description of how the different evaluation variables affect the reliability of evaluations.

4.2.2.1 Hypotheses

Based on the results from FSRP, the following hypotheses are tested to gain a deeper understanding of the influence of the variables on the evaluation results of the R_c metric:

a. The larger the modelling unit, the more reliable the reported R_c is.
b. The lower the TTR, the more reliable the reported R_c is.
c. The higher the quality of the technology, the more reliable the reported R_c is.

In addition to the independent variables that have been shown to influence the reliability of the R_c metric, two additional hypotheses are investigated. These hypotheses aim to investigate methodological variables as described in the TEMAA evaluation framework.

a. Word tokens ensure a more reliable R_c metric than using word types.
b. The best way to distinguish between different spelling checkers is to use low-frequency words.

The following sections investigate each of these individual hypotheses; they describe experiments to further verify these initial results and provide a better understanding of how each of the independent variables affects the results, variance and reliability of the R_c metric.

4.2.2.2 Data-attribute experiments

a. The larger the modelling unit, the more reliable the reported R_c is.

It can reasonably be assumed that the larger the modelling unit is, the more reliable the R_c metric should be, as alluded to by El-Emam (2000:21). This follows from the fact that the larger the modelling unit, the more representative of the entire population[9] the sample is. Large evaluation corpora are, however, expensive and time-consuming to develop (i.e. tagging incorrectly spelled words) and the tagging of the modelling unit is prone to human error. For this reason, the following section shows how the size of the modelling unit affects R_c in spelling checker evaluations, while limiting the size of the modelling unit to make tagging and development easier and more cost-effective.

In order to determine the standard deviation (σ) of the results of R_c, Afrikaans modelling units of differing sizes between 1,000 and 30,000 words are used to determine what the σ for each set of experiments is. The corpora are again randomly selected from the larger 320,000-word set without specifically controlling for domain, ratio or any other attributes. Each modelling unit contains only correctly spelled words, which are then used to do the evaluations of R_c. Figure 4.2-2 presents the values of σ for three Afrikaans spelling checkers on modelling sets of 1,000, 5,000,

[9] Note that population in all of these experiments refers to the entire written language, i.e. all the words that are correct in the language. Each evaluation corpus is only a sample of this population and is then used to extrapolate results for the entire population.

10,000, 15,000, 20,000, 25,000, and 30,000 words. The σ indicates to what extent the different samples are spread around the mean. The smaller this σ, the more closely values are clustered, and thus the more reliable a single test metric is.

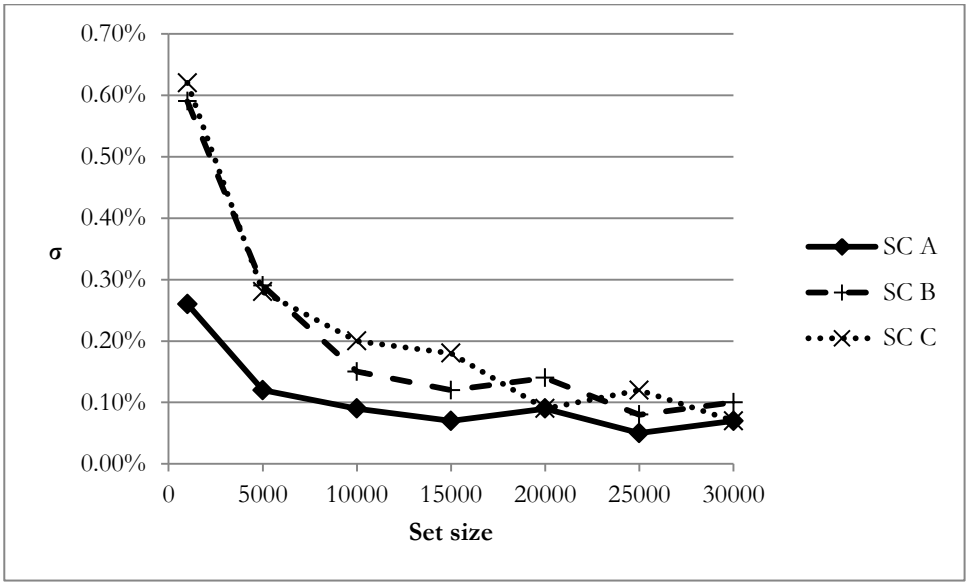

Figure 4.2-2: σ of Afrikaans spelling checkers on texts of differing sizes

The results in Figure 4.2-2 show that σ decreases as the size of the text increases, although this decrease is not linear, but rather follows a logarithmic curve. The nature of the logarithmic curve is to decrease steeply when comparing the results between the smaller samples, and then to tail off as the modelling unit size increases, with the variance between different modelling sets becoming ever smaller. The strong correlation between SC B and SC C is attributable to the fact that the two spelling checkers perform very similarly with regard to R_c. This is an indication that the larger the corpora that are used, the smaller the σ around the mean becomes, but that stability of the metric can only be approximated to a given point, after which slight variance in the σ always occurs.

Text size	**SC A**	**SC B**	**SC C**
1,000	±0.52%	±1.18%	±1.24%
5,000	±0.24%	±0.58%	±0.56%
10,000	±0.18%	±0.30%	±0.40%
15,000	±0.14%	±0.24%	±0.36%
20,000	±0.18%	±0.28%	±0.18%
25,000	±0.10%	±0.16%	±0.24%
30,000	±0.14%	±0.20%	±0.14%

Table 5: 95% CI for Afrikaans spelling checkers on texts of differing sizes

The data in Figure 4.2-2 indicates that the σ actually increases for some of the spelling checkers with corpora that are larger than 15,000 words, as opposed to decreasing further. This indicates that the influence of the modelling unit size becomes negligible with modelling units larger than 15,000 words, and that the amount of variance around the mean always differs, but only to a very small degree.

This is supported by doing an *F* test, which determines how significant the variance between two data sets is. The null hypothesis (H_0) in this case would be that the variation between a smaller modelling set and the next larger set is not significant. For this, the *F* test is used to calculate the significance of the variance between the experiments on 1,000 5,000, 10,000, 15,000, 20,000, and 25,000 and 30,000-word modelling sets. If the variance is not significant, it can be assumed that a smaller sample modelling set would give results that are as reliable and representative of the entire population as that of a next larger modelling set.

The results of the *F* test in Table 6 give the p values (i.e. the error) of the *F* test for each of the spelling checkers. In cases where $p < 0.05$, the variance in the results between two sets of samples is significant, and the H_0 must be rejected, but in cases where $p > 0.05$, the H_0 can be accepted, which means that the variation between two sample sets is not significant. The results of this test show that even smaller modelling units of 10,000 words does not show significant variance compared to the results of the 15,000 word experiments, as all of the p values for text of more than 10,000 words are above 0.05, and the null hypothesis can be accepted. This implies that with modelling unit sizes above 10,000 words, the variance between different samples does not show significant variance, and that a modelling unit of 10,000 words will be adequate to evaluate a spelling checker reliably.

	SC A	**SC B**	**SC C**
Number of words	**p**	**p**	**p**
1,000 & 5,000	0.158624	0.016894	0.009533
5,000 & 10,000	0.658938	0.043997	0.018136
10,000 & 15,000	0.812596	0.363295	0.063524
15,000 & 20,000	0.429711	0.753033	0.108450
20,000 & 25,000	0.803229	0.447425	0.704208
25,000 & 30,000	0.619481	0.772811	0.400820

Table 6: p values of a one-tailed *F* test with different sample sizes for R_c

In cases where only a single spelling checker is available, another method for determining the R_c of a spelling checker with 95% confidence is to determine the 95% CI of the spelling checker (cf. Addendum A). By doing this calculation with a 15,000 word modelling unit, it can reliably be determined with 95% confidence what a spelling checker's R_c value is, and this interval gives one an approximation of the range into which other similar evaluations will fall for future evaluations.

There are two other important details to keep in mind with regard to the results presented in this section. Firstly, the results cannot necessarily be generalised to all languages, as the linguistic complexity of the language possibly has an influence on the number of words necessary to attain an accurate evaluation. Since spelling checkers are concerned with the orthography of a language, specifically recognising correctly spelled words and flagging incorrect words, the morphological structure of a language must be included as one of the variables in the evaluation process. Conjunctively written languages and synthetic ones have a larger number of possible words in the language (Pirkola, 2001:331), and by implication, a larger number of words are necessary to evaluate a spelling checker accurately. This differentiation in morphological complexity is representable as a variable in the TTR, where morphologically more complex languages have a higher TTR, which affects the reliability measure established by the proposed model for spelling checker evaluation.

Secondly, it would seem that the quality of the spelling checker that is evaluated also has an influence on the accuracy of the results. From the results in Figure 4.2-2, spelling checkers with a lower R_c, have a higher level of variance around the mean than a spelling checker with a higher R_c. This is supported by the mean R_c scores of the spelling checkers in Table 7. This difference in variance is not as perceptible with small differences between the R_c results (as between SC B and SC C), but does become evident in instances where there are larger differences (e.g. between SC A and SC B, and SC A and SC C).

	SC A	**SC B**	**SC C**
15,000 words	99.24%	96.04%	96.04%

Table 7: R_c mean of Afrikaans spelling checkers on modelling sets of 15,000 words

The effect of these two variables on the evaluation is the focus of the following experiments with the aim of getting a better sense of the relationship of these variables with R_c results.

b. The lower the TTR, the more reliable the reported R_c is.

When looking at the TTR of the stratified corpora, the TTR follows a regression curve that is related to the text size – that is, as the text size increases, the TTR decreases.

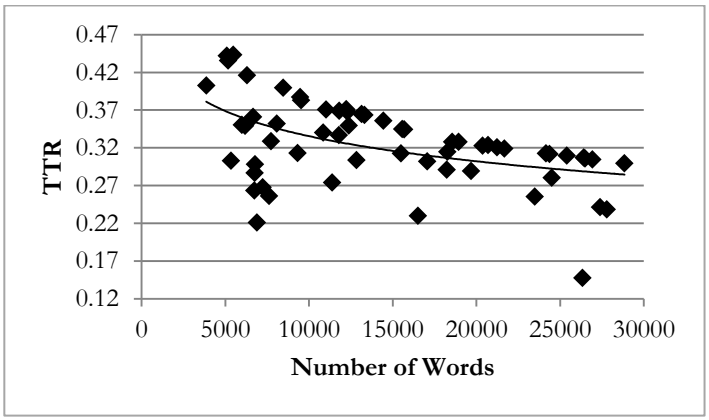

Figure 4.2-3: TTR for Afrikaans stratified corpora

Similarly, TTR has an inverse relationship to the variability of R_c results –; as TTR decreases, the variability between the results decreases. This indicates that the reliability of evaluations can be predicted by the variability in TTR.

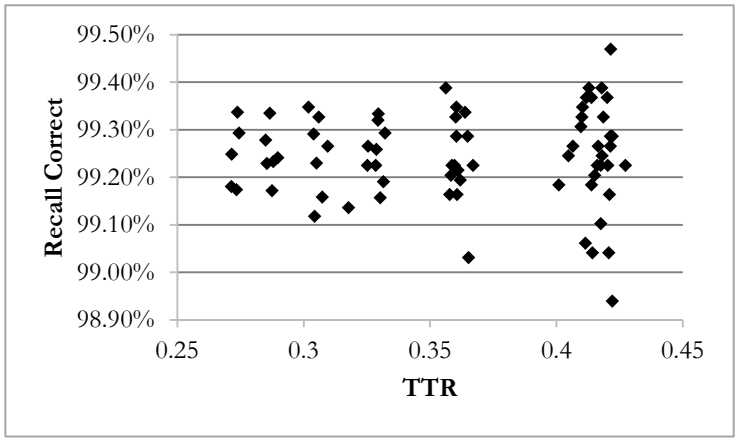

Figure 4.2-4: Relationship between TTR and R_c for Afrikaans corpora

This decrease in variability supports FSRP results that indicate that TTR is a good predictor of variability. The main reason for this is that the smaller the relative variability within the modelling unit, the more consistent the results of the reported metric are. Since TTR expresses the relation between the number of unique words and the total number of words in the modelling unit, the lower this value, the less variability is likely to occur in the reported metric.

This conclusion is supported when considering the σ of R_c in relation to the TTR (Table 8). Across all three spelling checkers, the variability decreases as the TTR of the modelling unit decreases. This provides additional support for FSRP results that indicates that TTR is the variable with the highest predictive quality for determining variation in evaluations.

TTR	**SC A**	**SC B**	**SC C**
0.4	0.12%	0.28%	0.28%
0.35	0.09%	0.15%	0.20%
0.3	0.07%	0.12%	0.12%
0.25	0.06%	0.08%	0.09%

Table 8: R_c σ of Afrikaans spelling checkers on modelling units with varying TTR

c. The higher the quality of the technology, the more reliable the reported R_c is.

One of the predictor variables identified during FSRP is the quality of the technology. This is a somewhat counter-intuitive finding since determining the value of R_c is required to determine the expected reliability of the evaluations. This section shows how the quality affects the reliability of the evaluations that are performed. From a theoretical point of view at least, the quality of technology relates to the error rate of the technology (i.e. how often does the spelling checker flag correct words) and the higher this error rate, the more likely it is that there will be variability in the results.

This assumption is supported by the results in Table 9, where three Afrikaans spelling checkers were evaluated on 15 modelling units of 5,000 words each, randomly selected from the full modelling corpus. The results show that the σ of lower quality tools, that is, those with a lower R_c mean, is more than double that of SC A which has the highest R_c mean.

	R_c mean	σ
SC A	99.26%	0.1179%
SC B	96.08%	0.2826%
SC C	96.08%	0.2753%

Table 9: Mean and σ of three Afrikaans spelling checkers on 15 modelling units of 5,000 words

4.2.2.3 Methodological variable experiments

In the previous chapter it was pointed out that the methods that are used to perform evaluations can also have a direct effect on the reported evaluation results. Since these variables are not quantifiable, it is not possible to include them in regression models that can describe the reliability of the evaluation. However, these variables are still valid sources of variability, and the following experiments show how the selection of a particular methodological implementation can affect both the results and the reliability of evaluations.

a. Using word tokens ensures a more reliable R_c metric than using word types.

As an extension of the experiments that were carried out in 4.2.2.2.a, the same experiment was repeated, but instead of using tokens, a list of types was used. This experiment reduces the number of words in the text, making it easier for the evaluator to review the full set of items in the document, but changes the nature of the evaluation, as it no longer represents the perspective that an end-user would have (cf. 4.2.2.2.b).

Table 10 contains the comparison of σ values when using types as opposed to using word tokens. These results show that the σ of types is larger than that of tokens in all of the experiments. From this it can be deduced that the range of values for types is wider than that of tokens. If one wanted to indicate that 95% of all evaluations fall within a certain range, the range for types would be larger. This means that the results of token evaluations have a smaller σ than those done with types. As a smaller σ can be interpreted to represent the actual R_c value more accurately, the metric is more reliable.

	SC A		SC B		SC C	
Text size[10]	**Tokens**	**Types**	**Tokens**	**Types**	**Tokens**	**Types**
5,000	0.12%	0.31%	0.29%	0.63%	0.28%	0.54%
10,000	0.09%	0.24%	0.15%	0.39%	0.20%	0.42%
15,000	0.07%	0.23%	0.14%	0.42%	0.18%	0.31%
20,000	0.09%	0.16%	0.12%	0.24%	0.09%	0.33%
25,000	0.06%	0.15%	0.10%	0.45%	0.12%	0.27%
30,000	0.07%	0.09%	0.08%	0.18%	0.07%	0.15%

Table 10: σ of three Afrikaans spelling checkers' R_c on tokens and types

As has been mentioned earlier, running text also reflects the experience of the end-user more accurately. As this study aims to simulate end-user experience, it is proposed that tokens (i.e. the full text) should be used to evaluate a spelling checker. Even though words can be added to a user-defined dictionary in *Microsoft® Office*, and would not be flagged more than once, using tokens rather than types is a closer approximation of what a user will experience, as the recognition of higher frequency words will increase the R_c metric to the approximate experience of a user.

b. The best way to distinguish between different spelling checkers is to use low-frequency words.

In the evaluations done by TEMAA (Paggio & Underwood, 1997; TEMAA, 1997), structured word frequency lists of correct words (also referred to as "base lists") were used for the evaluation of R_c. These lists consisted of the most frequent word types in the modelling corpus, divided into frequency groups, which could then be used to describe the R_c of a spelling checker at different frequency intervals. This was done in order to show the functionality of the spelling checker on more and less frequent words. For these experiments, the least frequent 18% words of the modelling corpus were excluded from the evaluations. According to TEMAA (1997:38), this would limit the number of misspelled words in the base list, and in addition evaluate the coverage of the most frequent words, which should not be flagged. Paggio & Music (1998), on the other hand, excluded words with a frequency of less than 20 occurrences in their evaluation of the SCARRIE project, as validation of the TEMAA framework.

[10] Text size is used here as the distinguishing attribute since the TTR decreases as the size of the text increases because it is more likely that a particular token appears multiple times in a larger set than in a smaller set, where the most frequent words are likely to appear multiple times.

In all the discussed evaluations, the list of correct input words was divided into three frequency categories:

1. highest-frequency words covering 48% of the modelling corpus;
2. high-frequency words covering 21% of the modelling corpus (cumulatively covering 69% of the modelling corpus);
3. lower-frequency words covering 13% of the modelling corpus (cumulatively covering 82% of the modelling corpus).

The words covering the remaining 18% of the modelling corpus were not included in the evaluations performed by these evaluators.

The R_c results for three Afrikaans spelling checkers are given in Table 11. These results followed the same guidelines used in the TEMAA evaluation, and excluded the least frequent 18% of word types in a modelling corpus of 320,000 words (described in 2.3.4). From these evaluations it is obvious that the lower frequency groups show ever larger distinctions between the R_c results. Although there is some variation in the 48-69% and 69-82% groups, these do not provide conclusive evidence as to which spelling checker performs the best on R_c. The results in these evaluations show that all three spelling checkers are comparable in terms of recognising the most frequent words of the language.

Percentage of modelling corpus covered	**SC A**	**SC B**	**SC C**
1-48%	100.00%	100.00%	100.00%
48-69%	99.76%	99.28%	99.28%
69-82%	99.63%	98.88%	98.88%
83-100%	96.31%	79.47%	79.51%

Table 11: R_c of three Afrikaans spelling checkers divided into word frequency coverage

When reviewing the results on the remaining 18% of the modelling corpus (the final row in Table 11), the difference in R_c is obvious on the lower frequency words. SC A performs noticeably better than both the other spelling checkers, recognising 96.31% of the final 18%, while both other spelling checkers recognise less than 80% of these less frequent words. One can therefore conclude that using less frequent words is a discriminatory way of determining the difference in the R_c functionality between different spelling checkers. However, it is also important to evaluate the most frequent words, as these should always be recognised.

Since one of the fundamental aspects of this study is to simulate the user's experience of the functionality of a spelling checker, it seems inconsistent to use word-frequency lists for the evaluations rather than using the text as a whole, which would be the way a user would use the spelling checker. The results presented in Table 12 are the weighted mean performance of the three spelling checkers on the same data used in Table 11. These results report higher R_c values than those of the evaluations on the frequency lists, as the higher-frequency words have a bigger effect on the results. The method of using running text can be assumed to describe the experience of the end-user more accurately, as the user has to ignore words flagged by the spelling checker approximately four times out of every 100 words (or once every 25 words) with SC B and SC C. This describes the experience much better than indicating that the user needs to ignore the spelling checker on every 10th *different* word.

	SC A	**SC B**	**SC C**
R_c (1-82% of modelling corpus).	99.87%	99.64%	99.64%
R_c (entire modelling corpus)	99.25%	96.04%	96.05%

Table 12: R_c figures of Afrikaans spelling checkers on a 320,000 word modelling corpus

4.2.3 STEP 9: SELECT AND VERIFY RELIABILITY MODEL FOR R_C

The experiments performed in the previous section provided additional evidence to support the identification of predictor variables for the R_c metric. Since this additional evidence supports the initial findings of FSRP, the output model from FSRP can be used to predict the CI that would be expected from future evaluations in the same paradigm, provided that the values of the three identified predictor variables are known. The required information for the model is part of the final step output from FSRP presented in Table 13.

Constant = 0.026; Std. Error of estimate =0.003

Variable	**β**	**Std. Error**
TTR	0.016	0.002
Quality	-0.031	0.001
Unit size	0.001	0.000

Table 13: Final step output from FSRP for R_c

In this study a simple linear regression model is used as the predictor model where the CI can be predicted for R_c using the following function:

$$Rc_i = 0.026 + (0.016 * x_{i1}) + (-0.031 * x_{i2}) + (-0.001 * x_{i3}) + 0.003 \quad (4.2\text{-}2)$$

where:

- x_{ij} are the values of the independent variables of the *i*th trial
  - x_{i1} is the TTR;
  - x_{i2} is the quality; and
  - x_{i3} is the unit size.

This model can now be used to predict the expected variability for a particular validation unit based on the values of the three variables in the particular evaluation corpus. As an example, a validation unit with:

- TTR = 0.3687
- quality = 0.9864
- unit size = 0.3579[11]

completes function 4.2-1 as follows:

$$Rc_i = 0.026 + (0.016 * 0.3687) + (-0.031 * 0.9864) + (-0.001 * 0.3579) + 0.003 \quad (4.2\text{-}3)$$

Completing the calculation gives a predicted value of ±0.004, or 0.4% for 95% of evaluations, which is smaller than ±1%, and the evaluation results on this unit are reliable.

In addition to the performed experiments, it is also necessary to verify the model by performing additional evaluations on data not used in the modelling stage, and comparing the results of these evaluations with the predicted CI from the model. For this purpose, four validation units were randomly selected from the validation corpus and run through each of the three Afrikaans spelling checkers to compare the predicted CI with the actual error of the particular evaluation.

[11] Note that unit size is normalised to a value between 0 and 1 for modelling purposes in order to have all of the variables on the same scale, and is widely accepted in machine learning circles (Mitchell, 1997; Marsland, 2009). This ensures that none of the model values are too small. For the purposes of these evaluations, the unit of normalisation is done by dividing the unit size value by 30,000, the largest unit size used in the modelling.

Speller Name	Unit size	TTR	Quality	Predicted 95% CI	Actual error
A	0.164500	0.413000	99.39%	±0.46%	-0.14%
A	0.374500	0.359700	99.22%	±0.36%	0.02%
A	0.495733	0.335600	99.27%	±0.31%	-0.03%
A	0.765200	0.275560	99.23%	±0.19%	0.02%
B	0.164500	0.413000	95.73%	±0.58%	0.32%
B	0.374500	0.359700	96.07%	±0.46%	-0.02%
B	0.495733	0.325600	96.03%	±0.39%	0.01%
B	0.765200	0.285560	96.12%	±0.30%	-0.07%
C	0.164500	0.413000	96.02%	±0.57%	0.05%
C	0.374500	0.359700	95.86%	±0.47%	0.20%
C	0.495733	0.325600	96.03%	±0.39%	0.02%
C	0.765200	0.285560	96.11%	±0.30%	-0.04%

Table 14: Predicted CI for R_c of three Afrikaans spelling checkers and the actual error on four validation units

The results in Table 14 show how the suggested model performs when predicting the CI for four different corpora on three different spelling checkers. In all twelve of these validation cases, the actual error falls well within the range of predicted variance, which is proof that the proposed model for predicting spelling checker variation is valid. This is further supported by the MSPR and MSE values which are 0.0000014 and 0.0000029 respectively. Since these two values are very close, the model is a good predictor of future behaviour according to the validation requirement set by Neter *et al.* (1996) and Field (2005).

4.3 MODELLING EVALUATION RELIABILITY: RECALL INCORRECT

4.3.1 STEP 7: VALIDATE RECALL INCORRECT

The second recall measure used in spelling checker evaluations is *recall incorrect* (R_i) (Starlander & Popescu-Belis, 2002:271), also referred to as *error coverage* by Paggio & Music (1998), Paggio & Underwood (1997), and TEMAA (1997). R_i can be defined as the number of incorrect words in the text that are flagged by the spelling checker, true negatives (*Tn*), in relation to the total number of incorrect words in the text, that is, the sum of all true negatives and false positives (*Fp*):

$$R_i = \frac{Tn}{Tn+Fp} \qquad (4.3\text{-}1)$$

R_i measures how well a spelling checker flags invalid words that do not belong to the target language, in relation to all the incorrect words in the evaluation corpus. The result of this metric is a proportional value between 0 and 1 that can be expressed as a percentile value, where the higher the value, the more incorrect words are flagged as incorrect. This can be interpreted as an estimation of the future probability that an incorrect word will be flagged as incorrect by the spelling checker, based on the results of the evaluation corpus.

Visually this can be represented as the area outside both the language and speller circles in Figure 4.3-1. The smaller the area covered by the speller outside of the language, the higher R_i is. In order to evaluate the R_i of the spelling checker, each word used as input to the spelling checker is classified as *Tn* or *Fp,* based on the response of the spelling checker. Since only incorrect words can be classified as *Tn* or *Fp* according to the definition of these terms for spelling checker evaluations (Starlander & Popescu-Belis, 2002), only incorrect input words are used to evaluate R_i, and consequently, only the variables related to incorrect words are considered as explanatory variables for R_i.

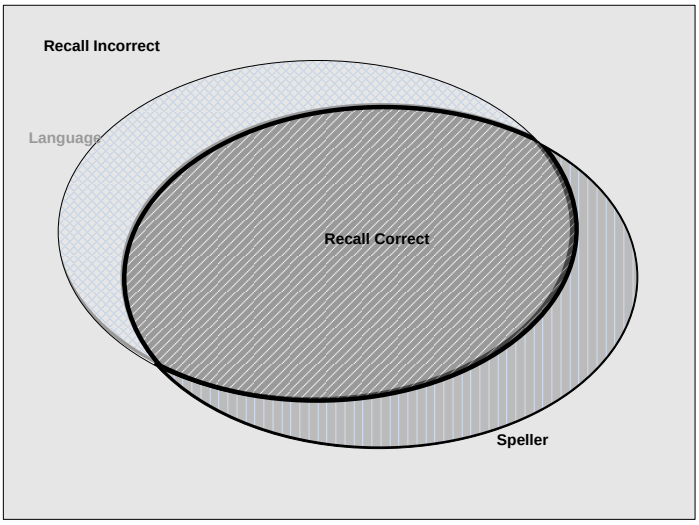

Figure 4.3-1: Visual representation of R_i

4.3.2 STEP 8: SELECT INDEPENDENT VARIABLES FOR R_I

As was mentioned earlier, reliability is the assurance that a particular evaluation reports results that are an accurate representation of the quality of a tool, but also

that the variability between similar evaluations is minimised. Determining the reliability of a particular evaluation for R_i is done in a similar way to the procedure followed for determining R_c reliability. Again, this entails creating a set of modelling units to identify relevant predictor variables that influence the R_i metric. After the identification of the relevant variables, experiments to further investigate how the predictor variables influence reliability are carried out, to achieve a deeper understanding of the variables and their effect on R_i. After these explanatory experiments, I provide a model that can predict the CI of the R_i metric for future evaluations.

The variables that are of interest for R_i evaluations are:

a. unit size;
b. error TTR[12];
c. heterogeneity; and
d. quality.

As with R_c, the first set of experiments for R_i is to determine which of the identified variables are good predictor variables for R_i. For this, FSRP is used to select the variables that are good predictors of the deviation of R_i. For this regression model, the minimum acceptable *F* limit for adding a variable to the model is 3.88, which is determined by $F(0.95; 1, 217)$. At each step in FSRP, the variable with the highest *F* over 3.88 is added to the regression model. In the first step, the quality variable has the highest *F* statistic, 42.801, and is entered into the model. The values in Table 18 represent the following steps in FSRP, and add the unit size variable in step 1. In the final step, neither the heterogeneity nor the error TTR variables have a sufficiently high *F* statistic to be added to the model.

[12] The error TTR is the type-to-token ratio of incorrect words in the set and is calculated in the same manner as TTR, except that it only takes incorrect words into account.

Candidate variables	*F* statistic	Numerator degrees of freedom	Denominator degrees of freedom
Step 1			
Heterogeneity	2.7639	1	216
Unit size	**6.7756**	1	216
Error TTR	5.2373	1	216
Step 2			
Heterogeneity	3.6033	1	215
Error TTR	0.6801	1	215

Table 15: Steps 1-2 FSRP results for R_i

Based on the results from FSRP the following variables are the best predictors of deviation for the R_i metric and will be included in the predictive model in Step 9:

- quality; and
- unit size.

The following section does additional experiments into the way the different variables affect the results reported by the R_i metric.

4.3.2.1 Hypotheses

Based on the previous discussion, the following hypotheses are tested to determine the influence of variables on evaluation results for the R_i metric:

a. The reliability of R_i increases with the number of errors.
b. The reliability of R_i is not dependent on the type of errors that are used in the evaluation.
c. The reliability of R_i is dependent on the origin of errors.

4.3.2.2 Experiments

a. R_i is more reliable with a larger number of incorrect words

This experiment shows that the R_i metric is influenced by the number of incorrect words used in the evaluation, and that using a small number of incorrect words does not yield reliable results. This hypothesis is made on the basis that the influence of a single or two *Fp* instances has a greater effect on the result than they would have if

they occurred in a larger set. For instance, if a single incorrect word is not flagged, and only 10 incorrect words are used in the test, R_i would be 90%. However, if 100 incorrect words were used, R_i would be 99% if the single instance was the only word not flagged by the spelling checker.

In order to evaluate the premise that the number of incorrect words used in the evaluation of R_i has an influence on the metric, a number of experiments were done using different numbers of incorrect words in each evaluation, ranging from 10 errors to 100 errors. The reason for selecting a minimum of 10 errors and a maximum of 100 errors is that the percentage of errors in a typical text is around 1% and 10 errors is 1% of a 1,000 word corpus, while 100 errors represents 1% of 10,000 words. Experiments on sets larger than 100 were not performed since no statistically significant difference is shown for experiments with 90 and 100 words and thus increasing the size of the data would not yield statistically significant differences.

The results reported in Table 16 indicate that there is a significant difference in the variance between samples where 10 and 20 errors are used ($p < 0.05$), but that this variance becomes insignificant when comparing 70 errors to 80 errors (p for all spelling checkers is > 0.05). This would indicate that in order for R_i to be accurately determined, at least 70 errors need to be used in a sample evaluation.

Number of errors	**SC A**	**SC B**	**SC C**
	p	**p**	**p**
10 & 20	0.00	0.00	0.00
20 & 30	0.00	0.01	0.00
30 & 40	0.00	0.16	0.00
40 & 50	0.02	0.22	0.00
50 & 60	0.03	0.25	0.01
60 & 70	0.06	0.33	0.02
70 & 80	0.34	0.47	0.09
80 & 90	0.37	0.52	0.15
90 & 100	0.88	0.57	0.75

Table 16: *F* tests for variance in R_i with different numbers of errors in text

Because minimum error percentages of more than 0.8% are reported in studies on error frequency in user-generated data (Kukich, 1992; Pollock & Zamora, 1984; Verberne, 2002), most corpora should have enough user errors to accurately measure R_i when using corpora that are large enough to also reliably evaluate R_c (i.e. at least

120 errors in 15,000 words). In cases where fewer than 80 errors occur, the results of R_i are not a reliable reflection of the spelling checker's functionality.

b. The reliability of R_i is dependent on the origin of errors

In order to evaluate how well a spelling checker can recognise incorrectly spelled words as incorrectly spelled, one needs a set of incorrectly spelled words. This set of incorrectly spelled words can be attained in one of two ways – either by automatically introducing errors to a list of correctly spelled words, or by collecting a list of incorrect words from existing electronic texts.

In the evaluations done by Paggio & Underwood (1997) and TEMAA (1997), the word frequency base lists of correct words are used to construct a list of incorrectly spelled words. The list of correct words was subjected to an error generation module (Errgen - TEMAA, 1997:77) that introduced errors to words by applying language-specific rewrite rules that simulated typical spelling errors in Danish.

Paggio & Music (1998), however, opted to use the spelling and typographical errors found in their evaluation corpus. Their published evaluation results were not conclusive, as only 99 incorrect words were found in their evaluation corpus, in which the SCARRIE system flagged only 20.

In order to make a decision on which type of error words to use (i.e. generated error words or user-generated error words), similar evaluations for Afrikaans were performed. The errors found during proofreading in the modelling corpus of 320,000 words were used as user-generated errors, and totalled 3,203 incorrect words, or 1.01% of the evaluation corpus. As no error-generation module exists for Afrikaans, a module based on the principles used by TEMAA was developed to simulate the types of errors Afrikaans writers make.

As a premise for the error-generation module, first all the incorrectly spelled words from the evaluation corpus were analysed to determine classes of errors, which could indicate some of the typical errors made by Afrikaans authors. In addition to the words from the evaluation corpus, a study by Butler & Van Huyssteen (2012) identified some common errors in Afrikaans, and other common errors from Afrikaans standard sources were also used (Carstens, 2001; Suid-Afrikaanse Akademie vir Wetenskap en Kuns, 2002). These sources were used to construct a

module that would simulate the types of spelling errors – cognitive errors according to Kukich (1992) – typically made by Afrikaans authors. These error types were grouped as follows, based on an analysis of the incorrect words found in the evaluation corpus and the discussed resources:

- Phonetic errors – Spelling errors caused by incorrectly replacing one grapheme with another, based on the phonemic representation of the graphemic form, for example:
  - /y/ written as /ei/ and vice versa (*verblyf* → **verbleif*).
  - /d/ written as /t/ at the end of words where the second last grapheme is an /m/, /n/ or /r/ (*hond* → **hont*).
  - /v/ and /f/ interchanged (*vars* →**fars*).
- Morphological errors
  - No singling of vowels in plural constructions (*paar* → *pare*, realised as **paare*).
  - No doubling of consonants in plural construction (*kar* → *karre*, realised as **kare*).
- Incorrect use of infix -s- in compounds in one of four ways:
  - Not used where it should be used (*stadsraad* → **stadraad*);
  - Used where it should not be used (*kerkklere* →**kerksklere*);
  - Inserted additionally where the second part of the compound already has an /s/ as its beginning grapheme (*eenheidstatus* → **eenheidsstatus*); and
  - Single word is changed to two words after the infix, where the first part is not a valid word (*werksklere* → **werks klere*).
- Diacritic errors
  - Incorrect use of diacritics and writing signs either by inserting where it is not necessary (e.g. *skêr* → **sker*) or excluded where it is required (*kontinuïteit* → **kontinuiteit*).

These rewrite rules were applied to the same word-frequency list used in the R_c evaluations of 4.2.2.2 b and 4.2.2.3. If any of the identified phenomena occurred in a

given input word, the appropriate change in the word was made according to the rewrite rule. The only exception to this was in cases where the phenomenon occurred at the first character of the word, as Mitton (1996), Pollock & Zamora (1984), and Yannakoudakis & Fawthrop (1983) found that very few errors occur with the first letter of a word. In cases where none of the phenomena occurred, one of the four typographical error types identified by Damerau (1964), Kukich (1992), and Pollock & Zamora (1984) was applied. The frequency of each type of typographical error was regulated to reflect the frequencies published by Peterson (1986:634) in his study of typing errors, and are as follows:

- deletion 34.4%;
- substitution 26.9%;
- insertion 20.3%; and
- transposition 13.1%.

Although Damerau (1964) indicates that 80% of errors are single letter errors, Pollock & Zamora (1984) indicate figures of 90%-95%, and in Peterson's study (1986), 94.7% of the errors were single-letter errors. As multiple errors in one word are infrequent, they were not included in the error generation module (also see Addendum B).

In the experiment the list of generated errors and user errors extracted from the evaluation corpora previously used, were used as input for the spelling checker to determine the R_i of the three spelling checkers on the two different types of errors. Results of the two experiments are given in Table 17.

	SC A	**SC B**	**SC C**
Generated errors	99.68%	99.90%	99.69%
User errors (types)	92.68%	89.46%	84.15%

Table 17: R_i of three Afrikaans spelling checkers on automatically generated and user errors

From these results it can be seen that some disparity exists in the results between the user errors and the generated errors. The number of words for the two evaluations is similar – 1,432 user-generated and 1,415 generated errors – and this variable would therefore not have an adverse effect on the results or the reliability of the results. This disparity again highlights the fact that even though results on similar sets may show a high level of reliability, the influence of methodological variables can have a

noticeable effect on the reported results. The results show a big difference between using automatically generated and user-generated errors, and depending on which is reported, the results could easily be misleading.

4.3.3 STEP 9: SELECT AND VERIFY RELIABILITY MODEL FOR R_I

Based on FSRP performed in 4.3.2, Table 18 shows the final step results of FSRP that is used to create the regression model for predicting the expected deviation for R_i represented in 4.3-2.

Constant = 0.102; Std. Error of estimate = 0.007

Variable	β	Std. Error
Unit size	-0.036	0.002
Quality	-0.076	0.004

Table 18: Final step output from FSRP for R_i

$$Ri_i = 0.102 + (-0.036 * x_{i2}) + (-0.076 * x_{i1}) + 0.007 \quad (4.3\text{-}2)$$

where:

- x_{ij} are the values of the independent variables of the *i*th trial
  - x_{i1} is the unit size; and
  - x_{i2} is the quality.

As before, the predictive validity of the model is validated by performing four evaluations with each of the three Afrikaans spelling checkers in order to determine firstly whether the particular evaluation would provide reliable results, and secondly how well the predictor model correlates with the actual results of the evaluation.

Speller Name	Unit size	Quality	Predicted 95% CI	Actual error
A	0.168100	91.00%	±3.38%	2.83%
A	0.376300	93.00%	±2.48%	1.30%
A	0.504433	95.33%	±1.84%	-1.53%
A	0.761600	94.60%	±0.97%	-0.77%
B	0.168100	74.00%	±4.67%	-4.27%
B	0.376300	68.50%	±4.34%	2.27%
B	0.504433	71.00%	±3.69%	-1.47%
B	0.761600	68.80%	±2.93%	0.77%
C	0.168100	72.00%	±4.82%	**-5.57%**
C	0.376300	64.50%	±4.64%	2.73%
C	0.504433	65.00%	±4.14%	1.33%
C	0.761600	66.20%	±3.13%	-0.03%

Table 19: Predicted CI for R_i of three Afrikaans spelling checkers and the actual error on four validation units

From the results in Table 19 it can be seen that the actual error of most of the evaluations are well within the range predicted by the model showing that the model is reliable. This is further supported by the MSPR and MSE values that are 0.0006294 and 0.0007539 respectively. Since these two values are very close, the model is a good predictor of future behaviour according to the validation requirement set by Neter *et al.* (1996) and Field (2005). The fact that all of the predicted values are relatively large (predicting a CI of between ±0.97% and ±4.82%) is a strong indicator that the results generated for R_i for Afrikaans spelling checkers very rarely is an accurate description of the R_i functionality of the spelling checkers.

4.4 MODELLING EVALUATION RELIABILITY: PRECISION INCORRECT

Precision pertains to the flagging accuracy of a spelling checker, the accuracy of a spelling checker in assigning non-flags (i.e. to recognise only correct words as correct), and producing good flags (i.e. to flag only incorrect words as incorrect) (Paggio & Underwood, 1997:15). Starlander & Popescu-Belis (2002:272) call these measures precision on incorrect words (P_i), and precision on correct words (P_c), while TEMAA (1997:13) only uses the former, and refers to it as precision. The more refined view of Starlander & Popescu-Belis (2002) is used in the subsequent evaluations and discussion as it encompasses TEMAA's precision measure, and also

considers the precision of accepting words as correctly spelled. The two metrics are referred to as precision incorrect (P_i) and precision correct (P_c) respectively.

4.4.1 STEP 7: VALIDATE PRECISION INCORRECT

The number of correct flags (i.e. true negatives) in relation to the total number of flags assigned by the spelling checker (i.e. true negatives plus false negatives) provide the measure of a spelling checker's precision incorrect (P_i). This measure reflects the accuracy with which a spelling checker recognises incorrectly spelled words. P_i is calculated as follows:

$$P_i = \frac{Tn}{Tn+Fn} \qquad (4.4\text{-}1)$$

The scores calculated by P_i are a fraction between 0 and 1 (and can also be expressed as a percentage), where 1 is the ideal, as the end-user expects a spelling checker to flag all incorrect words, and only incorrect words. This results in a spelling checker that is 100% accurate in the task it was set.

The P_i measure is of interest to the evaluator as it indirectly gives an indication of how often a user has to ignore flags assigned by the spelling checker. The more correctly spelled words are flagged by the spelling checker, false negatives (*Fn*), the lower P_i is, and the more times it will be necessary for a user to ignore the spelling checker. In terms of the user, having to ignore the spelling checker is a disturbance, and decreases the user's experience of the spelling checker's functionality.

If a spelling checker is primarily concerned with recognising spelling errors, it would be expected that the confidence that the user has in a spelling checker recognising spelling errors should be the most important spelling checker attribute. From the fact that TEMAA does not measure P_c and that all evaluations in the literature are more concerned with P_i rather than P_c indicates that there has been more emphasis on not bothering the user with false flags rather than making sure all errors are recognised.

4.4.2 STEP 8: SELECT INDEPENDENT VARIABLES FOR P_I

The precision of spelling checkers, both P_i and P_c, is directly dependent on both R_i and R_c, as precision draws the correlation between these two metrics. Because these

are correlation metrics, the same variables that are relevant to R_i and R_c potentially have an influence on P_i and P_c. However, if these metrics are reliable, it is likely that the variables identified there do not have any additional effect on the reliability of P_i and P_c. To verify this assumption, the variables identified for both R_c and R_i are used in a FSRP to ensure that no other variables are of consequence to the calculation of P_i and P_c. In addition to the aforementioned variables, the following variables potentially have an influence on the accuracy and reliability of the precision metrics:

a. unit size;
b. TTR;
c. error TTR;
d. heterogeneity;
e. target density; and
f. quality.

As before, the set of predictor variables identified for determining P_i is processed by FSRP to determine the set of good independent variables for predicting the CI of P_i. For this regression model, the minimum acceptable *F* limit for adding a variable to the model is 3.88, which is determined by $F(0.95; 1, 217)$. At each step in FSRP, the variable with the highest *F* over 3.88 is added to the regression model. In the first step, the TTR variable has the highest *F* statistic, 275.553, and is entered into the model. The values in Table 20 represent the following steps in FSRP, and add the quality and unit size variables in steps 1 and 2 respectively. In the final step, none of the remaining variables have a sufficiently high *F* statistic to be added to the model.

Candidate variables	*F* statistic	Numerator degrees of freedom	Denominator degrees of freedom
Step 1			
Unit size	17.0549	1	216
Error TTR	0.1794	1	216
Heterogeneity	0.0922	1	216
Target density	0.3579	1	216
Quality	**746.4844**	1	216
Step 2			
Unit size	**107.1254**	1	215
Error TTR	0.0253	1	215
Heterogeneity	0.0016	1	215
Target density	0.1429	1	215
Step 3			
Error TTR	0.1749	1	214
Heterogeneity	0.0764	1	214
Target density	2.104	1	214

Table 20: Steps 1-3 FSRP results for P_i

Based on the results from FSRP the following variables are the best predictors of the CI for the P_i metric and will be included in the predictive model in Step 9:

- unit size;
- TTR; and
- quality.

It is interesting that none of the error related variables are good predictors of variability. This can be attributed to the fact that the density of errors in the data is low enough for the variables associated with the incorrect words not to have enough of a predictive quality to accurately predict the variability that can be expected for P_i. This conclusion is tested in more detail in the experiments performed in the rest of this section.

4.4.2.1 Hypotheses

The variables selected by FSRP show that only those variables relevant to the recall metrics are valid predictor variables for the evaluation reliability of P_i. However,

looking at the P_i results in Table 21, it is obvious that P_i can be grossly misleading depending on the percentage of errors (target density) of the evaluations. With this in mind, the following hypotheses are investigated to determine the relationship between target density and the reported results:

a. The higher the percentage of errors in the text, the higher P_i will be.
b. P_i can be stabilised by normalising the percentage of errors in the text.

	Precision
Text 1	18.60%
Text 2	46.91%
Text 3	63.64%
Text 4	28.89%
Text 5	43.64%
Text 6	77.68%
Text 7	45.83%

Table 21: P_i of SCB on a selection of different texts

4.4.2.2 Experiments

a. The higher the percentage of errors in the text, the higher P_i will be.

In the TEMAA evaluation, approximately 6,700 correct words and 4,500 incorrect words were used in evaluating the P_i metric of spelling checkers. If these words were used to construct a text, the errors would make up more than 40% of the words of the text (6,700 + 4,500 = 11,200 words). In a similar experiment for Afrikaans (Table 22), the results of three Afrikaans spelling checkers on different modelling sets show that P_i reports scores of above 98.90% for all three spelling checkers. However, on a user-generated text, the percentage of errors is not nearly as high as in this modelling set, and from the mean of P_i on user-generated texts with varying error percentages, P_i scores are as much as 84 percentage points lower. This indicates that there is a relationship between the percentage of errors (target density) in a text and the value of P_i.

		SC A	SC B	SC C
Experiment A	**Text with 40% errors**	99.64%	98.92%	98.92%
Experiment B	**Texts with 1% errors**	56.99%	15.54%	14.92%
	Texts with 2% errors	72.31%	26.69%	25.73%
	Texts with 3% errors	80.51%	36.39%	34.99%
	Texts with 4% errors	84.07%	42.60%	41.25%
	Texts with 5% errors	87.05%	49.05%	48.06%
	Texts with 6% errors	89.33%	53.33%	55.72%
	Texts with 7% errors	90.90%	57.92%	59.78%
	Texts with 8% errors	91.93%	61.15%	62.79%
	Texts with 9% errors	92.80%	63.63%	65.83%
	Texts with 10% errors	93.12%	67.44%	68.82%

Table 22: P_i on a text with 40 % errors and means of user-generated texts of different sizes

In order to test the hypothesis that the percentage of errors has an effect on P_i, a modelling unit of 1,000 words was seeded with different percentages of errors. The errors from user-generated texts were collected and placed in sets of thirty 1,000 word texts with a given percentage of errors in each set. The increase in errors is compensated for by a decrease in the number of correct words in order to keep the text size the same. In this way, 300 texts of 1,000 words each were evaluated with error percentage of between 1% and 10%. In addition to these 300 texts a single text with 40% errors was created to duplicate the experiment that TEMAA reported in their evaluations.

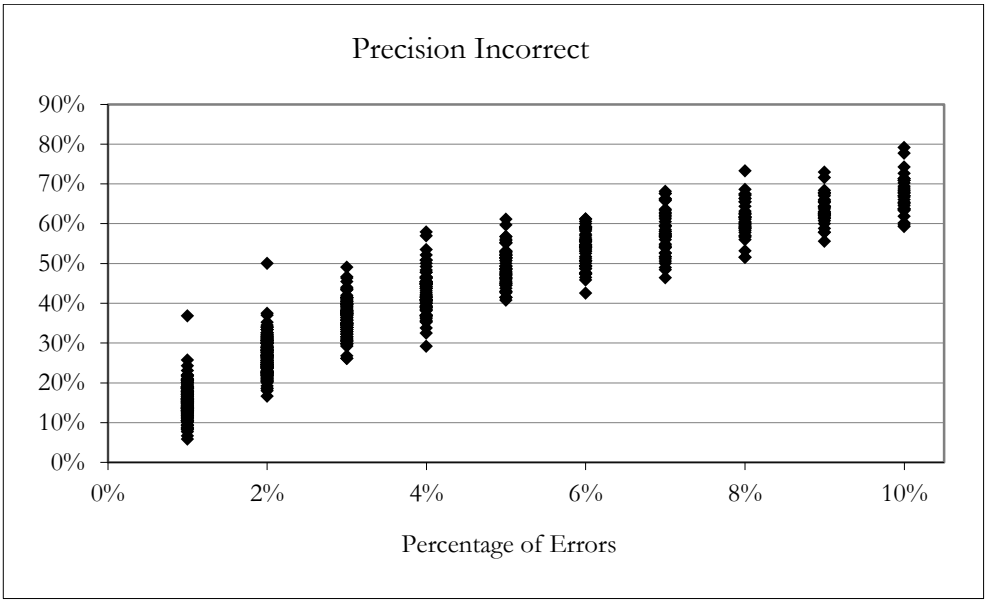

Figure 4.4-1: P_i of SC B on 1,000 word texts with error percentages ranging from 1% to 10%

The results in Figure 4.4-1 show that the percentage of errors in the text has an effect on the overall result of the P_i measure. This shows that P_i increases as the percentage of errors increases in the modelling unit. However, these results also show that the metric does not stabilise, that is to say the σ does not decrease as the percentage of errors increases. This means that the target density does not improve the reliability of the metric, which supports the result of FSRP, but does indicate that for some metrics the actual result can be manipulated by increasing the target density. On those grounds, precision measures should report their target density when reporting the metric as there is a direct correlation between the target density and the value reported by the metric.

This is further supported by the p values of an *F* test between the means of each set of experiments, as shown in Table 23, where all the values are greater than 0.05. As has been indicated previously, if $p > 0.05$, there is no significant variance between the two sets of samples. This means that in order to compare two spelling checkers, the percentage of errors in the text should be the same. The problem stemming from these results, given a user-generated approach, is that there is no control over the percentage errors in the text, and for this reason P_i cannot be used as a reliable comparative evaluation metric. Van Huyssteen *et al.* (2004) offer a possible solution for this problem, which is explored in the following experiment.

Error percentage	p
1% & 2%	0.128575783
2% & 3%	0.760538055
3% & 4%	0.958004875
4% & 5%	0.918282468
5% & 6%	0.885342164
6% & 7%	0.732768472
7% & 8%	0.645921361
8% & 9%	0.726369273
9% & 10%	0.854189247

Table 23: p values of *F* test for P_i of different percentage of errors for SC B

b. P_i can be stabilised by normalising the percentage of errors in the text.

In order for P_i to be considered as a reliable comparative evaluation metric, Van Huyssteen *et al.* (2004) propose normalising the percentage of errors to a predetermined value, and then using the normalised percentage to standardise the resultant P_i of an evaluation. This is calculated by using the following function:

$$Normalised\ P_i = \frac{Tn*\frac{\beta}{\alpha}}{\left(Tn*\frac{\beta}{\alpha}\right)+Fn} \qquad (4.4\text{-}2)$$

where:

- β represents the normalisation percentage;
- α is the original percentage of errors in the text;
- *Tn* is the sum of true negatives in the evaluation; and
- *Fn* is the sum of false negatives in the evaluation.

In order to evaluate the effects of normalisation, 40 experiments[13] are performed with 1% incorrect words and 40 experiments with 8% incorrect words in the text, then normalising the percentage of errors to 5%[14]. The specific error percentage that is used here is not important, as the results remain comparable irrespective of what the normalised percentage of errors is. This is because the normalisation is just rescaling the values to a shared base, as can be seen in Figure 4.4-3 where the same results used in Figure 4.4-2 are normalised to 1% errors.

The results of the normalised 1% and 8% experiments are compared to the results of the original experiments with 5% errors in the text. From the results (Figure 4.4-2), it can be seen that although there is variance in the results, the range into which P_i values fall (between 40% and 60%) is similar to the values of the experiments with 5% errors. Where 8% errors in the previous set of results (Figure 4.4-1) had P_i values of between 52% and 71%, and 1% errors only yielded results of between 11% and 22%, the results of the three normalised experiments show similar values.

[13] 40 was used because it is a large enough set of randomly selected cases for the central limit theorem to apply (n >30) unless the population is skewed (Field, 2005)

[14] The normalisation to a particular percentage is not important since the normalisation is just rescaling the results to a shared value. This could also have been 10% or 0.5% and only aims to place the different experiments on comparable grounds

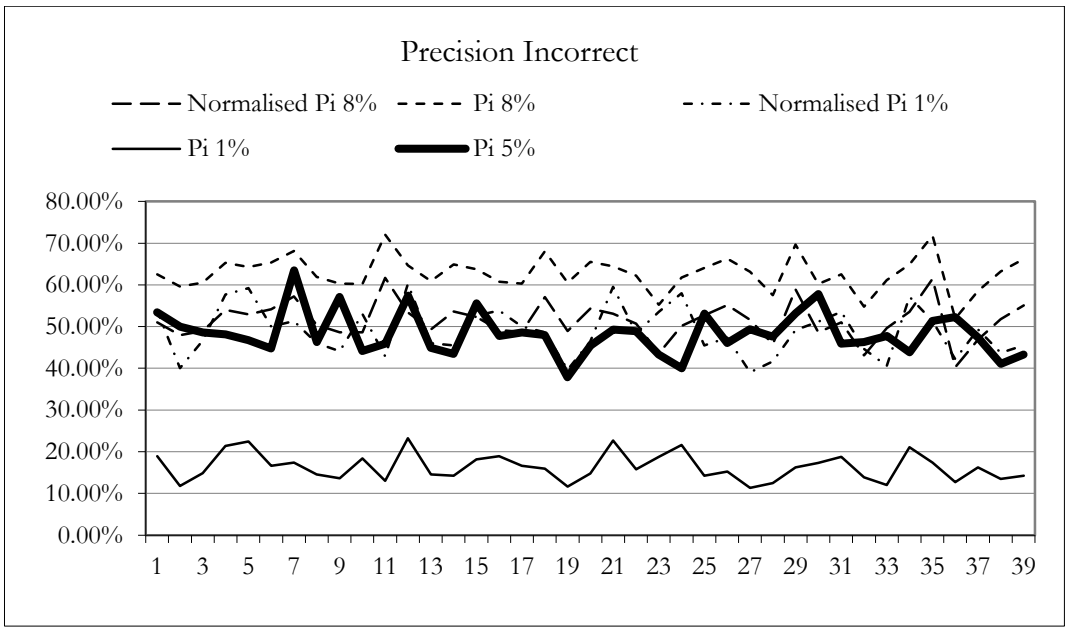

Figure 4.4-2: The effect of error percentage normalisation to 5% on P_i scores of SC B

From these experimental results, the following deductions can be made. Firstly, it is important to decide on a specific value for the percentage of errors in an evaluation text if the results of P_i from different spelling checkers or evaluations need to be compared. This can be done either by artificially seeding a text with the predetermined percentage of incorrect words, preferably with user-generated errors, or by normalising the percentage of errors to a pre-set value after the completion of the evaluation. Both these methods would give similar results.

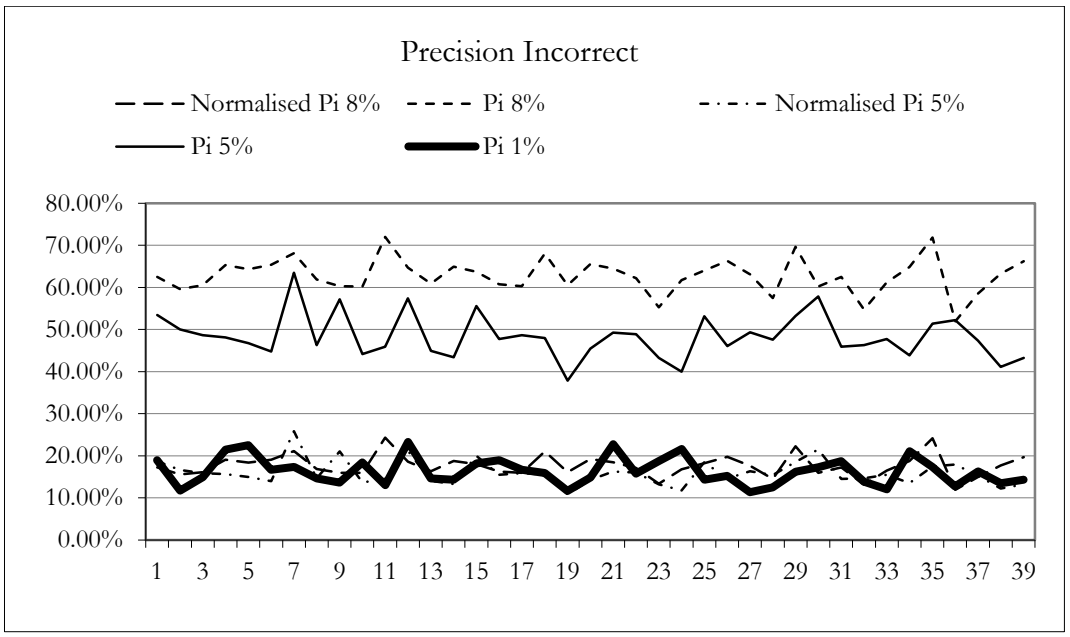

Figure 4.4-3: The effect of error percentage normalisation to 1% on P_i scores for SC B

This correlation between P_i scores with different error percentages is also reflected in an *F* test of the variance between the different error percentages. In Table 23 all the p values in the *F* test are greater than 0.05. This means that the variance between the different datasets is not significant, and it follows that even though the range of results has significant differences, the variation within each of these sets is not significant if related to scores for a higher error percentage. This in turn is evidence that normalising the percentage of errors mathematically gives comparable results when different modelling units are used.

4.4.3 STEP 9: SELECT AND VERIFY RELIABILITY MODEL FOR P_I

Even though P_i can be deceptive in the results that it gives, the results presented in the previous section show that it should still be possible to predict the rate of variability, given the evaluation corpus properties for those variables identified as good independent variables. Based on the previous modelling results, function 4.4-3 describes the regression model to predict variability for P_i of Afrikaans spelling checkers based on the final step output of FSRP for P_i presented in Table 24.

Constant = -0.078; Std. Error of estimate = 0.002

Variable	β	Std. Error
Unit size	0.021	0.002
TTR	0.215	0.010
Quality	0.021	0.001

Table 24: Final step output of FSRP for P_i

$$Pi_i = -0.078 + (0.021 * x_{i1}) + (0.215 * x_{i2}) + (0.021 * x_{i3}) + 0.002 \quad (4.4\text{-}3)$$

where:

- x_{ij} are the values of the independent variables of the *i*th trial
  - x_{i1} is the unit size;
  - x_{i2} is the TTR; and
  - x_{i3} is the quality.

Using four evaluations for the three Afrikaans spelling checkers to verify the models is repeated as before. The results presented in Table 25 show that once again the results of P_i fall well within the range of predicted values even though only two of the variables in the evaluation are used to predict the variability in the evaluations. Noteworthy in these results is the fact that very few of the predicted intervals produce results that are reliable according to the definition in 2.2. This is further indication that P_i may not be a reliable evaluation metric.

Speller Name	Unit size	TTR	Quality	Predicted 95% CI	Actual error
A	0.164500	0.413000	75.21%	±3.20%	-3.07%
A	0.374500	0.359700	70.99%	±2.41%	0.79%
A	0.495733	0.325600	72.59%	±1.97%	-0.97%
A	0.765200	0.285560	71.45%	±1.65%	0.37%
B	0.164500	0.413000	26.15%	±2.17%	0.41%
B	0.374500	0.359700	26.25%	±1.47%	0.52%
B	0.495733	0.325600	26.76%	±1.00%	-0.38%
B	0.765200	0.285560	26.56%	±0.70%	-0.13%
C	0.164500	0.413000	26.97%	±2.19%	-1.26%
C	0.374500	0.359700	24.11%	±1.43%	1.21%
C	0.495733	0.325600	25.03%	±0.97%	0.50%
C	0.765200	0.285560	25.76%	±0.69%	-0.23%

Table 25: Predicted CI for P_i of three Afrikaans spelling checkers and the actual errors on four validation units

The results presented regarding P_i show that it is possible to predict the expected variability for a particular evaluation, given the correct set of evaluation variables. The model's reliability is further supported by the MSPR and MSE values that are 0.0001497 and 0.0001747 respectively. Since these two values are very close, the model is a good predictor of future behaviour according to the validation requirement set by Neter *et al.* (1996) and Field (2005). There may however still be unreported variability in the results that are dependent on factors not captured by the variability methodology because of methodological procedures. In such cases, values for these variables must be reported along with the evaluation results and the expected reliability interval.

4.5 MODELLING EVALUATION RELIABILITY: PRECISION CORRECT

4.5.1 STEP 7: VALIDATE PRECISION CORRECT

Precision correct (P_c) is computed by dividing all true positives (i.e. correct words not flagged by the spelling checker) by the total number of words that are not flagged (i.e. true positives plus false positives). This metric describes the accuracy of the spelling checker in recognising correct words, and is calculated as follows:

$$P_c = \frac{Tp}{Tp+Fp} \qquad (4.5\text{-}1)$$

P_c is a proportional value that can be expressed as a percentage, and is of interest to an evaluator as it directly indicates to the user how confident they can be that all the words that are accepted as correct by the spelling checker are in fact correctly spelled. This is important, because the closer this value is to 100%, the more readily a text that has been spellchecked can be accepted to be 100% correct (i.e. containing no incorrectly spelled words).

4.5.2 STEP 8: SELECT INDEPENDENT VARIABLES FOR P_C

The variables that are relevant for precision correct are the same variables that are applicable to both R_c and R_i, since P_c is a relational measure of the behaviour of the

spelling checker on both correct words and errors. The set of variables that are possible relevant independent variables for the P_c metric are:

a. unit size;
b. TTR;
c. error TTR;
d. heterogeneity; and
e. quality.

As before, the set of predictor variables identified for determining P_c is processed by FSRP to determine the set of good independent variables for predicting the CI of P_c. For this regression model, the minimum acceptable *F* limit for adding a variable to the model is 3.88, which is determined by *F*(0.95; 1, 217). At each step in FSRP, the variable with the highest *F* over 3.88 is added to the regression model. In the first step, the quality variable has the highest *F* statistic, 176.383, and is entered into the model. The values in Table 26 represent the following steps in FSRP, and add the TTR and unit size variables in steps 1 and 2 respectively. In the final step, none of the remaining variables have a sufficiently high *F* statistic to be added to the model.

Candidate variables	***F* statistic**	**Numerator degrees of freedom**	**Denominator degrees of freedom**
Step 1			
Unit size	172.1165	1	216
TTR	**238.5184**	1	216
Error TTR	51.5286	1	216
Heterogeneity	0.204	1	216
Step 2			
Unit size	**8.6684**	1	215
Error TTR	1.4473	1	215
Heterogeneity	1.7427	1	215
Step 3			
Error TTR	1.6555	1	214
Heterogeneity	0.8568	1	214

Table 26: Steps 1-3 FSRP results for P_c

Based on the FSRP results presented here the following variables are good predictors of the expected CI of P_c and will be included in the predictive model in Step 9:

- unit size;
- TTR; and
- quality.

The effect of these variables on the reliability of P_c is investigated in more detail in the following section.

4.5.2.1 Hypotheses

Based on the variables related to P_c the following hypothesis can be stated:

a. P_c is more reliable the larger the size of the evaluation text.

4.5.2.2 Experiments

a. P_c is more reliable the larger the size of the evaluation text.

Based on the results of previous experiments, it seems obvious that the number of words in an evaluation corpus would have an influence on the results of P_c. In the same experiments used for the evaluation of R_c, texts of different sizes between 5,000 and 30,000 words were used to determine how the size of the text influences the P_c metric.

Number of words	SC A	SC B	SC C
5,000	0.04%	0.05%	0.11%
10,000	0.03%	0.05%	0.08%
15,000	0.03%	0.02%	0.04%
20,000	0.04%	0.02%	0.05%
25,000	0.03%	0.03%	0.05%
30,000	0.02%	0.03%	0.06%

Table 27: σ of P_c for three Afrikaans spelling checkers

The results in Table 27, however, show little variation between the results of the different experiments. This is supported by the results of an *F* test (Table 28), where the p values of the variation between the different experiments all indicate that there is no significant variation between the results of the different evaluations. This would indicate that the size of the text does not have a significant influence on the stability

of the P_c measure, and that any modelling unit that is larger than 5,000 words should yield reliable and comparable results.

Number of words	SC A	SC B	SC C
	p	p	p
5000 & 10,000	0.1517	0.3016	0.2463
10,000 & 15,0000	0.2852	0.3330	0.5385
15,000 & 20,000	0.4627	0.6688	0.3125
20,000 & 25,0000	0.1959	0.2048	0.7946
25,000 & 30,000	0.5120	0.9899	0.8394

Table 28: p values for an *F* Test on P_c for three Afrikaans spelling checkers

4.5.3 STEP 9: SELECT AND VERIFY RELIABILITY MODEL FOR P_c

The predictive regression model for predicting the variability in P_c is described by function 4.5-2 based on the final step output from the FSRP for P_c performed in 4.5.2.

Constant = 0.075; Std. Error of estimate = 0.000

Variable	β	Std. Error
Unit size	0.0014	0.002
TTR	0.005	0.010
Quality	-0.077	0.001

Table 29: Final step output of FSRP for P_c

$$Pc_i = 0.075 + (0.0014 * x_{i1}) + (0.005 * x_{i2}) + (-0.077 * x_{i3}) \quad (4.5\text{-}2)$$

where:

- x_{ij} are the values of the independent variables of the *i*th trial
  - x_{i1} is the unit size;
  - x_{i2} is the TTR; and
  - x_{i3} is the quality.

The results of using this predictive model on four evaluation corpora for Afrikaans spelling checkers provide proof that the model accurately predicts the expected variability when compared to actual error in results from validation units. This is further supported by the MSPR and MSE values that are 0.0000003 and 0.00000001 respectively. Since these two values are very close, the model is a good predictor of future behaviour according to the validation requirement set by Neter *et al.* (1996)

and Field (2005). There is one exception, where the actual error of SC C on the first validation unit falls outside of the predicted CI. Although this is an indicator of the an incorrect model, the model design only predicts with 95% accuracy, and in 5% of cases evaluations may fall outside the predicted values, but the model and modelling technique should still be considered valid since all other examples and tests have succeeded, and a single incorrect result does not invalidate the model.

Speller Name	Unit size	TTR	Quality	Predicted 95% CI	Actual error
A	0.164500	0.413000	99.82%	0.04%	0.04%
A	0.374500	0.359700	99.86%	0.04%	0.03%
A	0.495733	0.325600	99.90%	0.04%	-0.03%
A	0.765200	0.285560	99.89%	0.06%	-0.02%
B	0.164500	0.413000	99.45%	0.07%	-0.06%
B	0.374500	0.359700	99.34%	0.08%	0.05%
B	0.495733	0.325600	99.39%	0.08%	-0.03%
B	0.765200	0.285560	99.34%	0.10%	0.02%
C	0.164500	0.413000	99.41%	0.08%	**-0.12%**
C	0.374500	0.359700	99.25%	0.09%	0.06%
C	0.495733	0.325600	99.26%	0.09%	0.03%
C	0.765200	0.285560	99.29%	0.10%	0.00%

Table 30: Predicted CI for P_c of three Afrikaans spelling checkers and the actual errors on four validation units

In contrast to P_i, the model predictions for all of the evaluations fall well within the ±1% range expected of a reliable evaluation. This is directly correlated to the fact that the number of targets in the units is much higher than that of P_i. For P_c, about 99% of the words in the validation unit are correct and therefore the effect of missing some of the 1% of errors is much lower. For P_i the inverse is true where flagging correct words has a much more pronounced effect on the results of P_i. The only way to ensure reliable P_i metrics would be to balance the number of incorrect and correct words, which stands against the tenet of using corpora that are representative of the types of data produced by actual users rather than test suites.

4.6 MODELLING EVALUATION RELIABILITY: SUGGESTION ADEQUACY

4.6.1 STEP 7: VALIDATE SUGGESTIONS ADEQUACY

Ever since Damerau's (1964) initial identification of automatic correction as an important computational tool, a large body of research has focused on finding suggestions for and correcting erroneous textual input. Over the past four decades, different techniques, such as n-grams, minimum edit distance and rule-based techniques, have been applied to generate and rank possible corrections for incorrectly spelled words (Kukich, 1992:393). The aim of previous research has focused on two aspects of suggestion generation, namely finding possible suggestions and ranking them where no unambiguous suggestion can be found. These approaches have been widely implemented in different applications, most notably in optical character recognition systems and text editing applications. By far the most widely used application where the inclusion of error correction is evident to users, is in spelling checkers.

In the early 1990s, Green & Hendry (1993:84) explicitly state that a metric should be established to evaluate the suggestion adequacy (SA) of spelling checkers, but there is still little consensus on the actual implementation of such a metric. SA refers to a spelling checker's ability to present the end-user with relevant and accurate suggestions for all true negatives (i.e. incorrect words flagged by the spelling checker)[15]. Different researchers, like Paggio & Underwood (1997), Paggio & Music (1998), Reynhaert (2006), Van Zaanen & Van Huyssteen (2003), Van Huyssteen *et al.* (2004), and the TEMAA project (1997), have investigated methods for evaluating SA as part of their research on spelling checker evaluations. There are a number of different ways in which the adequacy of suggestions can be scored to attain a single metric. Each of these will be discussed briefly to illustrate how the scoring system would influence the SA result. An evaluation with each scoring system will then be done, to determine which is likely to be the most reliable metric for SA.

[15] Note that the SA of a spelling checker should only be based on true negatives, and not on all negatives, since the aim is to determine how well the spelling checker can suggest corrections for incorrect words it flags.

4.6.1.1 Percentile SA

In TEMAA's implementation (1997:33) of SA, the metric is not calculated as a single score, but rather as percentile scores of where the suggestions appear in the suggestion window. The SA metric therefore results in a table similar to Table 31.

1st suggestion correct (%)	Correct suggestion among 2-4 (%)	Correct suggestion not among first 4 (%)	No correct suggestion (%)	No suggestion (%)
70%	17%	4%	6%	3%

Table 31: Hypothetical example of the implementation of SA by TEMAA (Paggio & Underwood, 1997)

Errors are usually categorised according to the type of error in the incorrect word, and evaluations are then performed to determine the effectiveness of suggestions about specific types of errors. These errors are used to derive suggestions from the spelling checker, and determining the position of the correct suggestion in the list of suggestions provided (TEMAA, 1997:77). The percentage of correct suggestions for each type of error and the ranking of the suggestion is then given as the result of the evaluation (cf. Table 32).

TEMAA (1997:33) and Paggio & Underwood (1997:14) propose that the SA of a spelling checker should be determined as percentile values of the following four sub-categories:

- How often is the first suggestion the correct suggestion?
- How often is a visible suggestion the correct suggestion?
- How often are all visible suggestions wrong?
- How often are no suggestions presented at all?

This method does not give a score to the overall SA of the spelling checker, but it indicates what percentage of suggestions occur in each of the four categories. See Table 32 (Paggio & Underwood, 1997:15) as an example.

Product	Error types	No. of errors identified	1st suggestion correct (%)	Correct suggestion among 2-4 (%)	Correct suggestion not among first 4 (%)	No suggestion (%)
A	**r errors**	16	19	81	0	0
	Suffixation	235	9	90	1	0
	Silent letters	3160	81	11	3	5
	Substitution	1061	76	13	5	6
	Loan words	72	51	24	7	18
B	**r errors**	12	17	58	25	0
	Suffixation	235	1	34	65	0
	Silent letters	3057	76	13	7	4
	Substitution	1025	83	8	6	3
	Loan words	72	81	10	1	8

Table 32: Example of SA evaluation of two Danish spelling checkers taken from Paggio & Underwood (1997:15)

This approach to evaluation is of great value to developers because it indicates the performance of the spelling checker on specific error phenomena. The problem with this approach is that there is no clear indication of the overall SA of the spelling checker, or an intuitive feel of how well a spelling checker provides suggestions. Consequently, other researchers have proposed SA measures which are more descriptive of the spelling checker's ability to provide accurate suggestions. These SA metrics are based on attributing different values to suggestions through different types of scoring systems. The different scoring systems are discussed in the following sections.

4.6.1.2 Scoring systems

The first scoring system is very simple (henceforth Simple Scoring System (SSS)), as each instance where a correct suggestion is made scores one point, irrespective of the ranking of the suggestion. With this metric, a spelling checker can score only one or zero, depending on whether the correct suggestion is somewhere in the list of suggestions. This scoring system is preferable in the sense that without a context, a spelling checker cannot determine the most likely suggestion in cases where different suggestions have the same likelihood of being correct. With the SSS, however, no credit is given for spelling checkers with more accurate ranking procedures, or

detracted for spelling checkers that give large numbers of suggestions (for instance more than 10) where the correct suggestion is often lower in the list of possibilities.

The second scoring system is that of Van Zaanen & Van Huyssteen (2003:95), where a differentiation is made between those suggestions in the first position and the rest of the suggestions. Suggestions are given a score according to the position of each correct suggestion, using the following scores for each suggestion:

- Correct suggestion within first three suggestions (S1) = 1;
- Correct suggestion anywhere else in list of suggestions (S2) = 0.5; and
- No Correct suggestion = 0.

Because this scoring system groups the first three suggestions together, the influence of ranking in suggestions is not fully reflected. These scores are then added together and divided by the true negatives (*Tn*) to get a score for SA, as calculated by the following metric:

$$SA = \frac{\sum S1 + \frac{\sum S2}{2}}{Tn} \qquad (4.6\text{-}1)$$

Van Huyssteen *et al.* (2004:96) suggest a similar but more strict scoring system, where incorrect suggestions are penalised if the correct suggestion is not anywhere in the list of suggestions. Their scoring system is set out as follows:

- The correct suggestion is the first suggestion (S1) = 1;
- The correct suggestion is a visible suggestion (S2) = 0.5;
- All visible corrections are incorrect (S3) = -0.5;
- No suggestions are offered = 0.

The calculation of the SA is then done as follows:

$$SA = \frac{\sum S1 + \frac{\sum S2}{2} - \frac{\sum S3}{2}}{Tn} \qquad (4.6\text{-}2)$$

The argument for this scoring system is that a user who is not familiar with a language (e.g. a non-native user) might attempt to spell a word not knowing its correct spelling. If the spelling checker does not provide the correct suggestion, the user could choose an incorrect word because of his/her doubt about its spelling. An

example of this is where the word **representitivity* is incorrectly spelled. The spelling checker offers as a suggestion the word *representatively* (adverb form of representative), rather than *representativity*. If the knowledge of the user is inadequate he/she could use the suggestion, and in so doing change the meaning of the sentence or cause a grammatical error.

Although the discussed example may occur in some instances, penalising a system for trying to provide a correct suggestion and not penalising instances where no suggestion is made, seems too harsh. This type of scoring may give an unreliable evaluation of the spelling checker, since the suggestion functionality may be better than another spelling checker in most cases, but because of this penalisation will be reflected as being similar to, or worse than, an inferior system.

This is evident from Table 33, where SC A performs worse than SC B in every other scoring system, yet attains a higher score than SC B with the Van Huyssteen *et al.* (2004) metric. A spelling checker that gives more correct suggestions in the first position may therefore have a lower SA score, even though the spelling checker more often gives correct suggestions than another spelling checker.

Suggestion positions	**SC A**	**SC B**	**SC C**
Only one suggestion	3	254	172
First	1018	793	869
Second	115	94	82
Third	32	41	57
Fourth	26	29	22
Lower than fourth	36	47	53
No correct suggestions	81	100	93
No suggestions	54	19	27
Incorrect words flagged	1365	1377	1375
Simple scoring system	0.9011	0.9136	0.9127
Van Zaanen & Van Huyssteen (2003)	0.8245	0.8370	0.8349
Van Huyssteen *et al.* (2004)	0.7652	0.7643	0.7673

Table 33: Results of different SA scoring systems on three Afrikaans spelling checkers[16]

Of course, with all of these scoring systems, the ideal spelling checker would score 1, since it is expected of the spelling checker to provide only one (correct) suggestion

[16] It has to be noted that the scores attained here (and those presented in Table 34), unlike other metrics, are not percentage values, except for the SSS, as the scores that are awarded for lower ranked suggestions are merely arbitrary values that try to give numeric values to suggestions, rather than indicating the percentage of times a user can expect to get a correct suggestion.

for each invalid word that is flagged. This is not always possible, however, because an incorrect word often has more than one possible correction, which can only be disambiguated in the context of the sentence where the error occurs.

In Table 33 there is one possible problematic feature that could influence the results of the SA scores, namely the number of instances used in the calculation of SA. In the experimental results in Table 33, the number of incorrect words used to calculate SA differs between the spelling checkers. The argument that only the words which are flagged by all three spelling checkers should be used in a comparative evaluation can be made on the basis of excluding R_i from the calculation. Another argument could be made that because a spelling checker flags an incorrect word, a suggestion for that word should be made, and that this should be taken into account when the metric is calculated. In order to determine if this does indeed influence the metric, a different experiment, using only words flagged by all three spelling checkers was done, and the results are given in the following table.

Suggestion positions	**SC A**	**SC B**	**SC C**
Only one suggestion	3	241	168
First	1009	785	854
Second	111	88	86
Third	29	43	48
Fourth	23	22	19
Lower than fourth	33	44	47
No correct suggestions	81	93	82
No suggestions	48	21	33
Incorrect words flagged	1337	1337	1337
Simple scoring system	0.9035	0.9147	0.9140
Van Zaanen & Van Huyssteen (2003)	0.8302	0.8411	0.8392
Van Huyssteen *et al.* (2004)	0.7696	0.7715	0.7779

Table 34: Results of different SA scoring systems on three Afrikaans spelling checkers

From the results in this table, it can be seen that there is little or no difference in the type of results that are given by the SA measure when compared to the results in the previous experiment. For this reason, it is believed that all incorrect words flagged by the spelling checker should be used in the evaluation of SA, rather than excluding words that are not flagged by other spelling checkers. The basis for this argument is made from the usage-based perspective that all incorrect words that are flagged should be included in the evaluation, and secondly that all other researchers, such as

Paggio & Underwood (1997), TEMAA (1997) and Paggio & Music (1998) use all the flagged incorrect words in their evaluation of SA.

In order to get an idea of the difference in values attained from the different scoring systems, Table 35 gives the results reported by Paggio & Underwood (1997:15) (the percentage scoring in Table 32) for two Danish spelling checkers using the different scoring systems.

	Product A	**Product B**
Simple scoring system	0.9142	0.8657
Van Zaanen & Van Huyssteen (2003)	0.8367	0.8006
Van Huyssteen *et al.* (2004)	0.8026	0.7024

Table 35: SA of two Danish spelling checkers with different scoring systems

From these results, it is clear that the choice of scoring system is important to the results that will be attained by SA. In the first instance, the results of the SSS are considerably higher than the other two scores, and evaluate the spelling checkers SA much less severely than the other two scores. The results in Table 35 indicate that Product A more often gives correct suggestions, and correct suggestions in higher ranked positions. The differentiation between the different scoring systems is also interesting, as there is no linear progression in difference between the different spelling checkers. With the SSS there is a difference of 0.0485 between the spelling checkers; a 0.0361 difference with Van Zaanen & Van Huyssteen's (2003) scoring system; and a 0.1002 difference with Van Huyssteen *et al.* (2004). The smaller score difference of SA between the results of the two spelling checkers when Van Zaanen & Van Huyssteen's (2003) metric is applied, stems from the fact that Product B ranks the correct suggestion in the first position more often than Product A. The large number of instances where Product B provides only incorrect suggestions, especially on r- and suffixation errors (in Table 32), is reflected in the big difference in SA between the two spelling checkers when the metric suggested by Van Huyssteen *et al.* (2004) is applied. This indicates that the type of scoring system that is used has a significant effect on the perceived difference of SA.

The following section will attempt to answer two questions. First there will be a statistical analysis of the predictor variables using the FSRP to identify the variables that are the best predictors of variability for the SA metric. After identifying the

predictor variables, the effect and stability of the different SA metrics will be investigated, in order to decide which one is the most accurate reflection of a spelling checker's suggestion functionality.

4.6.2 STEP 8: SELECT INDEPENDENT VARIABLES FOR SA

From the previous discussion, the following possible predictor variables have been identified for the evaluation of SA:

a. unit size;
b. error TTR;
c. error type; and
d. quality.

Once more, the set of predictor variables identified for determining SA is processed by FSRP to determine the set of good independent variables for predicting the CI of SA. For this regression model, the minimum acceptable *F* limit for adding a variable to the model is 3.88, which is determined by $F(0.95; 1, 217)$. At each step in FSRP, the variable with the highest *F* over 3.88 is added to the regression model. In the first step, unit size has the highest *F* statistic, 231.02, and is entered into the model. The values in Table 36 represent the remaining steps in FSRP, and add the substitution percentage variable in step 1. In the final step, none of the remaining variables have a sufficiently high *F* statistic to be added to the model.

The obvious oddity in these results is that neither the error TTR nor the quality of the suggestions is correlated to the variability of the results. Error TTR may not be particularly indicative of variability, since the ratio is very high in most of the evaluations performed for the modelling. In other words, most of the errors in each modelling unit are unique, with very few overlapping errors. This is to be expected, since it is unlikely that multiple errors will occur in the same way multiple times in a single unit. As far as the quality variable is concerned, there is no obvious reason why it should not be a good predictor of variability, as is the case for all the other metrics.

Candidate variables	*F* statistic	Numerator degrees of freedom	Denominator degrees of freedom
Step 1			
Error TTR	0.2952	1	217
Substitution %	**7.7852**	1	217
Deletion %	3.6825	1	217
Insertion %	0.011	1	217
Transposition %	0.097	1	217
Quality	0.1967	1	217
Step 2			
Error TTR	0.028	1	216
Deletion %	0.2248	1	216
Insertion %	0.0618	1	216
Transposition %	0.4235	1	216
Quality	0.001	1	216

Table 36: Steps 1-2 FSRP results for SA

4.6.2.1 Hypotheses

Based on the previous discussion, the following hypotheses are tested to determine the influence of variables on evaluation results for the SA metric:

a. The SSS is an accurate measure of SA functionality.
b. The larger the unit size, the more reliable the SA metric is.

4.6.2.2 Experiments

a. The SSS is an accurate measure of SA functionality

From the discussion in 4.6.1.2 it is clear that the scoring system that is used in the evaluation of SA will have a marked influence on the results reported by SA evaluations. However, it is unclear which scoring system provides the most accurate reflection of a spelling checker's suggestion functionality. The single biggest problem with determining the scoring system is that any choice is arbitrary because it is almost impossible to indicate at a theoretical level that the one scoring system more accurately reflects the suggestion functionality of a spelling checker than another. From the results in Table 33 it is clear that the different scoring systems give very different reports on the suggestion functionality. In this section, the reliability of each of these scoring systems will be evaluated to determine if there is a difference in

reliability between the metrics, which can then be used as a scientific basis for choosing one system over another.

The results reported in Table 37 are the σ of scores attained by three Afrikaans spelling checkers on 10 test sets of 120 incorrectly spelled *Tn* instances. 120 words were used, as this is the average of minimum incorrect instances that will appear in a text of 15,000 words, based on the 0.8% error frequency reported by Kukich (1992), Pollock & Zamora (1984) & Verberne (2002). The results indicate that the SSS gives results that have a narrower distribution than the other two scoring systems. On this basis it can be argued that this SSS is the most reliable reflection of the SA attribute of a spelling checker. This stability can be attributed to the fact that the SSS does not take ranking into account, and variance in ranking does not have an effect on the metric, as it does with the other two metrics.

	SC A	**SC B**	**SC C**
Simple scoring system	0.12	0.08	0.10
Van Zaanen & Van Huyssteen (2003)	0.53	0.43	0.47
Van Huyssteen *et al.* (2004)	0.76	0.71	0.78

Table 37: σ of different scoring systems on three Afrikaans spelling checkers

It can be argued that because the SSS does not take ranking into account, which has previously been indicated to be one of the important aspects of SA to be evaluated, this does not accurately reflect the experience of the end-user. On the other hand, one can also argue that ranking is not really important to an end-user, and that the correct suggestion should be anywhere in the list of suggestions for this attribute to be useful to the user. This again highlights the fact that the choice of scoring system is to a large extent arbitrary, and for this reason the choice here is based exclusively on the reliability of the metric, in which case the SSS is the best choice. Furthermore, using the SSS is the only way of getting a score that can be converted to a percentage, as it represents the percentage of times a user can expect the spelling checker to provide a correct suggestion for an incorrectly spelled word.

b. The larger the unit size, the more reliable the SA metric is

As with most of the metrics discussed so far, the number of instances used in the evaluation of SA has an influence on the results reported by the metric. For this

hypothesis to be validated, the three Afrikaans spelling checkers were used to evaluate SA results on modelling sets of different sizes, ranging from 100 to 600 incorrect words. The variation in each of these sets is compared to the variation of the next larger set. If the variance between sets is not significant, this will indicate that the metric is stable, and therefore reliable.

In each case, the incorrect word is paired with the correct suggestion, and the suggestion from the spelling checker is compared with this correct suggestion. Scores are awarded to each spelling checker according to the SSS, and the results of these tests are summarised in Table 38.

	SC A	**SC B**	**SC C**
100 & 200	0.29	0.35	0.32
200 & 300	0.43	0.41	0.44
300 & 400	0.57	0.61	0.59
400 & 500	0.76	0.84	0.82
500 & 600	0.84	0.87	0.86

Table 38: p scores for SA of three Afrikaans spelling checkers.

The results in Table 38 indicate that even with a fairly small number of *Tn,* it is possible to get results where the difference between results is not statistically significant, as the variance between 100 and 200 input words is not significant ($p > 0.05$ in all evaluations). This provides the required evidence that the number of incorrect words found in a 15,000 word text at an error rate of 0.8% (120 errors) should be large enough to give reliable SA results, and therefor tests this functionality of a spelling checker reliably.

4.6.3 STEP 9: SELECT AND VERIFY RELIABILITY MODEL FOR SA

The predictive regression model for predicting the variability in SA is described by function 4.6-1 based on the final step output from the FSRP for SA performed in 4.6.2.

Constant = 0.065; Std. Error of estimate = 0.012

Variable	β	Std. Error
Unit size	-0.054	0.003
Substitution %	-0.036	0.013

Table 39: Final step output of FSRP for SA

$$SA_i = 0.065 + (-0.054 * x_{i1}) + (-0.036 * x_{i2}) + 0.012 \quad (4.6\text{-}3)$$

where:

- x_{ij} are the values of the independent variables of the *i*th trial
  - x_{i1} is the unit size; and
  - x_{i2} is the % of errors that are substitution errors.

The results of using this predictive model on four evaluation corpora for Afrikaans spelling checkers provide proof that the model accurately predicts the expected variability when compared to actual error in results from validation units. This is further supported by the MSPR and MSE values that are 0.0008763 and 0.0009026 respectively. Since these two values are very close, the model is a good predictor of future behaviour, according to the validation requirement set by Neter *et al.* (1996) and Field (2005).

It is interesting that even though the experiments in 4.6.2.2 indicated that variability in SA results are not statistically significant when the validation unit has more than 100 errors, the 95% interval predicted by the regression model is much larger than the expected ±1% interval expected of a reliable evaluation. This means that even though the variance between sets may not be significant, the actual error cannot ensure reliable evaluations with such small units.

Speller name	Unit size	Substitution %	Predicted 95% CI	Actual error
A	0.06731	35.00%	±6.08%	6.03%
A	0.21731	54.90%	±4.55%	2.46%
A	0.36154	57.53%	±3.68%	-2.08%
A	0.57308	52.21%	±2.73%	-0.33%
B	0.06731	35.00%	±6.08%	-2.36%
B	0.21731	54.90%	±4.55%	-2.65%
B	0.36154	57.53%	±3.68%	2.70%
B	0.57308	52.21%	±2.73%	-1.26%
C	0.06731	35.00%	±6.08%	5.70%
C	0.21731	54.90%	±4.55%	-2.04%
C	0.36154	57.53%	±3.68%	-1.64%
C	0.57308	52.21%	±2.73%	1.87%

Table 40: Predicted CI for SA of three Afrikaans spelling checkers and the actual errors on four validation units

Another aspect of the SA regression model to take note of is that, unlike the regression models for the other metrics, the predictions for SA are not dependent on the particular speller, but are exclusively dependent on the variables in the test set, which means that the predicted intervals are the same for all three spelling checkers on each of the four SA validation units.

4.7 MODELLING EVALUATION RELIABILITY: OVERALL LINGUISTIC ADEQUACY METRICS

Ever since Van Rijsbergen (1979:174) introduced the *E* (effectiveness) measure as a combinatory metric for information retrieval, different metrics have been established which attempt to combine more than one measure, to find an overall metric for the attributes of an NLP application. Some of these, such as the F-measure, have been widely accepted in the NLP evaluation community (Jurafski & Martin, 2000; Kim *et al.*, 2012; Luo *et al.*, 2012; Manning & Schütze, 1999; Rennie, 2004; Sasaki, 2007; Smrž, 2004), but there are very few instances of this kind of metric for the evaluation of spelling checkers. In this section, different possible metrics are described, which could fulfil this function. Each metric is first theoretically validated to determine whether the metric accurately describes and represents the functionality attribute. These metrics are then implemented in evaluations in order to validate each empirically. In subsection 4.7.3 a critical overview of the metrics and their reliability

is given, with the intention of finding the metric that provides the most reliable and descriptive representation of the overall functionality of the spelling checker.

4.7.1 STEP 7: VALIDATE METRICS

4.7.1.1 Predictive accuracy

Starlander & Popescu-Belis (2002) propose the predictive accuracy PA (PA) measure, which is a prediction of the likelihood that any given word, correct or incorrect, will be accurately classified by the spelling checker. PA is calculated as follows:

$$PA = \frac{Tp+Tn}{Tp+Tn+Fp+Fn} \qquad (4.7\text{-}1)$$

This gives a good overall view of the linguistic efficiency of a spelling checker, since it determines how accurate a spelling checker is at performing both of the tasks it is expected to perform. By means of this metric, one can express what the spelling checker does right (i.e. the sum of *Tp* and *Tn*) in terms of everything that the spelling checker does (i.e. all true and false positives and negatives). This can be interpreted to be the accuracy with which any word that the spelling checker encounters will be handled. Like many of the other scores, this score is represented as a fraction between 0 and 1, which can be interpreted as a percentage value. Theoretically, this metric gives an accurate description of a spelling checker's overall functionality.

4.7.1.2 F-measures

As an extension of the four basic attribute metrics (R_c, P_c, R_i, and P_i), Starlander & Popescu-Belis (2002:275) have introduced two combinatory metrics based on the F-measures that relate the performance of recall and precision on correct and incorrect words. The F-measure has been widely used in NLP evaluations, especially in the evaluation of IR systems (Manning & Schütze, 1999:269; Sasaki, 2007; Smrž, 2004:206; Theeramunkong & Okumura, 1997:36), as well as NE recognition systems (CoNLL, 2002; 2003; Kim *et al.*, 2012; Luo *et al.*, 2012; Puttkammer, 2006: 81), though it has not been widely used in the evaluation of spelling checkers.

F-measures are functions that calculate a harmonic mean between the recall and precision measures, rather than an average (or arithmetic mean) of the values

(Devore & Peck, 1986:67; Sasaki, 2007). The harmonic mean is a mathematical measure of the central tendency of a random variable, which can never be zero (Limbrunner *et al.*, 2000:59), and is most commonly used in the field of engineering. The harmonic mean does not give the average of a given set of values, but rather a weighted average.

Burk (1985), as referred to by Limbrunner *et al.* (2000:60), proves that the harmonic mean of random variables is always smaller than the geometric mean, which is always smaller than the arithmetic mean, but is always greater than the smallest instance. The implication of using the harmonic mean rather than the average or geometric mean, is that it tends to be more severe on lower scores. It therefore gives a more critical evaluation of a system's performance in terms of all the functionality, and preventing trivial strategies from being over-emphasised when reporting evaluation results.

Trivial strategies in the context of spelling checkers can be described as instances where too many words are either accepted as correct or incorrect, thereby decreasing the precision of the system. As an example, a spelling checker that flags all words as incorrect (flagging correct words as incorrect) would have a very high R_i, but a low P_c. Using the F-measure will report negatively on these types of trivial strategies .

An additional aspect of F-measures that makes these measures worth considering is the fact that F-measures can be adapted, by giving different weights to precision and recall scores according to the perceived importance of the different sub-attributes (Chantree *et al.*, 2005:6). Because the functionality of a spelling checker relates both to its coverage of the language and its identification of errors, the F-measure has to be calculated for both the correct words and the errors. Different formulations of the F-measure are found in the literature, and although the formulations appear different, all calculate the same results. For this reason, only the calculation put forward by Starlander & Popescu-Belis (2002:275) is given here. The F-measures for correct and incorrect words are calculated as follows:

$$fm_c = \frac{x}{\frac{y}{P_c}+\frac{z}{R_c}} \quad (4.7\text{-}2) \qquad fm_i = \frac{x}{\frac{y}{P_i}+\frac{z}{R_i}} \quad (4.7\text{-}3)$$

Weighting can be done by applying the ratios on y and z in 4.6-1 and 4.6-2 where the ratio of P_c:R_c or P_i:R_i is represented by y:z and $x = y + z$. In order to get a 3:1 ratio in favour of precision, the formula would be applied as follows:

$$fm_c = \frac{4}{\frac{3}{P_c}+\frac{1}{R_c}} \quad (4.7\text{-}4) \qquad fm_i = \frac{4}{\frac{3}{P_i}+\frac{1}{R_i}} \quad (4.7\text{-}5)$$

Although this weighting option is available, it is very rarely used in practice, since there is little theoretical ground for determining that either precision or recall is more important in calculating either of these F-measures. As such, this study uses the standard balanced weighting where

$x = 2$; and

$y = z = 1$.

4.7.1.3 Mean time between failures

Another possible overall metric suggested by Riley *et al.* (2004) is an engineering metric known as mean time between failures (MTBF), which calculates the failure rate of a system in relation to the total time the system is active. This will then give the evaluator the estimated time a system works correctly before a problem or error occurs, and is calculated as follows:

$$MTBF = \frac{TotalTime}{NumberOfFailures} \quad (4.7\text{-}6)$$

As an example, a system that fails 14 times in 3,600 minutes can be said to have a MTBF of 257.14 minutes. Riley *et al.* (2004) propose adapting the MTBF metric to determine how many false flags are assigned per page of text, in order to represent pages rather than time, as follows:

$$FFPP = \frac{Fn}{NumberOfPages} \quad (4.7\text{-}7)$$

The false flags per page (FFPP) metric is a good descriptive replacement metric for P_i, as it indicates how many times a user has to ignore flags assigned by the grammar checker on each page, as opposed to a percentage figure. Scores from this metric represent how a user will see a page of text, and the lower the FFPP score, the fewer flags are assigned to correct words. Assuming that a spelling checker with an R_c score of more than 99% is a good spelling checker, the FFPP score should be less than 3[17].

[17] Given an average of 300 words per page, 99% R_c means flagging 3 correct words per page on average.

Each additional 3 FFPP equates to approximately 1% drop in the R_c of the spelling checker (i.e. a spelling checker with an R_c of 94% would have 18 FFPP).

There are two problems with this metric. First, the number of words per page varies depending on variables such as font size, font types, and line spacing. Second, the approach of Riley *et al.* (2004) is only concerned with *Fn* and not with *Fp*. As a grammar or spelling checker should also be concerned with the flagging of all incorrect instances, their approach is adapted in the following section to include *Fp*, and is referred to as false instances per page (FIPP).

4.7.1.4 False instances per page

The FIPP measure is appealing because it gives an indication of how often a user has to correct or ignore errors made by the spelling checker in terms of a single page. This not only represents P_i, but also P_c, and is therefore a better overall metric than FFPP.

As previously discussed, the fact that FFPP does not take the number of words that constitute a page into account, the metric is not ideal for comparative evaluations. However, the argument can also be made that because this metric attempts to describe the user experience, this should not be regulated. In addition, the fact that different languages have differing averages for the number of words per page, this is reflected in the metric, as that is how the user would experience the spelling checking session. On the basis of these two arguments, I prefer that the number of words not be controlled, as this compensates for the differing nature of languages. There is, however, a case to be made to regulate the font and page layout if the metric is used comparatively. For this reason it makes sense to calculate FIPP in terms of the number of pages, where each "page" is defined by a set number of characters. Based on a widely used font, Pt. 12 Arial with 1.15 spacing, there are approximately 2,500 characters per page on average. Using that assumption, FIPP can be calculated as follows:

$$FIPP = \frac{Fn+Fp}{NumberOfPages} \qquad (4.7\text{-}8)$$

4.7.2 STEP 8: SELECT INDEPENDENT VARIABLES FOR COMBINATORY METRICS

The nature of combinatory metrics is that they combine one or more of the more specific evaluation metrics discussed in the previous sections. For this reason the set of independent variables that may be relevant to predicting the reliability of any of the combinatory metrics should be the union of all of the variables identified for each of the individual metrics. The following is the union of those identified items:

a. unit size;
b. TTR;
c. error TTR;
d. heterogeneity; and
e. quality.

As was done for the other metrics, FSRP will be used to determine which of these independent variables are good predictors of variability for each individual metric. Based on the results of FSRP, models for each metric are created and verified. FSRP is used to select those variables that are good independent variables for accurately modelling the expected variability of the combinatory metrics. FSRP selects the model with the best predictive capability, given the results of a large set of evaluations performed on modelling sets selected from the modelling corpora.

The data selection process for each of the modelling units used to do this modelling experiment randomly selected 70 modelling units containing between 1,000 and 30,000 words. In both cases the only restriction on selecting the number of words is that a document can only be selected once, so no data duplication would be present in the 70 modelling units. For this regression model, the minimum acceptable F limit for adding a variable to the model is 3.88, which is determined by $F(0.95; 1, 217)$. At each step in FSRP, the variable with the highest F over 3.88 is added to the regression model.

4.7.2.1 Predictive accuracy

For predictive accuracy, the first variable entered into the model is the TTR variable which has an F statistic of 220.427. The results of the remaining steps of FSRP are given in Table 41 and show that quality and unit size variables are the other

independent variables that are relevant to predicting the CI of PA. In the final step, neither error TTR nor heterogeneity have a sufficiently high *F* statistic to be included in the model.

Candidate variables	*F* statistic	Numerator degrees of freedom	Denominator degrees of freedom
Step 1			
Unit size	5.7703	1	216
Error TTR	0.1516	1	216
Heterogeneity	0.1327	1	216
Quality	**431.8109**	1	216
Step 2			
Unit size	**17.3473**	1	215
Error TTR	1.4216	1	215
Heterogeneity	1.3057	1	215
Step 3			
Error TTR	1.7642	1	214
Heterogeneity	0.3523	1	214

Table 41: Steps 1-3 FSRP results for PA

4.7.2.2 fm_i

For fm_i, the first variable entered into the model is the quality variable which has an *F* statistic of 216.444. The results of the remaining steps of FSRP are given in Table 42 and show that TTR and unit size are the other independent variables that are relevant to predicting the variability of fm_i. In the final step, neither error TTR nor heterogeneity has a sufficiently high *F* statistic to be included in the model.

Candidate variables	*F* statistic	Numerator degrees of freedom	Denominator degrees of freedom
Step 1			
Unit size	217.5903	1	216
TTR	**295.026**	1	216
Error TTR	56.5552	1	216
Heterogeneity	0.3094	1	216
Step 2			
Unit size	**5.5812**	1	215
Error TTR	2.2095	1	215
Heterogeneity	2.5484	1	215
Step 3			
Error TTR	2.4094	1	214
Heterogeneity	1.6405	1	214

Table 42: Steps 1-3 FSRP results for fm_i

4.7.2.3 fm_c

For fm_c, the first variable entered into the model is the TTR variable which has an *F* statistic of 1142.346. The results of the remaining steps of FSRP are given in Table 43, and show that unit size and quality are the other independent variables that are relevant to predicting the variability of fm_c. In the final step, neither error TTR nor heterogeneity has a sufficiently high *F* statistic to be included in the model.

Candidate variables	*F* statistic	Numerator degrees of freedom	Denominator degrees of freedom
Step 1			
Unit size	**53.2264**	1	216
Error TTR	0.583	1	216
Quality	4.059	1	216
Heterogeneity	0.0094	1	216
Step 2			
Error TTR	1.0205	1	215
Quality	**4.9168**	1	215
Heterogeneity	0.8912	1	215
Step 3			
Error TTR	1.1663	1	214
Heterogeneity	0.7974	1	214

Table 43: Steps 1-3 FSRP results for fm_c

4.7.2.4 FIPP

Lastly, for FIPP, the first variable entered into the model is the TTR variable which has an *F* statistic of 239.037. The results of the remaining steps of FSRP are given in Table 44, and show that quality and unit size are the other independent variables that are relevant to predicting the variability of FIPP. In the final step, neither error TTR nor heterogeneity has a sufficiently high *F* statistic to be included in the model.

Candidate variables	***F* statistic**	**Numerator degrees of freedom**	**Denominator degrees of freedom**
Step 1			
Unit size	3.5847	1	216
Error TTR	0.1371	1	216
Quality	**432.5077**	1	216
Heterogeneity	0.0397	1	216
Step 2			
Unit size	**10.4496**	1	215
Error TTR	1.3452	1	215
Heterogeneity	0.7339	1	215
Step 3			
Error TTR	1.5774	1	214
Heterogeneity	0.1737	1	214

Table 44: Steps 1-3 FSRP results for FIPP

From the FSRP results presented here, all of the combinatory metrics have the same set of best independent variables for predicting the CI for each of the metrics. The following variables are used in the next step of the methodology to create the predictive model for each metric:

- unit size;
- TTR; and
- quality.

All of the results in Table 44 are in line with the previous results, except that the number of incorrect words was present in the calculation of R_i but does not have the predictive capacity for the combinatory metrics. A likely reason for this is that the number of errors in a text is highly correlated with the number of correct words in the text, and so the predictive capacity of the incorrect words is already accounted

for by a correlated variable, namely unit size, and the number of incorrect words does not contribute any additional information.

The results from FSRP can now be used to create linear regression models that describe the expected CI of these individual metrics, which can be used to predict variability of validation units for future evaluations and spelling checkers. These models are discussed and validated in the next section.

4.7.3 STEP 9: SELECT AND VERIFY RELIABILITY MODELS FOR COMBINATORY METRICS

As before, FSRP is used to create regression models that predict the CI for each of the metrics. Once again, the aim here is to determine whether a particular evaluation is reliable on the basis of the definition of reliability provided in 2.2.

4.7.3.1 Predictive accuracy

The final step output of FSRP for predictive accuracy presented in Table 45 provides the necessary information to generate the linear regression model in 4.6-9.

Constant = 0.030; Std. Error of estimate = 0.000

Variable	β	Std. Error
Correct TTR	0.020	0.002
Quality	-0.031	0.001
Unit size	0.002	0.000

Table 45: Final step output of FSRP for predictive accuracy

$$PA_i = 0.03 + (0.02 * x_{i1}) + (-0.031 * x_{i2}) + (0.002 * x_{i3}) \quad (4.7\text{-}9)$$

where:

- x_{ij} are the values of the independent variables of the *i*th trial
  - x_{i1} is the TTR;
  - x_{i2} is the quality of the tools; and
  - x_{i3} is the unit size.

This model can now be used to generate predictions on the same validation units used for all the other evaluation metrics. The results presented in Table 46 show that the model does accurately predict the expected CI of all of the validation sets. This is

further supported by the comparison of MSE to MSRP where the values (0.0000011 and 0.0000035 respectively) are very close together, thus validating the predictive capability of the model.

Speller Name	Unit size	TTR	Quality	Predicted 95% CI	Actual error
A	0.164500	0.413000	99.22%	±0.78%	-0.08%
A	0.374500	0.359700	99.10%	±0.72%	0.04%
A	0.495733	0.325600	99.19%	±0.68%	-0.06%
A	0.765200	0.285560	99.14%	±0.65%	0.00%
B	0.164500	0.413000	95.30%	±0.90%	0.23%
B	0.374500	0.359700	95.52%	±0.83%	0.02%
B	0.495733	0.325600	95.53%	±0.79%	-0.02%
B	0.765200	0.285560	95.57%	±0.76%	-0.05%
C	0.164500	0.413000	95.54%	±0.90%	-0.06%
C	0.374500	0.359700	95.23%	±0.84%	0.25%
C	0.495733	0.325600	95.41%	±0.79%	0.05%
C	0.765200	0.285560	95.51%	±0.76%	-0.04%

Table 46: Predicted CI for PA of three Afrikaans spelling checkers and the actual errors on four validation units

4.7.3.2 fm_i

The final step output of FSRP for fm_i represented in Table 47 provides the necessary information to generate the linear regression model presented in 4.6-10.

Constant = 0.037; Std. Error of estimate = 0.005

Variable	β	Std. Error
Quality	-0.099	0.004
Correct TTR	0.176	0.026
Unit size	0.012	0.005

Table 47: Final step output of FSRP for fm_i

$$fmi_i = 0.037 + (0.176 * x_{i1}) + (-0.099 * x_{i2}) + (0.012 * x_{i3}) + 0.005 \quad (4.7\text{-}10)$$

where:

- x_{ij} are the values of the independent variables of the *i*th trial
  - x_{i1} is the TTR;
  - x_{i2} is the quality; and
  - x_{i3} is the unit size.

This model can now be used to predict the expected CI for each of the validation units used to validate the predictive models of the other evaluation metrics. As before, the actual error for all the validation tests falls well within the predicted CI. This is further supported by the comparison of MSE to MSRP where the values (0.0001154 and 0.0003565 respectively) are very close together, thus validating the predictive capability of the model.

Speller Name	Unit size	TTR	Quality	Predicted 95% CI	Actual error
A	0.164500	0.413000	85.62%	±3.19%	-2.31%
A	0.374500	0.359700	82.77%	±2.79%	0.52%
A	0.495733	0.325600	83.86%	±2.22%	-0.67%
A	0.765200	0.285560	83.08%	±1.92%	0.25%
B	0.164500	0.413000	41.08%	±7.60%	0.49%
B	0.374500	0.359700	41.23%	±6.90%	0.62%
B	0.495733	0.325600	41.86%	±6.38%	-0.47%
B	0.765200	0.285560	41.62%	±6.02%	-0.18%
C	0.164500	0.413000	42.11%	±7.50%	-1.59%
C	0.374500	0.359700	38.53%	±7.17%	2.15%
C	0.495733	0.325600	39.71%	±6.59%	0.63%
C	0.765200	0.285560	40.63%	±6.12%	-0.29%

Table 48: Predicted CI for fm_i of three Afrikaans spelling checkers and the actual errors on four validation units

4.7.3.3 fm_c

The final step output of FSRP for fm_c represented in Table 49 provides the necessary information to generate the linear regression model presented in 4.6-11.

Constant = -0.067; Std. Error of estimate = 0.003

Variable	β	Std. Error
Correct TTR	0.215	0.013
Unit size	0.018	0.002
Quality	-0.002	0.001

Table 49: Final step output of FSRP for fm_c

$$fmc_i = -0.067 + (0.215 * x_{i1}) + (-0.002 * x_{i2}) + (0.018 * x_{i3}) + 0.003 \quad (4.7\text{-}11)$$

where:

- x_{ij} are the values of the independent variables of the *i*th trial

- x_{i1} is the TTR;
- x_{i2} is the quality; and
- x_{i3} is the unit size.

This model can now be used to predict the expected CI for each of the validation units used to validate the predictive models of the other evaluation metrics. As before, the actual error for all of the validation tests falls well within the predicted CI. This is further supported by the comparison of MSE to MSRP where the values (0.0002997 and 0.00021966 respectively) are very close together, thus validating the predictive capability of the model.

Speller Name	Unit size	TTR	Quality	Predicted 95% CI	Actual error
A	0.164500	0.413000	95.20%	±2.59%	1.55%
A	0.374500	0.359700	96.31%	±1.45%	0.70%
A	0.495733	0.325600	97.57%	±0.87%	-0.83%
A	0.765200	0.285560	97.17%	±0.52%	-0.42%
B	0.164500	0.413000	84.86%	±3.27%	-3.00%
B	0.374500	0.359700	81.09%	±2.95%	1.51%
B	0.495733	0.325600	82.83%	±2.33%	-1.04%
B	0.765200	0.285560	81.30%	±2.10%	0.51%
C	0.164500	0.413000	83.51%	±3.40%	**-4.01%**
C	0.374500	0.359700	78.19%	±3.24%	1.94%
C	0.495733	0.325600	78.56%	±2.75%	0.96%
C	0.765200	0.285560	79.44%	±2.28%	-0.05%

Table 50: Predicted CI for fm_c of three Afrikaans spelling checkers and the actual errors on four validation units

4.7.3.4 FIPP

The final step output of FSRP for FIPP represented in Table 51 provides the necessary information to generate the linear regression model presented in 4.6-12.

Constant = -1.991; Std. Error of estimate = 0.109

Variable	β	Std. Error
Correct TTR	5.637	0.559
Quality	0.030	0.001
Unit size	0.350	0.108

Table 51: Final step output of FSRP for FIPP

$$FIPP_i = -1.991 + (5.637 * x_{i1}) + (0.030 * x_{i2}) + (0.350 * x_{i3}) + 0.109$$

(4.7-12)

where:

- x_{ij} are the values of the independent variables of the *i*th trial
  - x_{i1} is the TTR;
  - x_{i2} is the quality; and
  - x_{i3} is the unit size.

This model can now be used to predict the expected CI for each of the validation units used to validate the predictive models of the other evaluation metrics. As before, the actual error for all of the validation tests falls well within the predicted CI. This is further supported by the comparison of MSE to MSRP where the values (0.0249942 and 0.0472933 respectively) are very close together, thus validating the predictive capability of the model.

Speller Name	Unit size	TTR	Quality	Predicted 95% CI	Actual error
A	0.164500	0.413000	2.340000	±0.573856	0.252000
A	0.374500	0.359700	2.700000	±0.357704	-0.126000
A	0.495733	0.325600	2.440000	±0.200114	0.166000
A	0.765200	0.285560	2.592000	±0.073282	-0.010000
B	0.164500	0.413000	14.100000	±0.926656	-0.682000
B	0.374500	0.359700	13.440000	±0.679904	-0.068000
B	0.495733	0.325600	13.400000	±0.528914	0.072000
B	0.765200	0.285560	13.284000	±0.394042	0.162000
C	0.164500	0.413000	13.380000	±0.905056	0.186000
C	0.374500	0.359700	14.310000	±0.706004	-0.744000
C	0.495733	0.325600	13.780000	±0.540314	-0.146000
C	0.765200	0.285560	13.476000	±0.399802	0.134000

Table 52: Predicted CI for FIPP of three Afrikaans spelling checkers and the actual errors on four validation units

Although the results in this table show that these models are mostly good predictor models, there are two instances where the estimates do not fall within the predicted range (highlighted in the tables). Since the models are only generated at 95%, these outliers should be expected, and do not invalidate the models. One more important aspect to note is that for PA, fm_i and fm_c, the reliability requirement set out in 2.2 does not directly hold for FIPP, since the metric does not report a fractional value between 0 & 1 where a set variability of value of ±1% is possible. One can, however,

specify that the variability should be below a threshold of 1 flag per page (i.e. the predicted CI should be less than ±0.5 flags per page). As with the definition of the 1% CI, this value is at the discretion of the evaluator, but is easily reportable for others to take into account when considering reported results.

4.8 CONCLUSION

In this chapter the second phase of the reliability evaluation methodology was applied to spelling checkers for each metric identified in the previous chapter. Each metric was initially validated, after which the independent variables for each metric were identified, using FSRP. For each identified variable, additional experiments for each metric provided additional insight into the relationship between each variable and each metric. Finally a regression model with these variables was selected and validated using additional validation units. This chapter targeted the following research goals:

a. To design a stepwise methodology for modelling evaluation reliability with linear regression procedures.
b. To provide a structured method for identifying variables that are predictors of evaluation reliability.
c. To provide a detailed investigation of how variables affect evaluation reliability.

The first part of the chapter focused on the evaluation of the reliability of widely used spelling checker evaluation metrics, and the variables that have an influence on these metrics. Table 53 provides the list of variables considered in this study and the metrics that each variable influences.

	R_c	R_i	P_i	P_c	PA	fm_i	fm_c	FIPP	SA
Modelling unit size	x	x	x	x	x	x	x	x	x
Number of errors		x	x		x	x	x	x	x
TTR	x	x		x	x	x	x	x	x
Error TTR		x	x		x	x	x	x	x
Quality	x	x	x	x	x	x	x	x	x
Target density			x	x	x	x	x	x	x
Error types									x
Word frequency	x				x	x	x	x	
Types or tokens	x	x	x	x					x
Usage-based or automatically generated data		x	x						x

Table 53: Matrix of independent variables for various spelling checker metrics

As part of the variable identification using FSRP, a set of linear regression models is generated that make it possible to predict what the variance of an evaluation is for a particular metric, given the values for the set of variables inherent in an evaluation. These models can be used to provide future predictions of result reliability for individual spelling checker evaluations. Given these models, an evaluator can make a decision about whether to adjust the corpus on which they are doing the evaluations, to ensure that the evaluation provides reliable results. The following models were created based on the evaluations performed in this study. More details on the variables of each model are available in the section references following each predictor model:

$$Rc_i = 0.026 + (0.016 * x_{i1}) + (-0.031 * x_{i2}) + (-0.001 * x_{i3}) + 0.003$$

$$Ri_i = 0.102 + (-0.036 * \mathrm{x_{i2}}) + (-0.076 * \mathrm{x_{i1}}) + 0.007$$

$$Pi_i = -0.078 + (0.021 * x_{i1}) + (0.215 * x_{i2}) + (0.021 * x_{i3}) + 0.002$$

$$Pc_i = 0.075 + (0.0014 * x_{i1}) + (0.005 * x_{i2}) + (-0.077 * x_{i3})$$

$$PA_i = 0.03 + (0.02 * x_{i1}) + (-0.031 * x_{i2}) + (0.002 * x_{i3})$$

$$fmi_i = 0.037 + (0.176 * x_{i1}) + (-0.099 * x_{i2}) + (0.012 * x_{i3}) + 0.005$$

$$fmc_i = -0.067 + (0.215 * x_{i1}) + (-0.002 * x_{i2}) + (0.018 * x_{i3}) + 0.003$$

$$FIPP_i = -1.991 + (5.637 * x_{i1}) + (0.030 * x_{i2}) + (0.350 * x_{i3}) + 0.109$$

Although several evaluation variables have been identified and included in the initial modelling investigations, only three variables have a statistically significant relationship with the variability seen in spelling checker error detection evaluation metrics, namely:

- unit size;
- TTR; and
- quality.

This means that for Afrikaans spelling checker evaluations, controlling for these three variables can relatively easily ensure reliable evaluation results. Each of these models should be applicable to any future Afrikaans spelling checker evaluation, but is likely to be more widely applicable. This assumption is investigated in more detail in the next chapter when testing the same methodology on different technologies and languages.

Apart from the error detection metrics, spelling checkers also provide corrections for errors, measured by the suggestion adequacy metric. The same variable selection procedure for suggestion adequacy as for the detection metrics was performed, with slightly different results. Only two of the variables identified for suggestion adequacy are good predictors of variability, namely unit size and percentage of substitution errors. It was further shown that very large validation units are necessary to ensure reliable evaluations, as validation units with 300 errors do not ensure reliable SA metric results. This means that even though small sets are able to ensure reliable R_i results, much larger error sets are necessary to get reliable SA results.

In addition to describing the process of taking into account the predictor variables, I performed additional in-depth controlled evaluations to determine the relationship between the relevant evaluation variables and evaluation metrics, to provide a good understanding of this relationship.

It was proved that the use of tokens is a better way of attaining reliable metrics and being representative of a user's experience for P_c. Although it was not explicitly proved for all of the metrics, it can be reasonably accepted that the same results can be expected from all of the other metrics tested in this study.

Based on current best practices implemented by evaluation teams such as TEMAA, the reliability of metrics when using high-frequency words was tested. It was found that although it is important to include high-frequency words in evaluations, these words are not ideal when trying to distinguish between different spelling checkers. It was shown that the biggest differentiation between spelling checkers can be made when the lowest cumulative 17% of a corpus is used as the evaluation material.

It was shown that the R_i metric is influenced by the number of incorrect instances used to calculate this metric. From the experiments it can be deduced that at least 70 errors need to form part of the evaluation for reliable R_i scores. Based on the fact that different researchers indicate minimum error levels of between 0.8% and 4%, a corpus of 15,000 words as required for R_c is sufficient for calculating R_i reliably. This would therefore provide at least 120 errors in a corpus of 15,000 words, which is enough to get reliable R_i scores.

In addition, it was proved that the percentage of errors in the text has a significant effect on the values of the P_i metric. It was found that the higher the percentage of errors in the data, the higher the P_i of the spelling checker is. The trend did not, however, account for variability in a set of experiments, as the variance in a set of similar experiments was significant at both lower and higher error percentages. With this in mind, two possible solutions for providing stable P_i metrics are suggested:

- error percentage should be normalised to a predetermined value when calculating P_i; or
- the modelling or validation unit should be seeded with additional errors in order for different evaluations to have the same percentage of errors in the text when comparative evaluations are done.

As a further investigation of current best evaluation practices, the difference between using word tokens and word types was also tested. It was found that metrics are more unstable, and therefore more unreliable, when using word types. The use of word tokens also reflects the experience of the end user better, and is therefore a more desirable way of measuring the performance of the spelling checker. This was specifically proved for R_c, but on the results of these evaluations, it could again be deduced that the same phenomena and results would occur for other metrics, namely R_i, P_i, P_c, and PA.

In addition to the discussed error detection metrics, another important aspect of spelling checker functionality is providing corrections for the detected errors. For this purpose there are several evaluation metrics that relate the number of times a spelling checker provides the expected correction, and in some instances takes the ordering of suggestions into account. In this part of the chapter I proved that although ordering metrics give slightly different results from a simple suggestion adequacy metric, the results are not statistically significant, and using the SSS is sufficient for measuring spelling checker suggestion adequacy.

Finally, overall linguistic adequacy measures were investigated, which reflected the extent to which the spelling checker fulfils the user's expectation and need. A number of possible metrics were described and tested for reliability, in order to determine which of the metrics would accurately describe the functionality of the spelling checker. Based on different arguments and inherent attributes of the metrics, it was shown that the PA metric is the best candidate for evaluating the overall linguistic adequacy of a spelling checker. In this calculation, a combination of the recognition of correct words and flagging of incorrect words, gives an overall impression of the spelling checker.

5. VALIDATING THE STEPWISE EVALUATION RELIABILITY METHODOLOGY

5.1 INTRODUCTION

The previous two chapters applied the proposed reliability evaluation methodology to three Afrikaans spelling checkers, with the aim of determining whether it is possible to establish a reliable evaluation using regression procedures. The results of this investigation showed that it is possible to model the reliability of an evaluation for different metrics by accounting for a small number of evaluation variables. It was further shown that the variables that influence the results of the metrics are different for different metrics because of the descriptive nature of those metrics. As each metric attempts to describe a particular part of the functionality of the technology, so the variables that the metric responds to are different for each metric.

Since the variance inherent in evaluations makes cross-system comparisons more difficult when the same data is not available, it was shown that it is possible to predict the variance of an evaluation based on the variables that influence particular metrics. This prediction is based on linear regression modelling, and the use of a created model to predict the variability that a specific evaluation is likely to exhibit prior to doing the evaluation. Providing the expected variability as part of the evaluation results also makes it possible for others to do comparable evaluation, and get a better sense of how reported evaluation results compare with their own and previous evaluations. This predictive quality also makes it possible, prior to doing the evaluation, to determine how reliable the results of the evaluation will be, and allows the annotator to adjust the evaluation in order to ensure reliable results.

In this chapter, the scope of the proposed methodology and linear regression modelling is extended to two additional technologies, and also to multiple languages for each of these technologies. The purpose of this additional application is to provide some evidence that the proposed methods are able to scale to a wider range of technologies, as well as different languages. For this purpose, POS taggers for Dutch, English, and Iberian Portuguese, as well as NE recognition for English and Spanish, are evaluated. With these validation procedures I aim to show that it is

possible to reliably predict evaluation variance for different technologies when accounting for the following set of evaluation variables:

- unit size;
- heterogeneity;
- target density;
- TTR;
- size of the classification set; and
- quality.

Please note that this list is not the same set of variables that was relevant for spelling checkers, since the two technologies that are evaluated here have different purposes and scopes. For each of the technologies, the information about the relevant variables for each evaluation metric is provided, as well as how the variable may influence the metric and thus the evaluation result. Furthermore, it is shown how not all variables are relevant to all technologies predicted by one or more statistical procedures that ensure accurate reporting of evaluation results.

5.2 VALIDATION: MULTIPLE TECHNOLOGIES AND LANGUAGES

In this section each step in the procedure is explicitly performed, as an example of how the problem space is approached and resolved with this new methodology. The aim of these experiments is to prove that the proposed method is implementable to additional technologies, as well as proving that the methodology is applicable to multiple languages without requiring language-specific processing or result adjustments. The models created in this chapter should also be useful for setting up future evaluations for these two technologies, as the models should remain applicable irrespective of the POS tagging or NE recognition implementation or the language that is evaluated.

There are two main reasons for selecting these two technologies. Firstly, POS taggers and NE recognisers are usually statistical classifiers, which is a different technological paradigm from error detection (spelling checkers). Secondly, the three technologies use different evaluation metrics to evaluate the attributes of the technology:

- Spelling checkers: recall correct, recall incorrect, precision incorrect, F-measures and predictive accuracy;
- POS taggers: accuracy;
- NE recognition: recall, precision, and F-measure.

Since it is not yet established whether or not the type of technology, language or metrics behaves differently when the proposed method is applied, this section investigates each of these assumptions in order to determine whether the methodology remains viable for all these conditions.

For both these technologies, I focus on doing usage-based evaluations to measure the functionality quality characteristic, and perform black box adequacy evaluations. This means that none of the underlying components are tested, but only the final output of the technology. These characteristics are tested on user data without any automatic procedures apart from comparing the results with the expected results which have been previously annotated as a gold standard. Since both technologies have the same scoping focus, detail about the first three scoping steps are not provided in the following sections for each technology.

There are two reasons for selecting different languages for the evaluations of NE recognition and POS tagging. Firstly, these experiments attempt to do a first validation of the evaluation methodology and the broader the scope of languages used for the validation the more likely it is that the methodology is applicable to a broad selection of languages. By selecting Germanic and Romance languages, it will be shown that the methodology is not dependent on a particular language group, but can be applied to a broader spectrum of languages. Secondly, the availability of data for evaluation purposes was an important consideration in language selection. In order to do validation experiments, large quantities of publicly available evaluation data are necessary to do all of the necessary experiments. The languages selected have large publicly available datasets which made it possible to do the validations without any additional annotation.

If the results from the different technologies, implementations, metrics and languages all provide comparable results to those found for spelling checkers, this will serve as a good first verification of the approach to creating evaluation models that predict evaluation reliability.

5.3 METHODOLOGY VALIDATION: PART OF SPEECH TAGGERS

The first set of evaluation validation experiments for verifying the methodology is carried out on POS taggers for Dutch, English, and Iberian Portuguese. The reason for selecting these languages is that they belong to different language families – Germanic and Romance – and also have large evaluation corpora available. For each language, a single maximum entropy POS tagger is used to calculate the metrics. Since the interest of this study is on the evaluation of the technology and not on the training of the taggers, I do not delve into the details of the individual taggers, but treat them as black boxes for the purpose of the evaluations.

5.3.1 STEP 4: COLLECT THE DATA

In order to evaluate the POS tagger for each of the languages, a set of data containing approximately 200,000 words for each of the three languages that formed part of the evaluation procedure validation was collected. The data used in this part of the process was completely separate from the data that was used to train the POS tagger models.

For the purposes of this validation process, the 160,000-word modelling corpus for each language was split into five smaller modelling sets differentiated by modelling unit size, where each set contained 10 modelling units for which the accuracy metric will be calculated. In each modelling set there was no sentence level overlap, and sentences were selected at random from the 160,000 set modelling corpus, to provide as much variation as possible in the data. This selection procedure means that there were 10 modelling units for each of the following modelling sets for each language:

- 1,000;
- 2,500;
- 5,000;
- 10,000;
- 15,000; and
- 20,000.

The division resulted in 60 modelling units per language, and all the data across all three languages was combined to do variance analysis and create the predictive regression model in the next step of the procedure. The choice of 10 sets per size for each language is based on the same principle as the well-established validation procedure in machine learning, of 10-fold cross-validation (Mitchell, 1997; Marsland, 2009). The main difference with the cross-validation procedure is that there is no separation between training and testing data, but rather separation between different evaluation runs to determine the variance between runs, and not just to get an average result for testing a particular training implementation on a limited data set.

It is also important to note that apart from controlling for size, none of the other variables are controlled for in the creation of the modelling units. In all other cases, there is explicit variability in the modelling units naturally occurring in the data and values for each of the variables ranges, as follows:

- modelling unit size: 1,000 to 20,000;
- size of tag set: 14 to 37;
- TTR: 0.23 to 0.54;
- quality: 60.88 to 90.97%

Figure 5.3-1-Figure 5.3-3 provide an overview of the results for each language that was tested in this evaluation.

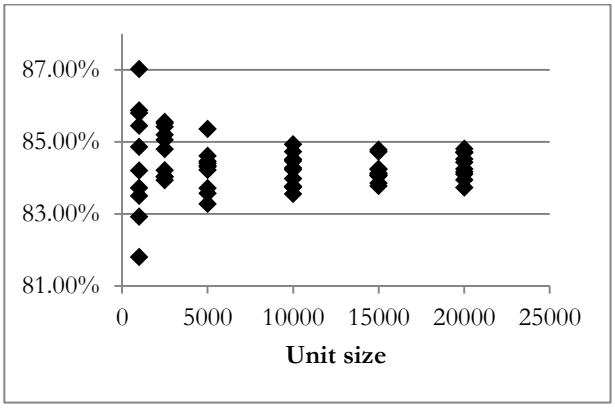

Figure 5.3-1 Dutch POS tagger precision on different modelling unit size

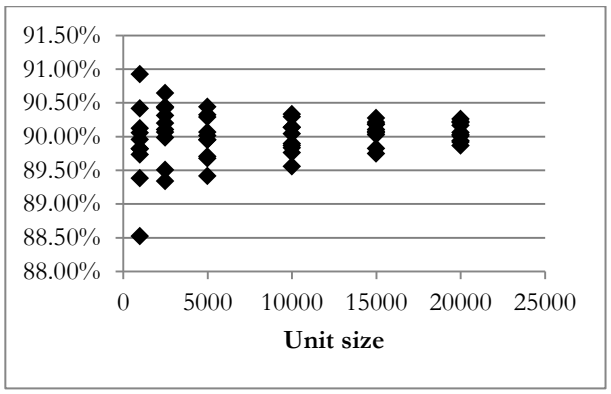

Figure 5.3-2: English POS tagger precision on different modelling unit sizes

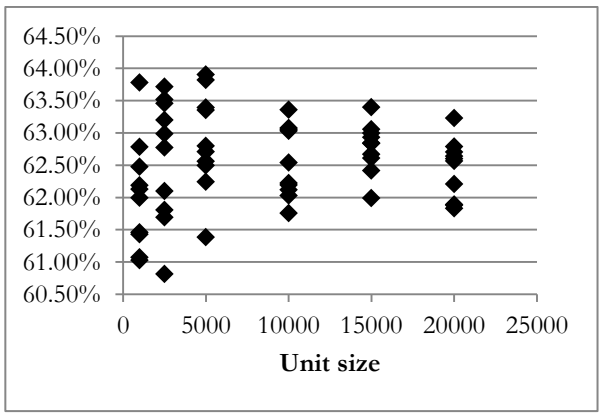

Figure 5.3-3 Iberian Portuguese POS tagger precision on different modelling unit sizes

The results of all of these evaluations are used in the next step to make the variable selection and calculate the model that accurately describes the variance within the modelling sets created in this step.

5.3.2 STEP 5: IDENTIFY POSSIBLE VARIABLES

As was indicated earlier in the chapter, each technology that is evaluated has a different set of inherent variables in the evaluation, depending on the type of technology and its target. For POS taggers, the target of the technology is every token in the input, with an expected result consisting of a single POS classification (tag) defined by the developer of the POS tagger. Since the POS tagger can be run on any input data of the target language, all the corpus-related variables could possibly influence the results of the POS tagger accuracy.

From the previous description, the following variables are possible sources of variability in the evaluation:

- **Size of the modelling unit**: This refers to the total number of tokens in the modelling unit. It is expected that larger modelling units have less variance in the evaluation results.
- **Size of the POS tag set**: This refers to the number of different POS tags the tagger can assign to any individual token. The larger the tag set, the more specific the tags for each individual word are, often containing more inflectional information in the tag than just the core POS. For example, the English standard Penn treebank tag set does not include only the core noun tag, but also the number associated with the noun (plural vs. singular). Depending on the intent of the tagger developer, the number of tags can vary greatly from as few as 12 POS tags (Petrov *et al.*, 2011) to more than 130 (Pilon, 2006). It is expected that the more fine-grained the tagger, the higher the variability in evaluation results will be.
- **TTR**: The TTR of the modelling unit can be a language-independent indicator of modelling unit homogeneity, as the lower this ratio, the fewer different tokens are used in the modelling unit, and the smaller the vocabulary in the modelling unit is. It is expected that the more homogeneous the modelling unit, the lower the variability of the evaluation results will be.
- **Quality of the tools**: One of the interesting results in the previous set of experiments is related to how the quality of tools affects the variability of evaluation results. The lower the quality of the tool, the higher the variance in the results of the evaluations will be when the same toolset is run on different validation units. This is the case because the best estimator of future error in the absence of other data is the current error. Since most metrics are indicators of error, using this as a variable in the process ensures that the error rate of the application is part of the calculation of expected variability. It is expected that the higher the quality of the tools, the lower the variance in the evaluation results will be.
- **Nature of the language**: Describing the complexity of language in terms of quantifiable parameters has been a contentious issue in the field of linguistics

(Crossley *et al.*, 2011; De Clercq *et al.*, 2013; Kate *et al.*, 2010). Assigning a complexity score to a language based on its morphologic or syntactic qualities, in addition to being complex, can also be construed as arbitrary. Having said that, there is little doubt that it is more difficult to create technologies for some languages because of their inherent qualities, and as such could influence the results reported by metrics. Comparing similar technologies across languages almost certainly requires an investigation of this inherent complexity. What may be an acceptable adequacy level for a particular language may be almost useless for another. The best-known measures of morphological complexity are the indexes of synthesis and fusion (Comrie, 1989; Pirkola, 2001; Puurula & Kurimo, 2007). The index of synthesis can be defined as the ratio of morphs per word form in the language, or the average number of morphemes per word (Pirkola 2001; Puurula & Kurimo, 2007). This ratio is indicated by a location on a scale with isolating languages at the one end of the spectrum and (poly)synthetic languages on the other. Every language falls somewhere on this scale

Isolating <——————————> Synthetic

The index of fusion can be divided into three indexes:

- the inflectional index of synthesis – the number of inflectional morphemes per the total number of words;
- derivational index of synthesis – the number of derivational morphemes per the total number of words; and
- compound index of synthesis – the number of compound components per the total number of words.

The more synthetic a language, the more distinct word forms appear, while there are fewer distinct word forms in isolating languages.

The index of fusion refers to the ratio of morphs per morpheme, or as Pirkola (2001) puts it, the ease with which morphemes can be separated from other morphemes in a word. This is again represented by a scale with agglutinative languages on the one end and fusional languages on the other. Except for isolating languages which have no agglutination or fusion, all languages fall somewhere on this scale.

Agglutinative <————————————> Fusional

Fusion refers to words where the end product is something other than the sum of the components. Strong verbs in Germanic languages, for instance, are a case in point, where a form such as "*took*" is monomorphemic, but denotes two distinct semantic instances – "*to take*" and "*past tense*" (Pirkola, 2001; Puurula & Kurimo, 2007). It is believed that the nature of a language has a significant effect on the results of an evaluation, and is included as one of the independent variables in the modelling procedure.

Target density is not considered as a valid independent variable for the POS tagger evaluations because each token in the modelling unit is a valid target for the tagger, and therefore all modelling units have a target ratio of 1:1. Since the size of the evaluation set is included, this information is already encoded as a variable, and there is no need to include target density as a variable.

The five independent variables identified here are used to attempt to create an evaluation model for POS taggers that can accurately predict the variance of future evaluations, given a particular set of values for these four variables. The modelling procedure for these five variables is completed in Steps 8 and 9.

5.3.3 STEP 6: IDENTIFY RELEVANT METRICS

The evaluation metric most widely used for POS tagger evaluation is the accuracy metric (Daelemans *et al.*, 1996:19; Diab *et al.*, 2004; Paroubek, 2007; Pilon, 2006). This metric is defined by comparing the relation of expected tags with the tag predicted by the POS tagger. This metric calculates the percentage of tokens that are assigned the correct POS tag, and is calculated as follows:

$$Accuracy = \frac{\#Correct\ POS\ assignments}{Total\ number\ of\ tokens} \qquad (5.3\text{-}1)$$

Since every unit in the evaluation corpus is a target for POS assignment, there is no notion of recall, and accuracy is the only metric that is of interest here. The following sections provide all the information regarding each step in the evaluation procedure as an explicit example of calculating and generating the relevant regression model for predicting future evaluation reliability.

5.3.4 STEP 7: VALIDATE THE METRICS

Since the accuracy metric used in this validation experiment is very widely used and accepted, I will not do the validation of the metric, and will rather focus on the validation of the regression modelling.

5.3.5 STEP 8: SELECT INDEPENDENT VARIABLES

This step in the evaluation verification methodology attempts to identify the variables that accurately predict the variability in different evaluations for each metric used for a particular technology. Since there is only a single evaluation metric for POS taggers, it is only necessary to do this procedure once. This procedure uses one of the standard variable selection and reduction techniques widely used in statistical analysis, namely FSRP (Neter *et al.*, 1996:332). The aim here is to determine whether any of the variables are correlated with the response variable.

Using FSRP (Neter *et al.*, 1996:346) the modelling shows that keeping the following three variables is required to accurately predict the variance in future evaluations:

- number of words;
- number of tags; and
- TTR.

The following table shows the beta and residual results of the selection algorithm, where the beta values are the weight beta values for each of the variables that should be included in the model.

Constant = 0.008; Std. Error of estimate = 0.002

Variable	Beta	t
Modelling unit size	-0.006	-17.207
Number of tags	-0.00016	-13.873
TTR	0.01128	8.951

Table 54: Final step result of FSRP for POS tagger precision

Based on these results, the proposed model for predicting the CI of a new validation unit is:

$$Accuracy_i = 0.008 + (-0.006 * x_{i1}) + (-0.00016 * x_{i2}) + (0.01128 * x_{i2}) + 0.002 \quad (5.3\text{-}2)$$

where:

- x_{ij} are the values of the independent variables of the *i*th trial
  - x_{i1} is the modelling unit size;
  - x_{i2} is the number of tags; and
  - x_{i3} is the TTR.

For future evaluation of POS taggers, solving this equation gives the predicted CI for that particular evaluation, given the set of independent variables. By extension, knowing the predicted CI makes it possible to determine what the range of variability for the particular evaluation is, where the higher this value, the higher the variability in the evaluation is, and the less reliable the evaluation is.

As an example, a POS tagging evaluation with 3,500 words/25 tags/0.28 TTR creates an evaluation with a predicted CI of ±0.008184 or ±0.8184%. This means that 95% of evaluation with similar inherent variables falls within a ±1.63% range around this evaluation, and means that the current evaluation is not reliable according to the definition in 2.2.

5.3.6 STEP 9: SELECT AND VERIFY THE RELIABILITY MODEL

The next step in our analysis of the data is verifying the results and the model created in the previous step. This step validates the model by creating 10 randomly selected validation units from the validation corpus without controlling for any of the variables. For each of these validation units, the model predicts what the variance of the evaluation is, given the values of the variables inherent in the modelling unit. Since the predicted CI provides a range around the mean within which future evaluations should fall, any deviation that falls within this range is a confirmation of the model.

Since the evaluations with the lowest variance in precision are the evaluations with 20,000 words (cf. Figure 5.3-1 - Figure 5.3-3), the mean of those experiments as the mean to compare variance in the other sets, is used. For each language, the means on the 20,000-word experiments were as follows:

Language	Mean
Dutch	84.33%
English	90.07%
Iberian Portuguese	62.43%

Table 55: Mean accuracy of POS taggers for three languages

The following table gives a summary of the independent variables as well as the predicted deviation of precision for the three different languages tested. In each case the actual error from the mean is reported, and should fall within the ± range of the predicted deviation. All the actual errors fall in the predicted CI and thus validate the model's predictive capability.

Language	Unit size	Number of tags	TTR	Predicted 95% CI	Actual error
Dutch	0.117300	14	0.266800	±1.01%	-0.88%
Dutch	0.243350	14	0.227300	±0.89%	-0.07%
Dutch	0.567900	14	0.216500	±0.68%	0.57%
English	0.056150	37	0.506000	±0.95%	0.11%
English	0.238450	37	0.272600	±0.57%	0.11%
English	0.839200	37	0.437700	±0.40%	-0.11%
Iberian Portuguese	0.054750	18	0.514400	±1.26%	0.96%
Iberian Portuguese	0.326600	18	0.318600	±0.88%	0.78%
Iberian Portuguese	0.662950	18	0.443900	±0.81%	0.19%

Table 56: Predicted CI and actual error values for POS taggers for three languages

Although this experiment on POS taggers provides support for the modelling technique proposed here, applying the modelling to more technologies adds additional support for the claim that the proposed methodology has merit and accurate predictive capability. In the next section, the same procedure and validation are applied to NE recognition to further verify methodology.

5.4 METHODOLOGY VALIDATION: NAMED ENTITY RECOGNISERS

The second set of validation experiments for the proposed methodology is performed on three different NE recognition implementations for two different languages. This means that six NE recognition systems are used to create and

validate the predictive model. As with POS taggers, this study is not concerned with the specifics of the different implementations, but only the evaluation results and variability in these results. All six systems used in this evaluation procedure are systems that were entered into the CoNLL's shared task in 2002 (Spanish) and 2003 (English).

From the systems participating in the shared task, I selected the systems that performed the best, the worst, and one randomly selected other system. Once again, each of the systems was treated as a black box implementation, where only the resultant labels were considered for the purposes of creating the evaluation variability modelling.

5.4.1 STEP 4: COLLECT THE DATA

Although the data was not specifically selected for the purpose of this study, the data available from the CoNLL shared task (CoNLL: 2002, 2003) has enough evaluation data to perform the required modelling and validation steps. For each language, 100,000 words of annotated data are available, where each named entity is marked with the expected named entity tag against which the tool outputs can be compared.

Much like the process for POS taggers, the modelling corpus is split into smaller modelling sets consisting of between 1,000 and 20,000 words. Each modelling set is randomly selected from the full validation corpus to ensure no overlap between the different modelling sets. For each modelling set, 10 modelling units are created by random selection of sentences from the modelling corpus. Each of the corpora is used to calculate precision, recall, and F-measure results, to determine what the variability between the different modelling units is, and which variables identified in the previous step will be relevant to the reliability of the evaluation procedure for each of the metrics.

5.4.2 STEP 5: IDENTIFY POSSIBLE VARIABLES

As before, the first step in the evaluation validation procedure is identifying those variables that are likely to have an effect on the reliability of the evaluations and metric results for a particular technology. Unlike POS tagging and spelling correction,

NE recognition does not target every token in the modelling corpus, because most tokens in the modelling corpus are not named entities, and therefore are not targets of the NE recognition system. With this in mind, the following inherent variables are likely independent variables in the evaluation process that could influence the reliability of the evaluation.

- **Modelling unit size:** this once more refers to the number of tokens in the modelling unit. It is expected that the larger this set, the more reliable the evaluation results will be for all three evaluation metrics.
- **Size of the NE recognition tag set:** Similarly to POS tagging, there are different classes of named entities, and the distinction that is made between classes and sub-classes can have an effect on the reliability of the precision and F-measure metrics. It is expected that smaller tag sets will lead to more reliable evaluation results. The reason why the size of the tag set will not have an effect on the recall metric is that this particular metric is only concerned with identifying named entities, and not with the particular classification of that entry.
- **TTR:** The TTR is an indicator of lexical variability which may affect all three the evaluation metrics, as larger lexical variability may uncover defects in a particular system's approach, which in turn may lead to increased variability in the evaluation results reported for each of the metrics. It is expected that a lower TTR will result in more reliable evaluation results.
- **Target density:** As has been pointed out before, NE recognisers does not target all tokens in the modelling unit, but only a subset of items that are positive examples of named entities. Similarly to precision incorrect for spelling checkers, the percentage of tokens in the modelling unit that are positive targets is likely to influence the reliability of the evaluations. Specifically, higher target density is likely to result in more reliable evaluation results. To calculate target density, we divide the number of tokens marked as named entities in the unit by the number of words in the unit.
- **Quality:** Lastly, a possible source of variability in the results is the quality of the tools that are being tested. It is hypothesised here that systems that have higher precision or recall values are more likely to provide consistent

evaluation results, which by definition means that the evaluations are more reliable.

The effect of these evaluation variables is tested and verified in the following sections, to determine their relative effect on the results of individual variables. Depending on the regression procedure results, a model for NE recognition reliability is created, that takes into account those evaluation variables that have a correlated relationship with the variance in the performed evaluations.

5.4.3 STEP 6: IDENTIFY RELEVANT METRICS

Unlike POS tagging, which had a single metric for calculating the quality of the technology, NE recognition evaluation uses three different metrics to determine the quality of different aspects of the NE recognition system. The first metric is the precision metric, which calculates the rate of correctness for all tokens identified as named entities – in other words, if the system indicates that a particular token is a named entity, how often is the system correct? The higher the precision value, the more accurately the system labels named entities. This metric is expressed as a fractional value between 0 and 1 by completing the following equation:

$$Precision = \frac{\#Correct\ NEs\ assigned}{Total\ \#\ NEs\ assigned} \qquad (5.4\text{-}1)$$

The second metric used for NE recognition evaluation is the recall metric. This metric calculates the detection rate at which named entities are identified by the system – in other words, what percentage of named entities in the modelling unit are recognised as named entities by the NE recogniser? This metric also calculates a fractional value between 0 and 1, where higher values indicate better recognition of named entities, by completing the following equation:

$$Recall = \frac{\#Correct\ NEs\ assigned}{Total\ \#\ NEs\ in\ the\ evaluation\ set} \qquad (5.4\text{-}2)$$

The last metric used to evaluate NE recognition systems is the F-measure. This is a combinatory metric that reports the harmonic mean between precision and recall, where the value is weighted towards the worse of the two contributing metrics. The reason for using this metric is that it penalises systems that favour one of the two important aspects of the task too heavily. As an example, creating a system that tags

every token as a named entity, the system would get 100% recall, but very low precision, say 10% for the sake of this example. The mean of those two results would be 55%, while the harmonic mean would weigh more heavily in favour of the low precision, and would report a result of 18.18%. This metric provides a way to enforce a balanced approach to feature development, and ensures that a completely exclusionary or completely inclusionary system cannot report better results than a balanced system. The metric calculated also reports a fractional value between 0 and 1, and is calculated as follows:

$$F = 2/\left(\frac{1}{Precision} + \frac{1}{Recall}\right) \qquad (5.4\text{-}3)$$

Since there are three evaluation metrics, the next sections follow a similar procedure and structure to the previous section, with the exception that the model generation is done for each metric separately. Doing the modelling separately is necessary because each metric may be influenced by a different set of variables, and the validation for each metric is different, depending on the metrics that are relevant for that metric. In the following sections the experiments, modelling and validation results are reported, including explicit steps and rationalisations.

5.4.4 STEP 7: VALIDATE THE METRICS

Since all of the metrics used in this validation experiment are widely used and accepted, I will not do the validation of the metrics, and will rather focus on the validation of the regression modelling.

5.4.5 STEP 8: SELECT INDEPENDENT VARIABLES

As before, this step in the methodology attempts to find those evaluation variables that have a statistically significant relation to particular evaluation metrics. Unlike the POS tagger evaluations discussed in the previous section, the procedure of identifying the variables is done three times for each of the evaluation metrics described in 5.4.3. This is necessary because there may be different variables that influence the variability of a particular evaluation metric, because the different metrics describe different target functionalities.

From the results of FSRP, only three of the target variables have an effect on the metrics:

- unit size;
- number of tags; and
- quality.

It is a bit surprising that neither target density nor TTR appears to be a good descriptor variable for any of the metrics, especially the precision metric. However, on further investigation, target density is not a good descriptor of the expected variability found in the evaluations, but it is a good descriptor of what the precision for a particular run will be. Since the interest here is in describing and minimising variability, predicting the precision value for a particular evaluation is not of interest at the moment.

Recall results:

Constant = 0.036; Std. Error of estimate = 0.006

Variable	Beta	Std. Error
Unit size	-0.022	0.001
# Tags	0.005	0.000
Quality	-0.038	0.007

Table 57: Final step results of FSRP for NE recognition recall

Precision results:

Constant = 0.074; Std. Error of estimate = 0.008

Variable	Beta	Std. Error
Unit size	-0.032	0.001
# Tags	0.004	0.000
Quality	-0.071	0.011

Table 58: Final step results of FSRP for NE recognition precision

F-measure results:

Constant = 0.052; Std. Error of estimate = 0.006

Variable	Beta	Std. Error
Unit size	-0.024	0.001
# Tags	0.004	0.000
Quality	-0.052	0.010

Table 59: Final step results of FSRP for NE recognition F-measure

Based on the results in Tables 57, 58 and 59, the following models are the three created models based on the modelling corpus across the six systems created in this step of the methodology.

$$Recall_i = 0.036 + (-0.022 * x_{i1}) + (0.005 * x_{i2}) + (-0.038 * x_{i3}) + 0.006 \quad (5.4\text{-}4)$$

$$Precision_i = 0.074 + (-0.032 * x_{i1}) + (0.004 * x_{i2}) + (-0.071 * x_{i3}) + 0.008 \quad (5.4\text{-}5)$$

$$FMeasure_i = 0.052 + (-0.024 * x_{i1}) + (0.004 * x_{i2}) + (-0.052 * x_{i3}) + 0.006 \quad (5.4\text{-}6)$$

where:

- x_{ij} are the values of the independent variables of the *i*th trial:
  - x_{i1} is the number of words;
  - x_{i2} is the number of tags; and
  - x_{i3} is the quality of the tools.

These models for each of the metrics can predict the expected variability of the evaluation, given values for each of the independent variables. In the next section, the validity of the proposed models is evaluated when applied to previously unseen validation units.

5.4.6 STEP 9: SELECT AND VERIFY THE RELIABILITY MODEL

As before, the next step in the procedure is to verify the models by testing their reliability when attempting to predict the variability of evaluation for each of the

metrics given the variables identified in the previous step. For this purpose, the mean is calculated for each system in each language on the 10 modelling units containing 20,000 words because the variability across these sets is the lowest in the initial set of evaluations.

	System 1	**System 2**	**System 3**
Recall:			
English:	97.41%	94.39%	96.76%
Spanish	94.80%	94.37%	81.68%
Precision:			
English:	88.94%	86.55%	85.23%
Spanish:	78.55%	87.75%	82.87%
F-Measure:			
English:	92.98%	90.30%	90.63%
Spanish:	85.91%	90.94%	82.27%

Table 60: Recall, precision and F-measure results for three different NE recognition systems on both English and Spanish

In the experiments validating the results presented in Table 60, I have randomly selected an English and Spanish set for each language from which the known independent variables are used, to predict the CI for the given validation units for each system that is to be evaluated. This predicted variability is compared with the actual variance, and the validity of the model is validated if the actual error falls within the range of variability predicted by the models.

The selected validation units have the following set of independent variables:

	English	**Spanish**
Quality	<System Specific>	<System Specific>
Number of tags	4	6
Unit size	4171	7305

Table 61: Variables for English and Spanish NE recognition validation

The results in Table 62 indicate that for all the systems across both English and Spanish, the variation in the evaluations falls within the predicted variability range from the models created for the three metrics respectively. This fact reinforces the validity of the models, and provides additional proof for the modelling approach of predicting the variability in a validation unit when the relevant independent variables in the evaluation are taken into account.

One additional point to consider when looking at the results in Table 62 is the fact that the model is much less accurate when predicting English than Spanish. The reason for this is that there is more variability in the Spanish modelling unit results, and consequently, in order to include 95% of all the results, the models predict a larger CI than may be strictly necessary if the models were only constructed on the English data. Models that attempt to predict confidence intervals across multiple systems and languages with a lot of variation in the results, are less accurate than models constructed on less variable results.

	System 1		**System 2**		**System 3**	
	Predicted 95% CI	**Actual error**	**Predicted 95% CI**	**Actual error**	**Predicted 95% CI**	**Actual error**
Recall						
English:	±2.04%	0.12%	±2.15%	0.57%	±2.06%	-0.62%
Spanish	±2.79%	1.91%	±2.81%	2.14%	±3.29%	2.96%
Precision:						
English:	±2.82%	0.42%	±2.99%	0.69%	±3.08%	-1.40%
Spanish:	±3.85%	-2.07%	±3.20%	-0.96%	±3.55%	0.20%
F-measure:						
English:	±2.06%	0.42%	±2.20%	0.69%	±2.19%	-1.40%
Spanish:	±2.86%	-0.33%	±2.59%	0.36%	±3.05%	1.68%

Table 62: Predicted CI and actual error for recall, precision and F-measure for three different NE recognition systems on English and Spanish validation units

It must be emphasised at this point that although all the experiments that have been done here fall neatly within the bounds of the predicted variability, the models are only predicting the variability at 95%, and it is expected that about 5% of evaluations will show more variability than the predicted variability. However, applying this methodology does provide a usable framework for performing similar predictive experiments to expand and improve the methodology.

5.5 CONCLUSION

This chapter focused on verifying the methodology and regression procedures proposed in the previous chapters, by applying the evaluation reliability methodology to an additional two NLP technologies across four different languages. The results

produced in this chapter explicitly showed that the identification of variables inherent in the evaluations are valid predictors of variability when applied to different technologies.

In this chapter this methodology was applied to POS tagging for Dutch, English, and Iberian Portuguese, with the aim of: a) identifying relevant variables; b) creating regression models to predict future evaluation variability; and c) validate the models by testing them on previously unseen validation units. The same was done for NE recognition for English and Spanish, with the same aims.

The results of the POS tagging showed that for the accuracy metric, three variables were relevant predictors of variability, namely:

- unit size;
- number of tags; and
- TTR.

These three predictor variables were selected using FSRP, and produced the following regression model that predicts the CI of POS tagger accuracy:

$$\text{Accuracy}_i = 0.008 + (-0.006 * x_{i1}) + (-0.00016 * x_{i2}) + (0.01128 * x_{i2}) + 0.002 \quad (5.5\text{-}1)$$

where:

- x_{ij} are the values of the independent variables of the *i*th trial:
  - x_{i1} is unit size;
  - x_{i2} is the number of tags; and
  - x_{i3} is the TTR.

Similarly, the experiments on NE recognition showed that three variables:

- unit size;
- number of tags; and
- quality of the tools.

can be used to predict each of the three evaluation metrics produced for NE recognition. The fact that the three variables are identical across all three metrics is purely coincidental, and does not mean that other technologies that have multiple

evaluation metrics should assume that identifying the variables for a single metric means that no other variables will be of interest for different metrics.

$$Recall_i = 0.036 + (-0.022 * x_{i1}) + (0.005 * x_{i2}) + (-0.038 * x_{i3}) + 0.006 \quad (5.5\text{-}2)$$

$$Precision_i = 0.074 + (-0.032 * x_{i1}) + (0.004 * x_{i2}) + (-0.071 * x_{i3}) + 0.008 \quad (5.5\text{-}3)$$

$$FMeasure_i = 0.052 + (-0.024 * x_{i1}) + (0.004 * x_{i2}) + (-0.052 * x_{i3}) + 0.006 \quad (5.5\text{-}4)$$

where:

- x_{ij} are the values of the independent variables of the *i*th trial:
  - x_{i1} is the number of words;
  - x_{i2} is the number of tags; and
  - x_{i3} is the quality of the tools.

From these results it can deduced that evaluation variability is not only a function of the language being evaluated, but also of the combination of variables such as the nature of the language, the evaluation methodology, and the evaluation corpus. In addition, it was shown that the set of variables that are relevant is not specific to a particular technology or even metric, but that different variables affect variability differently for different technologies.

Even though the variables were different for different technologies and metrics, it does appear that the models that are created for a particular technology are applicable to all instances of that technology, irrespective of the quality of the technology or of the language, as long as the variables associated with those metrics and technologies are taken into account in the created model. This does, however, mean that this procedure must be run at least once for each technology, to identify relevant variables and create models that can predict the variability for future evaluation based on those variables.

Although there was no evidence in the results of the evaluations done in this chapter, the modelling may be language or language-family related, and some models may require language-specific models to be created in order to accurately generate these

predictions. This assertion needs to be evaluated in future work across a wider spectrum of technologies and languages, especially those languages not covered in this thesis such as Uralic, Asiatic and agglutinative languages, as well as more complex technologies such as machine translation and language generation technologies.

6. CONCLUSION

6.1 SUMMARY

As the field of NLP has developed over the last two decades, producing accurate and reliable evaluation results has become more important. Even though the adequacy evaluation of these applications was introduced during the 1960s, little work was done on the structuring and standardisation of evaluations until the 1990s. Over the last two decades, an extensive body of work on the evaluation of NLP application has been done, with the aim of establishing a standardised set of evaluation criteria and benchmarks for different NLP applications.

Even though these evaluations have become an integral part of how progress in the field is measured, there is still uncertainty about the accuracy and reliability of the results that are reported at conferences and in journal papers. It is often easy to show how one evaluation favours a particular technological approach, while the same approach performs worse when evaluated a second or third time, sometimes even in the same publication. With this in mind, the goals of this study were to determine what the source of this variability in evaluation results is, and if it is possible to model this variability through linear regression procedures that are adaptable to the technology being evaluated and the evaluation methodology being used.

The first chapter gave a brief overview of the nature of NLP evaluations and the efforts to standardise evaluation methodologies and metrics. This discussion focused on the variability that is still observed in NLP evaluations. Based on the overview of NLP evaluations, the following objectives were set for this study:

a. To explore and define reliability for NLP evaluations.
b. To design a stepwise methodology for modelling evaluation reliability with linear regression procedures.
c. To provide a structured method for identifying variables that are predictors of evaluation reliability.
d. To provide a detailed investigation of how variables affect evaluation reliability.

e. To validate the stepwise evaluation reliability methodology by applying it to multiple NLP technologies and languages.

Chapter 2 started with an overview and definition of evaluation reliability, in order to establish the focus of the study. This overview of reliability culminated in the following definition of evaluation reliability, based on Gaussian distributions:

> *An evaluation metric is reliable when 95% of future evaluations are within a ±1% interval range from the reported result.*

Following the discussion on reliability, the second part of the chapter provided the framework for a new evaluation methodology that allows evaluators to determine an evaluation's reliability statistically. This proposed methodology consists of two phases, the scoping and modelling phases, totalling nine steps. The methodology is a combination of existing evaluation procedures such as those proposed by EAGLES (1995a; 1995b), TEMAA (1997), SCARRIE (1998) and statistical approaches to validation and reliability prediction that have not been used in evaluation procedures before. Each individual step in the procedure was described in detail, along with the principles for each of the steps, to make the exact nature and purpose of each step clear. The final two steps in the process focused on the statistical reliability prediction of evaluations that have not previously been used in evaluation procedures at all. The steps described the use of FSRP for selecting evaluation variables and a prediction model for predicting the expected variability of the evaluation based on the variables in the evaluation. A detailed case study of the stepwise evaluation reliability methodology was done in Chapters 3 and 4 by applying the methodology to spelling checker evaluations.

Chapter 3 started the application of the stepwise evaluation reliability methodology on a single well-defined and understood technology, namely spelling checkers. This chapter focused on the implementation of the scoping phase of the evaluation methodology by applying the following steps from the methodology:

1. Define the purpose.
2. Define the type.
3. Define the method.
4. Collect the data.

5. Identify possible variables.
6. Identify relevant variables.

For the purposes of this study, the implementation of the methodology focused on a black box adequacy evaluation that used a combination of automatic and usage-based evaluation methods. This case study focused only on using gold-standard evaluations and measuring the functionality attribute (EAGLES, 1995a; King, 1998) using a balanced evaluation corpus based on the stratification of the BNC. This corpus was split into a modelling corpus and a validation corpus, which was used to produce a model from the former and validate the model on the latter corpus.

The fifth step in the methodology was the identification of independent evaluation variables, that is, the set of variables that were possible predictors of the expected variability in the evaluation. Two types of variables were identified for spelling checkers, namely methodological variables and data attribute variables. It is only possible to statistically model data attribute variables, but it is also important to discuss and demonstrate the effect of methodological variables, and this was explored in more detail during the modelling phase of the methodology in Chapter 4. The first column of Table 63 provides the full set of data attribute variables identified for the evaluation of spelling checkers.

The final step in the first phase was the identification of relevant evaluation metrics. Evaluation metrics are the measurement instruments used to determine the quality of the spelling checker for a particular functional attribute. For the evaluation of spelling checkers, the following set of metrics was used, based on the evaluations done by Riley *et al.* (2004), SCARRIE (1998), Starlander & Popescu-Belis (2002), TEMAA (1997), and Van Huyssteen *et al.* (2004). These metrics were:

- recall correct (R_c);
- recall incorrect (R_i);
- precision correct (P_c);
- precision incorrect (P_i);
- suggestion adequacy (SA);
- predicative accuracy (PA);

- mean time between failures (MTBF);
- F-measure correct (fm_c); and
- F-measure incorrect (fm_i).

During the application of the second phase of the methodology in Chapter 4, each of these metrics was discussed and validated in detail, to determine their descriptive competency.

Chapter 4 continued the case study of the stepwise evaluation reliability methodology, by performing the modelling phase for spelling checkers. The chapter provided detailed descriptions of each metric used in the evaluation of spelling checkers, as well as the variables that could possibly affect each of these metrics. Using FSRP, the relevant independent (or predictor) variables were identified and combined into regression models that would be able to predict the variability, in the form of the expected variance, of future evaluations, given the set of variables.

	R_c	**R_i**	**P_i**	**P_c**	**PA**	**fm_i**	**fm_c**	**FIPP**	**SA**
Modelling unit size	x	x	x	x	x	x	x	x	x
Number of errors		x	x		x	x	x	x	x
TTR	x	x		x	x	x	x	x	x
Error TTR		x	x		x	x	x	x	x
Quality	x	x	x	x	x	x	x	x	x
Target density			x	x	x	x	x	x	x
Error types									x
Word frequency	x				x	x	x	x	
Types or tokens	x	x	x	x					x
Usage-based or automatically generated data		x	x						x

Table 63: Matrix of possible independent variables for various spelling checker metrics

As part of this regression procedure, it was found that different metrics were influenced by different variables, and that no single model would be able to predict the reliability of an evaluation for all metrics, but that different combinations of variables were applicable to different metrics. This followed from the fact that different variables described different behavioural attributes of the NLP technology.

Based on the results of the initial variable identification, more in-depth investigations of the relationship between individual metrics and variables were done, using sets of controlled experiments to explicitly show the influence of variables on the results produced by evaluations. Some of the experiments also showed how different methodological implementations affected the evaluation results, and how this should be figured into evaluation strategies.

One of the goals of this study was to show that the proposed evaluation methodology had wider application than just spelling checker evaluations, and that the modelling could also be done across multiple languages. In Chapter 5 the applicability of the stepwise evaluation reliability methodology was validated by applying it to two additional technologies, POS tagging and NE recognition, across multiple languages. These two technologies had different inherent functionality to that of error correction, in that they attempted to categorise words through classification, and also had different metrics and inherent variables from the error detection evaluations.

The first part of the chapter focused on the validation of the methodology on POS taggers for Dutch, English and Iberian Portuguese. By following the methodology proposed by the study, I showed that it is possible to create a model that accurately predicts the 95% CI of the evaluation, by accounting for the unit size, number of categories, and TTR. Similarly, the second part of the chapter applies the methodology to NE recognition for English and Spanish on three different systems for each language. This application is slightly different because the same language was used multiple times in both the modelling and validation phases, but the results still showed that it is possible to predict the CI based on a subset of the available variables, namely unit size, number of tags, and quality of the technology.

The results of these additional validation cases showed that it is possible to identify evaluation variables for different technologies and different metrics, by applying the same procedures used in the previous chapters. This validated both the procedure and the results produced by the evaluation procedure, and allowed future evaluations to apply and test the evaluation reliability framework to additional metrics and NLP technologies.

6.2 CONTRIBUTIONS

In this study it was shown that even though evaluations form a very important part of the NLP research field, there are still important unanswered questions about the reliability of reported evaluation results. In order to establish reliability more consistently, a new stepwise evaluation reliability methodology is proposed that standardises the statistical validation of evaluations by accounting for inherent evaluation variables. This methodology makes it possible for future evaluators to accurately predict the expected variability of their evaluations, and to adjust the evaluations to account for the variables in a way that ensures reliable evaluation results.

The importance of the new methodology relates specifically to three problematic areas of evaluations. Firstly, once a model for a particular technology and metric has been established, it is possible to predict the variability of an evaluation prior to doing the evaluation, and evaluations can be constructed in a way that guarantees reliable evaluations. Furthermore, the scope and amount of annotated data can be minimised, allowing for more data to be used as training data, thus allowing for better planning and minimising the cost of evaluations.

Secondly, the methodology makes it possible for different researchers working on different data sets to compare evaluation results more systematically without the need for working on shared evaluation sets. This is of particular value in fields of NLP, where proprietary information is used in the modelling and evaluation of technologies. This means it is difficult, if not impossible, to reliably compare evaluation results without a viable way of establishing evaluation reliability.

In addition to the proposed methodology, I also provide a first set of reliability prediction models for spelling checker, POS tagging, and NE recognition technologies. Each system and each metric model allows future researchers to predict the reliability of their evaluations based on the variables in the evaluation data they have available, without having to follow the entire stepwise evaluation reliability methodology.

6.3 FUTURE WORK

Based on the results of this study, various areas of research on evaluation reliability, and specifically the stepwise evaluation reliability methodology, need to be further validated and investigated. The first area that requires more investigation is further validation of the proposed regression models for spelling checkers, POS taggers, and NE recognition systems. In all three cases, the methodology was applied across multiple systems and languages, but the applicability to languages from other language families, and with different classification scope, may show improvements that can be made to either the models or the methodology.

A second topic of interest is determining whether there are more accurate statistical methods for generating the prediction models. As with most statistical analysis, there is often more than one way to approach and solve a problem. Although FSRP was used in this methodology for variable and model selection, there may be other statistical modelling and variable selection approaches that create more accurate prediction models.

Following from the variable selection process, there may be other variables that are not accounted for in this study. Two possible evaluation variables that were not specifically included here are the complexity of the text using readability scores, and the complexity of the language. It should be investigated whether additional attributes are necessary to improve the accuracy of the predictions made by the current evaluation models. It would also be of interest to determine if there is a set of global variables that are applicable and relevant to all technologies, or at least all technologies with a similar focus.

Lastly, the methodology should be applied to more NLP technologies, to establish the variables that are relevant to other technologies, and to create initial reliability models for those technologies to help other researchers be able to predict their evaluation reliability and expand the set of variables that are accounted for in evaluations. This is especially relevant to more complex technologies where more variables may be required to accurately predict evaluation reliability.

ADDENDUM A.

Statistical approaches to data interpretation[18]

Generally, two types of methodological surveys for statistical data interpretation are distinguished, namely complete and sample surveys. Complete surveys determine the value of each object in the entire population, while sample surveys are carried out on a group of objects from the entire population, known as the sample. As the objects in spelling checker evaluations are correctly and incorrectly spelled words from an infinite set of words for a given language, it would be impossible to do a complete survey which would give results that are 100% accurate. The evaluation is therefore always done on a sample of the entire language.

The drawback of this approach is that the results and conclusions are and always will be hypothetical, and deductions can only be made with a certain amount of confidence. However, this is the only way of determining approximations of the actual attribute value. The statistical inference in this study is based on the statistical presupposition that estimations about the entire population can be made after testing a sample of the population. For spelling checker evaluations, this can be interpreted as evaluating a sample (i.e. the modelling or validation unit) of the population (i.e. the entire standard written variant). Based on the sample, conclusions about the linguistic efficiency of a spelling checker could then be made. In the following sections, overviews of the different concepts needed to do the inference are given, and all are used in the study.

Distributions

Statistical inference involves using data from a sample to generalise findings to the entire population. By finding the distribution of a sample, generalisations about the entire population can be made, based on particular statistical principles. The distribution of a sample can be approximated by constructing a histogram, which is a pictorial representation of a frequency distribution. Different distributions have different inherent qualities, and once the distribution of a given sample has been

[18] The descriptions in this chapter are introductory in nature, and full reference to relevant authors are not provided. The chapter is based on introductory statistical reference works by the following authors: Devore & Peck (1986); Field (2005); Neter *et al.* (1996).

attained, different inferential principles can be applied to the sample data, based on the distribution of the sample data. Although various other distributions exist, this discussion of distributions is limited to the normal distribution, as the data sets in this study produce normal distributions.

Since the variables in the evaluation of spelling checkers are always continuous variables (i.e. the value of the variable comprises the entire interval on the number line) this means that the value of the metric can take any value between 0 and 1. The probability distribution of a continuous random variable is specified by a density curve such that the total area under the curve is 1. The probability of a value falling in any particular interval is the area under the density curve and above the interval. Thus the probability of a variable x falling between two values a and b can be denoted by $a < x < b$, and is represented by the area between a and b in Figure 6.3-1.

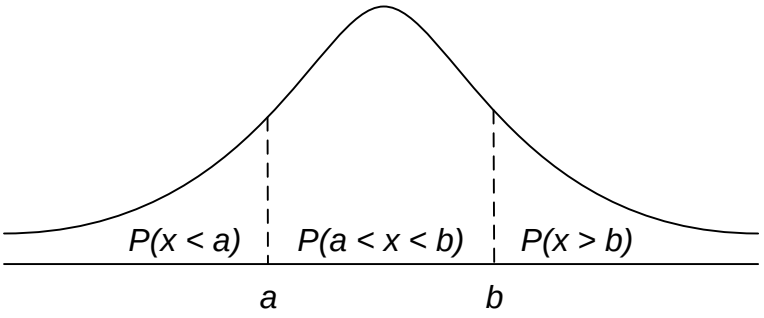

Figure 6.3-1: Probabilities as areas under a probability density curve with a normal distribution (Devore & Peck, 1986)

Different probability curves exist, and according to the type of curve that the data represents, calculations can be made to determine the CI.

Confidence Interval

The fact that only samples of the entire language are used during the evaluation process, the results of any evaluation can never be 100% accurate, as this can only be determined if the entire population is evaluated. One can, however, determine a confidence limit or interval, and conclude that the values of any evaluation will fall within the CI with a degree of certainty. A CI, according to Devore & Peck

(1986:273), is "an interval of plausible values for the characteristic" that is constructed in such a way as to determine with a suitably high degree of confidence, that a characteristic value will fall within a particular interval.

The CI is of importance if one wants to determine within which range of values a metric will probably fall, given a certain set of criteria. In this way evaluations can be structured to attain values within a certain limit, with a high enough confidence level to generalise an evaluation over the entire population.

The CI can be calculated by using the following formula:

$$P\left\{\frac{m}{n} - u_\alpha * \sqrt{\frac{\frac{m}{n} * (1 - \frac{m}{n})}{n}} < x < \frac{m}{n} + u_\alpha * \sqrt{\frac{\frac{m}{n} * (1 - \frac{m}{n})}{n}}\right\} \cong 1 - \alpha \qquad (6.3\text{-}1)$$

where:

- n is the sample size
- m is the number of favoured elements in the sample
- u_α = 1.96 for α = 0.05 (meaning that one can estimate with 95% certainty that the value of the sample (x) will fall between the values on the left and right).

The following hypothetical example illustrates how the CI can be applied, and how this can be interpreted. As a hypothetical example, in an evaluation of 1,000 correct words it was found that 965 of the words were accepted as correctly spelled by a spelling checker, yielding a recall score of 96,50%. Let the significance level be 1-α = 0.95 and construct a CI:

$$\frac{m}{n} = \frac{965}{1000} = 0.965;$$

$$\sqrt{\frac{\frac{m}{n} * (1 - \frac{m}{n})}{n}} = \sqrt{\frac{0.965 * 0.035}{1000}} = 0.0058$$

CI:

$$0.965 - 1.96 * 0.0058 < x < 0.965 + 1.96 * 0.0058$$

$$0.953632 < x < 0.976368$$

This means that from the hypothetical evaluation it can be concluded that 95% of similar evaluations have recall scores higher than 95.36% and lower than 97.64%. In order to increase the significance level to 99%, u_{α} has to increase to 2.58, which would make the CI $0.950036 < x < 0.979964$.

Standard deviation

A sample's standard deviation (σ) is described as the average of positive and negative deviation of the sample measures from the mean (arithmetic average) of the sample. This σ is expressed in the same units as the sample values, and denotes how far values on average lie above or below the mean. The greater σ in the sample, the greater the variability in the sample data. σ is of significance to this study, as the stability of the metrics described in this study are deduced from σ of the values attained for each metric.

In order to use σ to deduce the characteristics of a population, it is necessary to construct a histogram of the sample data. The form of the histogram then gives an indication of the type of inferential statistics that can be used to make assumptions about the population. Of specific interest in this regard is the normal distribution that forms the curve presented in Figure 6.3-1. According to this histogram, the distribution curve represents a normal distribution. Based on the normal distribution of the histogram, the chance that a value will fall within a specific region can be calculated according to the empirical rule. This rule states the following:

"If the histogram of values in a data set can be reasonably well approximated by a normal curve then:

- approximately 68% of observations will fall within 1 σ of the mean;
- approximately 95% of the observations will fall within 2 σs of the mean;
- approximately 99.7% of the observations will fall within 3 σs of the mean.

This is similar to the CI, as it gives an indication of the probability that an observation falls within a given range of values. For instance, if a sample has a mean of 0.965 and a σ of 0.0057, the CI of a random sample x will be as follows:

- 68% Confidence level: $0.9593 < x < 0.9707$

- 95% Confidence level: $0.9536 < x < 0.9764$
- 99.7% Confidence level: $0.9479 < x < 0.9821$

These calculations can be used to determine how well a given experiment approximates the actual value of a metric, and can therefore be used to determine how a given variable influences the values attained by a metric.

F Test

The *F* test is a variance ratio test that compares the variance between two or more independent samples and indicates whether the variance is significant. This test validates or rejects the hypothesis that the variance in different samples is significantly different or not. The aim of the *F* test is therefore to prove either the null hypothesis (H_0) or the alternative hypothesis (H_a) as true, where:

- H_0: $\mu_1=\mu_2=\mu_3\ldots=\mu_k$
- H_a: at least two among $\mu_1, \mu_2, \mu_3\ldots\mu_k$ are different

The calculation of the *F* test represents how closely the values of samples are grouped. This makes it possible to compare samples with different means and values, to determine whether the variance within each sample is deemed significant. Depending on the significance of the variance, a given hypothesis relating to the sample can then be accepted or rejected.

The single factor ANOVA *F* test uses an *F* ratio of the regression mean square (MSR) and the mean square root of error (MSE). These two measures are calculated as follows:

$$MSR = \frac{n_1(\bar{x}_1 - \bar{\bar{x}})^2 + n_2(\bar{x}_2 - \bar{\bar{x}})^2 + \ldots + n_k(\bar{x}_k - \bar{\bar{x}})^2}{k-1} \qquad (6.3\text{-}2)$$

$$MSE = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2 + \ldots(n_k - 1)s_k^2}{N-k} \qquad (6.3\text{-}3)$$

where:

- n is the sample size
- $\bar{x}$ is the sample mean

- $\bar{\bar{x}}$ is the average of all observations
- s^2 is the sample variance
- k is the number of populations being compared

The *F* test utilises the test statistic:

$$F = \frac{MSR}{MSE} \qquad (6.3\text{-}4)$$

to attain *F* distribution of the samples. For an *F* distribution (Figure 6.3-2) the α-level (probability of an error) is calculated by obtaining the critical values from an *F* distribution critical value table[19].

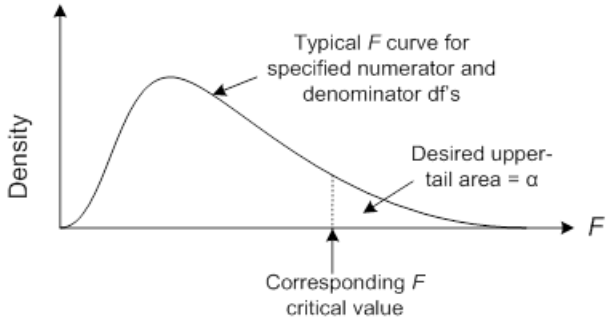

Figure 6.3-2: F curve and F critical value

The calculation of this critical value is dependent on the number of populations being compared (numerator) and the total number of samples being used (denominator). The upper-tail area (α) (to the right of the *F* critical value in Figure 6.3-2) of the *F* curve represents the area where H_0 is rejected. The following example gives a clearer understanding of the *F* test.

An individual's critical flicker frequency (cff) is the highest frequency at which the flicker in a light source can be detected. An investigation carried out to see if the true average cff depends on iris colour yielded the following data.

[19] The *F* distribution critical value table is a standardised set of values relating the significance level of *F* distributions of given Numerator and Denominator degrees of Freedom. This table can be found in most introductory statistical data analysis handbooks.

	Iris colour	Data	n	$\bar{x}$	s^2
1.	Brown	26.8, 27.9, 23.7, 25.0, 26.3, 24.8, 25.7, 24.5	8	25.59	1.86
2.	Green	26.4, 24.2, 28.0, 26.9, 29.1	5	26.92	3.40
3.	Blue	25.7, 27.2, 29.9, 28.5, 29.4, 28.3	6	28.17	2.33

Table 64: Data from a cff test

Also: N = 19, k = 3, $\bar{\bar{x}}$ = 26.75

For a level 0.05 test (i.e. with 95% certainty), an F test is performed through the following steps:

1. Let μ_1, μ_2 and μ_3 denote the true average cff of individuals with brown, green, and blue iris colours.
2. H_0: $\mu_1=\mu_2=\mu_3$
3. H_a: at least two among μ_1, μ_2, μ_3 are different
4. Test statistic: F = MSTr/MSE
5. Rejection region:

   Numerator degrees of freedom = k - 1 = 2

   Denominator degrees of freedom = N - k - 19 - 3 = 16

   For α = 0.05, F critical value = 3.63

   H_0 will be rejected in favour of H_a if $F > 3.63$

6. Calculation:

$$MSTr = \frac{8(25.59-26.75)^2 + 5(26.92-26.75)^2 + 6(28.17-26.75)^2}{3-1} = 11.50$$

$$MSE = \frac{(8-1)(1.86) + (5-1)(3.40) + (6-1)(2.33)}{19-3} = 2.39$$

and F = 11.50/2.39 = 4.81

7. Since 4.81 > 3.63 H_0 is rejected and it can be accepted that true cff average depends on iris colour because the variance within the samples are significant.

This same method is used in the metric validations, but the calculations are not done in full, as the rejection level (α) of the experiments can be generated automatically by statistical software. This makes it possible to evaluate a given null hypothesis.

ADDENDUM B.

Afrikaans error generation module

The first part of the module creates an "alphabet" for the language, by determining all the possible alphanumeric characters that occur in the language, by analysing all the words in a corpus of the language. This alphabet is then used to create all the possible insertions that occur for each input word. Each of the characters of this "alphabet" is then inserted in every position of the word to create typographical errors. This alphabet is also used to create all the possible substitutions of the word, by changing each letter in the word with each letter in the alphabet.

The rest of the module creates all the possible deletions of each word, by removing each character in the word and all the transpositions by exchanging all the positions of characters next to one another. The word "*prose*" would then give the following transpositions:

- *rpose
- *porse
- *prsoe
- *proes

The decision to transpose only adjacent characters was made because transposing each character to all the possible positions in a word would result in too many words that are only vaguely related to the original word. As an example, the word *prose* can be transposed to the following possibilities by just changing the position of "*p*":

- *rpose
- *ropse
- *rospe
- *rosep

From these examples, it can be seen that the second to fourth transpositions are closer to *rope*, *ropes* and *rose* than to *prose* than less likely variations of *prose*.

The number of possible variations given a word of *n* letters, and assuming an alphabet of 38 characters – the English alphabet (including the apostrophe and hyphen) and numerical characters – is calculated as follows (Peterson, 1986: 635):

- Insertions: $38(n + 1)$
- Substitution: $37n$
- Deletion: n
- Transposition: $n - 1$

The total number of variations for each language would differ according to the number of characters in the alphabet and the number of characters in the words that are used to create the variations. As an example, the word *"man"* has all the following forms supposing an alphabet consisting of [*man*].

aan

aman

amn

an

maa

maan

mam

mamn

mana

manm

mann

mman

mmn

mn

mna

mnan

mnn

nan

nman

REFERENCE LIST

Abney, S., Flickinger, D., Gdaniec, C., Grishman, R., Harrison, P., Hindle, D., Ingria, R., Jelinek, F., Klavans, J., Liberman, M., Marcus, M., Roukos, S., Santorini, B. & Strzalkowski, T. 1991. A procedure for quantitatively comparing the syntactic coverage of English grammars. (*In* Proceedings of the DARPA Speech and Natural Language Workshop, Pacific Grove, CA. p. 306-311).

Akkaya, C., Conrad, A., Wiebe, J. & Mihalcea, R. 2010. Amazon mechanical Turk for subjectivity word sense disambiguation. (*In* Proceedings of the North American Association for Computational Linguistics: Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, Los Angeles, CA. p. 195-203).

ALPAC (Automatic language processing advisory committee). 1966. Languages and machines: computers in translation and linguistics. Washington, DC: National Academy of Sciences.

Ambati, V., Vogel, S. & Carbonell, J. 2010. Active learning and crowd-sourcing for machine translation. *Language resources and evaluation*, 7:2169-2174.

Amigó, E., Artiles, J., Li, Q. & Ji, H. 2011. An evaluation framework for aggregated temporal information extraction. (*In* Balog, K., De Vries, A.P., Serdyukov, P. & Wen, J-R., *eds.* Special Interest Group on Information Retrieval 2011: Workshop on Entity-Oriented Search. Delft: TU Delft. p. 59-64).

Amigó, E., Gonzalo, J., Gimenez, J. & Verdejo, F. 2011. Corroborating text evaluation results with heterogeneous measures. (*In* Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland. p. 455-466).

Bailin, A. & Grafstein, A. 2001. The linguistic assumptions underlying readability formulae: a critique. *Language & communication*, 21:285-301.

Balkan, L., Arnold, D. & Meijer, S. 1994. Test suites for natural language processing. *Aslib proceedings*, 47(4):95-98.

Banerjee, S. & Lavie, A. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. (*In* Proceedings of the Association for Computational Linguistics Workshop: Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor, MI. p. 65-72).

Berg-Kirkpatrick, T., Burkett, D. & Klein, D. 2012. An empirical investigation of statistical significance in NLP. (*In* Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers, Jeju, Korea. 1:995-1005).

Bernsen, N.O. & Dybkjær, A. 2000. A methodology for evaluating spoken language dialogue systems and their components. (*In* Proceedings of the 2nd International Conference on Language Resources and Evaluation, Athens, Greece. p. 183-188).

Bianchi, F. 2012. Culture, corpora and semantics: methodological issues in using elicited and corpus data for cultural comparison. Lecce: University of Salento.

Biber, D. 1993. Representativeness in corpus design. *Literary and linguistic computing*, 8(4):243-257.

Burk, F. 1985. By all means. *The American mathematical monthly*, 92(1):50.

Bussmann, S., Jennings, N.R. & Wooldridge, M. 2001. On the identification of agents in the design of production control systems. (*In* Ciancarini, P. & Wooldridge, M., *eds.* Agent-oriented software engineering. Berlin: Springer-Verlag. p. 141-162).

Butler, A. & Van Huyssteen, G.B. 2012. 5000 Afrikaans woorde, gekategoriseer volgens spellingprobleme. Weergawe 1.0. Potchefstroom: NWU. (Unpublished).

Callison-Burch, C. 2009. Fast, cheap, and creative: evaluating translation quality using Amazon's mechanical Turk. (*In* Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland. p. 286-295).

Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C. & Schroeder, J. 2008. Further meta-evaluation of machine translation. (*In* Proceedings of the 3rd Workshop on Statistical Machine Translation, Columbus, OH. p. 70-106).

Callison-Burch, C., Osborne, M. & Koehn, P. 2006. Re-evaluating the role of BLEU in machine translation research. (*In* Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, Trento, Italy. p. 249-256).

Carstens, W.A.M. 2003. Norme vir Afrikaans: enkele riglyne by die gebruik van Afrikaans. Pretoria: Van Schaik.

Chantree, F., Kilgarriff, A., De Roeck, A. & Willis, A. 2005. Technical report: using a distributional thesaurus to resolve coordination ambiguities. http://computing-reports.open.ac.uk/index.php/content/download/180/1087/file/TR2005_02.pdf Date of access: 12 Sep. 2009.

Chung, T.M. 2003. A corpus comparison approach for terminology extraction. *Terminology*, 9(2):221-246.

Cimiano, P., Ciravegna, F., Domingue, J., Handschuh, S., Lavelli, A., Staab, S. & Stevenson, M. 2003. Requirements for information extraction for knowledge management. (*In The* 2nd Workshop on Knowledge Markup and Semantic Annotation, Sanibel, FL. p. 89-94).

Cimiano, P., Handschuh, S. & Staab, S. 2004. Towards the self-annotating web. (*In* Proceedings of the 13th International Conference on World Wide Web, New York, NY. p. 462-471).

Cleverdon, C.W. 1967. The Cranfield tests on index language devices. *Aslib proceedings*, 19:173-192.

Cohen, K.B., Hunter, L.E. & Palmer, M. 2012. Assessment of software testing and quality assurance in natural language processing applications and a linguistically inspired approach to improving it. (*In* Proceedings of the Joint Workshop on Intelligent Methods for Software System Engineering, Mont Pellier, France. p. 11-17).

Comrie, B. 1989. Language universals and linguistic typology: syntax and morphology. Chicago, IL: University of Chicago Press.

CoNLL (Conference on computational natural language learning). 2002. CoNLL shared task: language-independent named entity recognition. http://www.cnts.ua.ac.be/conll2002/ner/ Date of access: 8 Aug. 2012.

CoNLL (Conference on computational natural language learning). 2003. CoNLL shared task: language-independent named entity recognition. http://www.cnts.ua.ac.be/conll2003/ner/ Date of access: 8 Aug. 2012.

Covington, M.A. & McFall, J.D. 2010. Cutting the Gordian knot: the moving-average type-token ratio (MATTR). *Journal of quantitative linguistics.* 17(2):94-100.

Crossley, S., Allen, D.B. & McNamara, D.S. 2011. Text readability and intuitive simplification: a comparison of readability formulas. *Reading in a foreign language*, 23(1):84-101.

Daelemans, W., Zavrel, J., Berck, P. & Gillis, S. 1996. MBT: a memory-based part of speech tagger-generator. (*In* Proceedings of the Workshop on Very Large Corpora, Copenhagen, Denmark. p. 14-27).

Damerau, F.J. 1964. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7(3):171-176.

De Clercq, O., Hoste, V., Desmet, B., Van Oosten, P., De Cock, M. & Macken, L. 2013?. Using the crowd for readability prediction. *Natural language engineering* (In press).

De Jong, M. & Schellens, P.J. 2000. Toward a document evaluation methodology: what does research tell us about the validity and reliability of evaluation methods? *IEEE transactions on professional communication,* 43(3):242-260.

Devore, J. & Peck, R. 1986. Statistics: the exploration and analysis of data. New York, NY: West Publishing Company.

Diab, M., Hacioglu, K., Jurafsky, D. 2004. Automatic tagging of Arabic text: from raw text to base phrase chunks (*In* Proceedings of Human Language Technology – North American Association for Computational Linguistics: Short Papers, Boston, MA. p. 149-152).

Dietterich, T.G. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895-1923.

Doyon, J., Taylor, K. & White, J. 1998. The DARPA machine translation evaluation methodology. (*In* Proceedings of the American Machine Translation Association, Langhorn, PA. p. 364-373).

Dybkjær, L., Hemsen, H. & Minker, W., *eds.* 2007. Evaluation of text and speech systems. Dordrecht: Springer.

El-Emam, K. 2000. A methodology for validating software product metrics. Ottowa: National Research Council.

Entwisle, J. & Powers, D.M.W. 1998. The present use of statistics in the evaluation of NLP parsers. (*In* Proceedings of the Conference on Computational Natural Language Learning: New Methods in Language Processing and Computational Natural Language Learning, Sydney, Australia. p. 215-224).

EAGLES (Expert advisory group for language engineering standards). 1995a. International standards organization 9126. http://www.issco.unige.ch/ewg95/node14.html Date of access: 12 Aug. 2009.

EAGLES (Expert advisory group for language engineering standards). 1995b. The EAGLES extension to ISO 9126. http://www.issco.unige.ch/ewg95/node15.html Date of Access: 12 Aug. 2009.

Field, A. 2005. Discovering statistics using SPSS. London: Sage.

Fillmore, C., Ide, N., Jurafsky, D. & Macleod, C. 1998. An American national corpus: a proposal. (*In* Proceedings of the 1st Conference on Language Resources and Evaluation, Granada, Spain. p. 965-969).

Finin, T., Murnane, W., Karandikar, A., Keller, N., Martineau, J. & Dredze, M. 2010. Annotating named entities in twitter data with crowdsourcing. (*In* Proceedings of the North American Association for Computational Linguistics: Workshop on Creating

Speech and Language Data with Amazon's Mechanical Turk, Los Angeles, CA. p. 80-88).

Galliers, J.R. & Sparck-Jones, K. 1993. Evaluating natural language processing systems. Cambridge: Cambridge University.

Green, T.R. & Hendry, D.G. 1993. Spelling mistakes: how well do correctors perform? (*In* INTERACT and CHI Conference Companion on Human Factors in Computing Systems, Amsterdam, The Netherlands. p. 83-84).

Grishman, R., & Sundheim, B. 1996. Message understanding conference-6: a brief history. *COLING*, 96:466-471.

Hirschman, A. & Thompson, H.S. 1997. Overview of evaluation in speech and natural language processing. (*In* Cole, R., Mariane, J. Uszkoreit, A., Varile, G.B., Zaenen, A., Zampolli, A. & Zue, V., *eds.* Survey of the state of the art in human language technology. Cambridge: Cambridge University Press. p. 409-414).

Hirschman, L. & Mani, I. 2003. Evaluation. (*In* Mitkov, R., *ed.* Handbook of computational linguistics. Oxford: Oxford University Press. p. 414-430).

Hovy, E., King, M. & Popescu-Belis, A. 2002. An introduction to MT evaluation. (*In* Proceedings of Language Resource and Evaluation: Workshop on Machine Translation Evaluation - Human Evaluators Meet Automated Metrics, Las Palmas, Spain. p. 1-7).

Hutchins, W.J. 1995. Machine translation: a brief history. (*In* Koerner, E.F.K. & Asher, R.E., *eds.* Concise history of the language sciences: from the Sumerians to the cognitivists. Oxford: Pergamon Press. p. 431-445).

Jurafski, D. & Martin, J.H. 2000. Speech and natural language processing: an introduction to natural language processing, computational linguistics and speech recognition. Upper Saddle River, NJ: Prentice Hall.

Kate, R.J., Luo, X., Patwardhan, S., Franz, M., Florian, R., Mooney, R.J. & Welty, C. 2010. Learning to predict readability using diverse linguistic features. (*In* Proceedings of the

23rd International Conference on Computational Linguistics, Beijing, China. p. 546-554).

Kilgariff, A. 1998. SENSEVAL: an exercise in evaluating WSD programs. (*In* Proceedings of the 1st Conference on Language Resources and Evaluation, Granada, Spain. p. 566-572).

Kim, S., Toutanova, K. & Yu, H. 2012. Multilingual named entity recognition using parallel data and metadata from Wikipedia. (*In* Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers, Jeju, Korea. 1:694-702).

King, M. 1996. Evaluating natural language processing systems. *Communications of the ACM*, 39(1):73-79.

King, M. 1998. Language resources and evaluation. (*In* Proceedings of the International Conference on Natural Language Processing and Industrial Applications, Moncton, Canada. p. 1- 13).

King, M. 1999. Evaluation design: the EAGLES framework. (*In* Nübel, R., Seewald-Heeg, U. & Verlag, G., *eds.* Proceedings of the Konvens '98: Evaluation of the Linguistic Performance of Machine Translation Systems, Bonn, Germany.).

King, M. 2005. Accuracy and suitability: new challenges for evaluation. *Language resources and evaluation*, 39:45-64.

King, M. & Maegaard, B. 1998. Issues in natural language system evaluation. (*In* Proceedings of the 1st Conference on Language Resources and Evaluation, Granada, Spain. p. 225-230).

Kitchenham, B., Pfleeger, S-L. & Fenton, N. 1995. Towards a framework for software measurement validation. *IEEE transactions on software engineering*, 21(12):929-944.

Koehn, P. 2004. Statistical significance tests for machine translation evaluation. (*In* Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain. p. 388-395).

Koller, A., Striegnitz, K, Byron, D., Cassell, J., Dale, R., Dalzel-Job, S., Moore, J. & Oberlander, J. 2009. Validating the web-based evaluation of NLG systems. (*In* Proceedings of the ACL-IJCNLP, Singapore. p. 301.-304).

Kukich, K. 1992. Spelling correction for the telecommunications network for the deaf. *Communications of the ACM*, 35(5): 80-90.

Lavelli, A., Califf, M., Ciravegna, F., Freitag, D., Giuliano, C., Kushmerick N. & Romano, L. 2004. IE evaluation: criticisms and recommendations. (*In* AAAI-04 Workshop on Adaptive Text Extraction and Mining, San Jose, California. p. 7-14).

Lavie, A., Sagae, K. & Jayaraman, S. 2004. The significance of recall in automatic metrics for MT evaluation. (*In* Frederking, R.E. & Taylor, K.B., *eds.* Proceedings of the 6th Conference of the Association for Machine Translation in the Americas, Washington, DC. p. 134-143).

Leech, G. 2006. New resources, or just better old ones? The Holy Grail of representativeness. *Language and computers*, 59(1):133-149.

Lehmann, S., Oepen, S., Regnier-Prost, S., Netter, K., Lux, V., Klein, J., Falkedal, K., Fouvry, F., Estival, D., Dauphin, E., Compagnion, H., Baur, J., Balkan, L. & Arnold, D. 1996. TSNLP - test suites for natural language processing. (*In* Proceedings of the 16th International Conference on Computational Linguistics, Copenhagen, Denmark. p. 711-717).

Lenci, A., Montemagni, S., Pirrelli, V., Soria, C., Netter, K. & Rajman, R. 1999. Evaluation of language and speech engineering (ELSE) final report: corpora for evaluation http://www.limsi.fr/TLP/ELSE/FullXreportXver302.htm Date of access: 20 Feb. 2008.

Lesk, M. 1996. The future of software. *Journal of the American Society for Information Science and Technology*, 47(4):326-327

Limbrunner, J.F., Vogel, R.M. & Brown, L.C. 2000. Estimation of harmonic mean of a lognormal variable. *Journal of hydrologic engineering*, 5(1):59-66.

Lin, C.Y. 2004. Rouge: a package for automatic evaluation of summaries. (*In* Proceedings of the Association for Computational Linguistics: Workshop on Text Summarization Branches Out, Barcelona, Spain. p. 74-81).

Lin, C.Y. & Hovy, E.H. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. (*In* Proceedings of the 2003 Language Technology Conference, Edmonton, Canada. p. 71-78).

Lin, J. & Katz, B. 2006. Building a reusable test collection for question answering. *Journal of the American Society for Information Science and Technology*, 57(3):851-861.

Liu, C. & Ng, H.T. 2012. Character-level machine translation evaluation for languages with ambiguous word boundaries. (*In* Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers, Jeju, Korea. 1:921-929).

Luo, F., Fang, P., Qiu, Q. & Xiao, H. 2012. Features induction for product named entity recognition with CRFs. (*In* IEEE 16th International Conference: Computer Supported Cooperative Work in Design, Wuhan, China. p. 491-496).

Lynn, T., Cetinoglu, O., Foster, J., Dhonnchadha, E. U., Dras, M. & van Genabith, J. 2012. Irish treebanking and parsing: a preliminary evaluation. (*In* Calzolari, N., Choukri, K., Declerck, T., Doğan, M.U., Maegaard, B., Mariani, J., Odijk, J. & Piperidis, S. *eds.* Proceedings of the 8th International Conference on Language Resources and Evaluation Turkey: ELRA. p. 1939-1946).

Macmullen, W.J. 2003. Requirements definition and design criteria for test corpora in information science. Chapel Hill: University of North Carolina.

Madnani, N., Tetreault, J., Chodorow, M. & Rozovskaya, A. 2011. They can help: using crowdsourcing to improve the evaluation of grammatical error detection systems. (*In* Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers, Portland, Oregon. 2:508-513).

Mani, I., House, D., Klein, G., Hirschman, L., Firmin, T. & Sundheim, B. 1999. The TIPSTER SUMMAC text summarization evaluation. (*In* Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics, Bergen, Norway. p. 77-85).

Manning, C.D. 2011. Part-of-speech tagging from 97% to 100%: is it time for some linguistics?. (*In* Gelbukh, A., *ed.* Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing: Part I. Berlin: Springer-Verlag. p. 171-189).

Manning, C.D. & Schütze, H. 1999. Foundations of statistical natural language processing. Cambridge, MA: MIT Press.

Manzi, S., King, M. & Douglas, S. 1996. Working towards user-oriented evaluation. (*In* Proceedings of the International Conference on Natural Language Processing and Industrial Applications, Moncton, Canada. p. 155-160).

Marsland, S. 2009. Machine learning: an algorithmic perspective. Boca Raton: Chapman & Hall.

Martin, A. & Przybocki, M. 2000. The NIST 1999 speaker recognition evaluation - an overview. *Digital signal processing,* 10(1):1-18.

Mitchell, T.M. 1997. Machine learning. Boston, MA: McGraw-Hill.

Mitton, R. 1996. English spelling and the computer. London: Longman.

Murthy, U., Pitrelli, J.F., Ramaswamy, G., Franz, M. & Lewis, B.L. 2008. A methodology and tool suite for evaluation of accuracy of interoperating statistical natural language processing engines. (*In* Proceedings of the International Speech Communication Association Conference, Brisbane, Australia. p. 2066-2069).

Nance, R.E. & Arthur, J.D. 1988. The methodology roles in the realization of a model development environment. (*In* Proceedings of the 1988 Winter Simulation Conference, San Diego, CA. p. 220-225).

NIST (National Institute of Science and Technology). 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. http://nist.gov/speech/tests/mt/ Date of access: 20 Feb. 2008.

Neter, J., Kutner, M.H., Nachtsheim, C.J. & Wasserman, W. 1996. Applied linear statistical models. 4th ed. Boston, MA: McGraw-Hill.

Netter, K., Armstrong, S., Kiss, T., Klein, J., Lehmann, S., Milward, D., Regnier-Prost, S., Schäler, R. & Wegst, T. 1998. Diagnostic and evaluation tools for natural language processing applications. (*In* Proceedings of the 1st International Conference on Language Resources and Evaluation, Granada, Spain. p. 573-579).

Olivier Hamon, E.L.D.A., Choukri, K. & Mokbel, C. 2010. Evaluation methodology and results. http://www.medar.info/MEDAR-evaluation_I.pdf Date of access: 13 Oct. 2012.

Paggio, P. & Music, B. 1998. Evaluation in the SCARRIE project. (*In* Proceedings of the 1st International Conference on Language Resources and Evaluation, Granada, Spain. p. 277-282).

Paggio, P. & Underwood, N.L. 1997. Validating the TEMAA LE evaluation methodology: a case study on Danish spelling. *Natural language engineering*, 1(1):1-18.

Palmer, M. & Finin, T. 1990. Workshop on the evaluation of natural language processing systems. *Computational linguistics*, 16(3):175-181.

Papineni, K., Roukos, S., Ward, T. & Zhu, W. 2002. BLEU: a method for automatic evaluation of machine translation. (*In* Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, PA. p. 311-318).

Paroubek, P. 2007. Evaluating part of speech tagging and parsing. (*In* Dybkjær, L., Hemsen, H. & Minker, W., *eds.* Evaluation of text and speech systems. Dordrecht: Springer. p. 291-310).

Pate, J.K. & Goldwater, S. 2011. Unsupervised syntactic chunking with acoustic cues: computational models for prosodic bootstrapping. (*In* Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics, Portland, Oregon. p. 20-29).

Peterson, J.L. 1986. A note on undetected typing errors. *Communications of the ACM*, 29(7):633-637.

Petrov, S., Das, D. & Mcdonald, R. 2011. A universal part-of-speech tagset. *arXiv*, 1104:2086.

Pilon, S. 2006. Outomatiese Afrikaanse woordsoortetikettering. Potchefstroom: NWU. (Thesis - MA).

Pirkola, A. 2001. Morphological typology of languages for IR. *Journal of documentation*, 57(3):330-348.

Pollock, J.J. & Zamora, A. 1984. Automatic spelling correction in scientific and scholarly text. *Communications of the ACM*, 27(4):358-368.

Popescu, I.I., Altmann, G. & Čech, R. 2011. The Lambda-structure of texts. Lüdenscheid: RAM-Verlag.

Prasad, R. & Sarkar, A. 2000. Comparing test-suite based evaluation and corpus-based evaluation of a wide-coverage grammar for English. (*In* Proceedings of the 2nd International Conference on Language Resources and Evaluation, Athens, Greece).

Puttkammer, M.J. 2006. Outomatiese Afrikaanse tekseenheididentifisering. Potchefstroom: NWU. (Thesis - MA).

Puurula, A. & Kurimo, M. 2007. Vocabulary decomposition for Estonian open vocabulary speech recognition. (*In* Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic. p. 89-95).

Rasmussen, E. 2003. Evaluation in information retrieval. (*In* The MIR/MDL evaluation project white paper collection edition. p. 45-49).

Rennie, J.D.M. 2004. Derivation of the F-measure. http://www.people.csail.mit.edu/jrennie/writing/fmeasure.pdf Date of access: 16 Oct. 2009.

Reynaert, M. 2006. Text induced spelling correction. Tillburg: Tillburg University. (Dissertation - PhD).

Riley, M., Craven, L. & Broman-Olsen, M. 2004. Customer-focused evaluation of a grammar checker. (*In* Proceedings of the 1st Workshop on International Proofing Tools and Language Technologies, Patras, Greece. p. 377-389).

Robertson, S. 2008. On the history of evaluation in IR. *Journal of Information Science*, 34(4):439-456.

Rodrigo, Á., Peñas, A., Hovy, E. & Pianta, E. 2010. Question answering for machine reading evaluation. http://clef2010.org/resources/proceedings/clef2010labs_submission_40.pdf Date of access: 19 Sept. 2011.

Rossi, A. & Fernandez, G. 2003. Definition and validation of design metrics for distributed applications. (*In* Proceedings of the 9th International Software Metrics Symposium, Sydney, Australia. p. 124-132).

Ruppenhofer, J., Sporleder, C., Morante, R., Baker, C. & Palmer, M. 2010. Semeval-2010 task 10: linking events and their participants in discourse. (*In* Erk, K. & Strapparava, C., *eds.* Proceedings of the 5th International Workshop on Semantic Evaluation, Uppsala, Sweden. p. 45-50).

Sasaki, Y. 2007. The truth of the F-measure. http://www.cs.odu.edu/~mukka/cs795sum11dm/Lecturenotes/Day3/F-measure-YS-26Oct07.pdf Date of access: 9 Sep. 2012.

Schwartz, R., Omri Abend, O., Roi Reichart, R. & Rappoport, A. 2011. Neutralizing linguistically problematic annotations in unsupervised dependency parsing evaluation. (*In* Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Long Papers, Portland, Oregon. 1:663-672).

Shin, H. & Stach, J.F. 2000. Using long runs as predictors of semantic coherence in a partial document retrieval system. (*In* North American Association for Computational Linguistics Workshop: Syntactic and Semantic Complexity in Natural Language Processing, Seattle, WA. p. 6-13).

Sinclair, J. 2004. Corpus and text: basic principles. (*In* Wynne, M., *ed.* Developing linguistic corpora: a guide to good practice. Oxford: Oxbow Books. p. 1-16).

Singhal, A. 2001. Modern information retrieval: a brief overview. *Bulletin of the IEEE computer society technical committee on data engineering*, 24(4):35-43.

Smrž, P. 2004. Quality control for wordnet development. (*In* Proceedings of the 2nd International Wordnet Conference, Brno, Czech Republic. p. 206-212).

Snover, M., Dorr, B., Schwartz, R., Micciulla, L. & Makhoul, J. 2006. A study of translation edit rate with targeted human annotation. (*In* Proceedings of the 7th Conference of the Association for Machine Translation in the Americas, Cambridge, MA. p. 223-231).

Soricut, R. & Brill, E. 2004. A unified framework for automatic evaluation using n-gram co-occurrence statistics. (*In* Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain. p. 613-620).

Starlander, M. & Popescu-Belis, A. 2002. Corpus-based evaluation of a French spelling and grammar checker. (*In* Proceedings of the 3rd International Conference on Language Resources and Evaluation, Las Palmas, Spain. p. 268-274).

Suid-Afrikaanse Akademie Vir Wetenskap En Kuns. 2002. Afrikaanse woordelys en spelreëls. Cape Town: Pharos Dictionaries.

Sundheim, B. M. 1992. Overview of the fourth message understanding evaluation and conference. (*In* Proceedings of the 4th conference on Message understanding, Association for Computational Linguistics. p. 3-21).

Taylor, B.N. & Kuyatt, R. 1994. Guidelines for evaluating and expressing the uncertainty of NIST measurement results Washington, DC: US Government Publishing.

TEMAA (A testbed study of evaluation methodologies: authoring aids). 1997. Evaluation framework for spelling checkers: final report. http://www.cst.dk/temaa/D16/d16exp.html Date of access: 20 Feb. 2008.

Theeramunkong, T. & Okumura, M. 1997. Grammar acquisition based on clustering analysis and its application to statistical parsing. (*In* Proceedings of the 5th Workshop on Very Large Corpora, Beijing, China. p. 31-40).

Thompson, H.S. 1991. Automatic evaluation of translation quality: outline of methodology and report on pilot experiment. (*In* Proceedings of the Evaluator's Forum, Les Rasses, Switzerland. p. 215-223).

Turian, J.P, Shen, L. & Melamed, I.D. 2003. Evaluation of machine translation and its evaluation. (*In* Proceedings of MT Summit IX, New Orleans, LA. p. 386-393).

Van Huyssteen, G.B., Puttkammer, M.J. & Eiselen, E.R. 2004. Evaluating evaluation metrics for spelling checker evaluations. (*In* Proceedings of the 1st Workshop on International Proofing Tools and Language Technologies, Patras, Greece. p. 91-99).

Van Rijsbergen, C.J. 1979. Information retrieval. London: Butterworths.

Van Zaanen, M.M. & Van Huyssteen, G.B. 2003. Improving a spelling checker for Afrikaans. (*In* Gaustad, T., *ed.* Computational linguistics in the Netherlands 2002: selected papers from the 13th CLIN meeting. Amsterdam: Rodopi. p. 143-156).

Vanni, M. & Reeder, F. 2000. How are you doing? A look at MT evaluation. (*In* White, J.S., *ed.* Proceedings of the 4th Conference of the Association for Machine Translation in the Americas, Cuernavaca, Mexico. p. 109-116).

Verberne, S. 2002. Context-sensitive spell checking based on word trigram probabilities. Nijmegen: Katholieke Universiteit Nijmegen. (Dissertation - MA).

Volokh, A. & Neumann, G. 2012. Task-oriented dependency parsing evaluation methodology. (*In* IEEE 13th International Conference on Information Reuse and Integration, Las Vegas, NV. p. 132-137).

Voorhees, E.M. 2002. The philosophy of information retrieval evaluation. (*In* Peters, C., Braschler, M., Gonzalo, J. & Kluck, M., *eds*. Evaluation of cross-language information retrieval systems. Berlin: Springer. p. 143-170).

White, J.S. & O'Connell, T. 1994. The ARPA MT evaluation methodologies: evolution, lessons, and future approaches. (*In* Proceedings of the 1st Conference of the Association for Machine Translation in the Americas, Columbia, MD. p. 193-205).

Yannakoudakis, E.J. & Fawthrop, D. 1983. The rules of spelling errors. *Information processing and management*, 19(2):87-99.

Zaidan, O.F. & Callison-Burch, C. 2010. Predicting human-targeted translation edit rate via untrained human annotators. (*In* Proceedings of the North American Association for Computational Linguistics, Los Angeles, CA. p. 369-372).

Zaidan, O.F. & Callison-Burch, C. 2011. Crowdsourcing translation: professional quality from non-professionals. (*In* Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Long Papers, Portland, Oregon. 1:1220-1229).

Zanettin, F. 2011. Translation and corpus design. *A journal of professional communication*, 26:14-23.