A Morphographemic Model for Error Correction in Nonconcatenative Strings

Tanya Bowden * and **George Anton Kiraz** †
University of Cambridge
Computer Laboratory
Pembroke Street, Cambridge CB2 3QG
{Tanya.Bowden, George.Kiraz}@cl.cam.ac.uk
http://www.cl.cam.ac.uk/users/{tgb1000, gk105}

Abstract

This paper introduces a spelling correction system which integrates seamlessly with morphological analysis using a multi-tape formalism. Handling of various Semitic error problems is illustrated, with reference to Arabic and Syriac examples. The model handles errors vocalisation, diacritics, phonetic syncopation and morphographemic idiosyncrasies, in addition to Damerau errors. A complementary correction strategy for morphologically sound but morphosyntactically ill-formed words is outlined.

1 Introduction

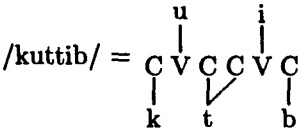
Semitic is known amongst computational linguists, in particular computational morphologists, for its highly inflexional morphology. Its root-and-pattern phenomenon not only poses difficulties for a morphological system, but also makes error detection a difficult task. This paper aims at presenting a morphographemic model which can cope with both issues.

The following convention has been adopted. Morphemes are represented in braces, { }, surface (phonological) forms in solidi, / /, and orthographic strings in acute brackets, ⟨ ⟩. In examples of grammars, variables begin with a capital letter. Cs denote consonants, Vs denote vowels and a bar denotes complement. An asterisk, *, indicates ill-formed strings.

The difficulties in morphological analysis and error detection in Semitic arise from the following facts:

- **Non-Linearity** A Semitic stem consists of a **root** and a **vowel melody**, arranged according to a **canonical pattern**. For example, Arabic /kuttib/ 'caused to write - perfect passive' is composed from the root morpheme {ktb} 'notion of writing' and the vowel melody morpheme {ui} 'perfect passive'; the two are arranged according to the pattern morpheme {CVCCVC} 'causative'. This phenomenon is analysed by (McCarthy, 1981) along the lines of autosegmental phonology (Goldsmith, 1976). The analysis appears in (1).[1]

(1)
DERIVATION OF /kuttib/



- **Vocalisation** Orthographically, Semitic texts appear in three forms: (i) **consonantal texts** do not incorporate any short vowels but *matres lectionis*,[2] e.g. Arabic ⟨ktb⟩ for /katab/, /kutib/ and /kutub/, but ⟨kaatb⟩ for /kaatab/ and /kaatib/; (ii) **partially vocalised texts** incorporate some short vowels to clarify ambiguity, e.g. ⟨kutb⟩ for /kutib/ to distinguish it from /katab/; and (iii) **vocalised texts** incorporate full vocalisation, e.g. ⟨tadaħraĵ⟩ for /tadaħraĵ/.

* Supported by a British Telecom Scholarship, administered by the Cambridge Commonwealth Trust in conjunction with the Foreign and Commonwealth Office.
† Supported by a Benefactor Studentship from St John's College.

[1] We have used the CV model to describe pattern morphemes instead of prosodic terms because of its familiarity in the computational linguistics literature. For the use of moraic and affixational models in handling Arabic morphology computationally, see (Kiraz,).

[2] 'Mothers of reading', these are consonantal letters which play the role of long vowels, and are represented in the pattern morpheme by VV (e.g. /aa/, /uu/, /ii/). *Matres lectionis* cannot be omitted from the orthographic string.

- **Vowel and Diacritic Shifts** Semitic languages employ a large number of diacritics to represent *enter alia* short vowels, doubled letters, and nunation.[3] Most editors allow the user to enter such diacritics above and below letters. To speed data entry, the user usually enters the base characters (say a paragraph) and then goes back and enters the diacritics. A common mistake is to place the cursor one extra position to the left when entering diacritics. This results in the vowels being shifted one position, e.g. *⟨wkatubi⟩ instead of ⟨wakutib⟩.

- **Vocalisms** The quality of the perfect and imperfect vowels of the basic forms of the Semitic verbs are idiosyncratic. For example, the Syriac root {ktb} takes the perfect vowel *a*, e.g. /ktab/, while the root {nḥt} takes the vowel *e*, e.g. /nḥet/. It is common among learners to make mistakes such as */kteb/ or */nḥat/.

- **Phonetic Syncopation** A consonantal segment may be omitted from the *phonetic* surface form, but maintained in the *orthographic* surface from. For example, Syriac ⟨mdīntâ⟩'city' is pronounced /mdītâ/.

- **Idiosyncrasies** The application of a morphographemic rule may have constraints as on which lexical morphemes it may or may not apply. For example, the glottal stop [ʔ] at the end of a stem may become [w] when followed by the relative adjective morpheme {iyy}, as in Arabic /samaaʔ+iyy/ → /samaawiyy/ 'heavenly', but /hawaaʔ+iyy/ → /hawaaʔiyy/ 'of air'.

- **Morphosyntactic Issues** In broken plurals, diminutives and deverbal nouns, the user may enter a morphologically sound, but morphosyntactically ill-formed word. We shall discuss this in more detail in section 4.[4]

To the above, one adds language-independent issues in spell checking such as the four Damerau transformations: omission, insertion, transposition and substitution (Damerau, 1964).

2 A Morphographemic Model

This section presents a morphographemic model which handles error detection in non-linear strings. Subsection 2.1 presents the formalism used, and subsection 2.2 describes the model.

2.1 The Formalism

In order to handle the non-linear phenomenon of Arabic, our model adopts the two-level formalism presented by (Pulman and Hepple, 1993), with the multi tape extensions in (Kiraz, 1994). Their formalism appears in (2).

(2)

TWO-LEVEL FORMALISM

LLC - LEX - RLC {⇒,⇔}
LSC - SURF - RSC

where
LLC = left lexical context
LEX = lexical form
RLC = right lexical context
LSC = left surface context
SURF = surface form
RSC = right surface context

The special symbol * is a wildcard matching any context, with no length restrictions. The operator ⇔ caters for obligatory rules. A lexical string maps to a surface string iff they can be partitioned into pairs of lexical-surface subsequences, where each pair is licenced by a ⇒ or ⇔ rule, and no partition violates a ⇔ rule. In the multi-tape version, lexical expressions (i.e. LLC, LEX and RLC) are n-tuple of regular expressions of the form $(x_1, x_2, \ldots, x_n)$: the ith expression refers to symbols on the ith tape; a nill slot is indicated by ε.[5] Another extension is giving LLC the ability to contain ellipsis, ... , which indicates the (optional) omission from LLC of tuples, provided that the tuples to the left of ... are the first to appear on the left of LEX.

In our morphographemic model, we add a similar formalism for expressing error rules (3).

(3)

ERROR FORMALISM
ErrSurf ⇒ Surf
{ PLC - PRC } where
PLC = partition left context (has been done)
PRC = partition right context (yet to be done)

[3]When indefinite, nouns and adjectives end in a *phonetic* [n] which is represented in the *orthographic* form by special diacritics.

[4]For other issues with respect to syntactic dependencies, see (Abduh, 1990).

[5]Our implementation interprets rules directly; hence, we allow ε. If the rules were to be compiled into automata, a genuine symbol, e.g. 0, must be used. For the compilation of our formalism into automata, see (Kiraz and Grimley-Evans, 1995).

The error rules capture the correspondence between the error surface and the correct surface, given the surrounding partition into surface and lexical contexts. They happily utilise the multi-tape format and integrate seamlessly into morphological analysis. PLC and PRC above are the left and right contexts of both the lexical and (correct) surface levels. Only the $\Rightarrow$ is used (error is not obligatory).

2.2 The Model

2.2.1 Finding the error

Morphological analysis is first called with the assumption that the word is free of errors. If this fails, analysis is attempted again without the 'no error' restriction. The error rules are then considered when ordinary morphological rules fail. If no error rules succeed, or lead to a successful partition of the word, analysis backtracks to try the error rules at successively earlier points in the word.

For purposes of simplicity and because on the whole is it likely that words will contain no more than one error (Damerau, 1964; Pollock and Zamora, 1983), normal 'no error' analysis usually resumes if an error rule succeeds. The exception occurs with a vowel shift error (§3.2.1). If this error rule succeeds, an expectation of further shifted vowels is set up, but no other error rule is allowed in the subsequent partitions. For this reason rules are marked as to whether they can occur more than once.

2.2.2 Suggesting a correction

Once an error rule is selected, the corrected surface is substituted for the error surface, and normal analysis continues - at the same position. The substituted surface may be in the form of a variable, which is then ground by the normal analysis sequence of lexical matching over the lexicon tree. In this way only lexical words are considered, as the variable letter can only be instantiated to letters branching out from the current position on the lexicon tree. Normal prolog backtracking to explore alternative rules/lexical branches applies throughout.

3 Error Checking in Arabic

We demonstrate our model on the Arabic verbal stems shown in (4) (McCarthy, 1981). Verbs are classified according to their **measure** (M): there are 15 trilateral measures and 4 quadrilateral ones. Moving horizontally across the table, one notices a change in vowel melody (active {a}, passive {ui}); everything else remains invariant. Moving vertically, a change in canonical pattern occurs; everything else remains invariant.

Subsection 3.1 presents a simple two-level grammar which describes the above data. Subsection 3.2 presents error checking.

(4)

ARABIC VERBAL STEMS

Measure	Active	Passive
1	katab	kutib
2	kattab	kuttib
3	kaatab	kuutib
4	ʔaktab	ʔuktib
5	takattab	tukuttib
6	takaatab	tukuutib
7	nkatab	nkutib
8	ktatab	ktutib
9	ktabab	
10	staktab	stuktib
11	ktaabab	
12	ktawtab	
13	ktawwab	
14	ktanbab	
15	ktanbay	
Q1	daħraĵ	duħriĵ
Q2	tadaħraĵ	tuduħriĵ
Q3	dħanraĵ	dħunriĵ
Q4	dħarĵaĵ	dħurĵiĵ

3.1 Two-Level Rules

The lexical level maintains three lexical tapes (Kay, 1987; Kiraz, 1994): pattern tape, root tape and vocalism tape; each tape scans a lexical tree. Examples of pattern morphemes are: $\{c_1v_1c_2v_1c_3\}$ (M 1), $\{c_1c_2v_1nc_3v_2c_4\}$ (M Q3). The root morphemes are {ktb} and {dħrĵ}, and the vocalism morphemes are {a} (active) and {ui} (passive).

The following two-level grammar handles the above data. Each lexical expression is a triple; lexical expressions with one symbol assume ε on the remaining positions.

(5)

GENERAL RULES

$$\text{R0:}\quad \begin{matrix} * & - & X & - & * \\ * & - & X & - & * \end{matrix} \Rightarrow$$

$$\text{R1:}\quad \begin{matrix} * & - & (P_c, C, \varepsilon) & - & * \\ * & - & C & - & * \end{matrix} \Rightarrow$$

$$\text{R2:}\quad \begin{matrix} * & - & (P_v, \varepsilon, V) & - & * \\ * & - & V & - & * \end{matrix} \Rightarrow$$

where $P_c \in \{c_1, c_2, c_3, c_4\}$,
$P_v \in \{v_1, v_2\}$,

(5) gives three general rules: R0 allows any character on the first lexical tape to surface, e.g. infixes, prefixes and suffixes. R1 states that any P $\in$ {c_1, c_2, c_3, c_4} on the first (pattern) tape and C on the second (root) tape with no transition on the third (vocalism) tape corresponds to C on the surface tape; this rule sanctions consonants. Similarly, R2 states that any P $\in$ {v_1, v_2} on the pattern tape and V on vocalism tape with no transition on the root tape corresponds to V on the surface tape; this rule sanctions vowels.

(6)

BOUNDARY RULES

R3: $\begin{matrix}(B, \varepsilon, \varepsilon) & - & + & - & * & \Rightarrow \\ * & - & \varepsilon & - & * & \end{matrix}$

R4: $\begin{matrix}(B,*,*) & - & (+,+,+) & - & * & \Rightarrow \\ * & - & \varepsilon & - & * & \end{matrix}$

where B $\neq$ +

(6) gives two boundary rules: R3 is used for non-stem morphemes, e.g. prefixes and suffixes. R4 applies to stem morphemes reading three boundary symbols simultaneously; this marks the end of a stem. Notice that LLC ensures that the right boundary rule is invoked at the right time.

Before embarking on the rest of the rules, an illustrated example seems in order. The derivation of /dħunriĵa/ (M Q5, passive), from the three morphemes {$c_1c_2v_1nc_3v_2c_4$}, {dħrĵ} and {ui}, and the suffix {a} '3rd person' is illustrated in (7).

(7)

DERIVATION OF M Q3 + {a}

			u				i		+		*vocalism tape*
d	ħ				r		ĵ	+			*root tape*
c_1	c_2	v_1	n	c_3	v_2	c_4	+	a	+		*pattern tape*
1	1	2	0	1	2	1	4	0	3		
d	ħ	u	n	r	i	ĵ		a			*surface tape*

The numbers between the surface tape and the lexical tapes indicate the rules which sanction the moves.

(8)

SPREADING RULES

R5: $\begin{matrix}(P_1, C, \varepsilon)\cdots & - & P & - & * & \Rightarrow \\ * & - & C & - & * & \end{matrix}$

R6: $\begin{matrix}(v_1, \varepsilon, V)\cdots & - & v_1 & - & * & \Rightarrow \\ * & - & V & - & * & \end{matrix}$

where $P_1 \in \{c_2, c_3, c_4\}$

Resuming the description of the grammar, (8) presents spreading rules. Notice the use of ellipsis to indicate that there can be tuples separating LEX and LLC, as far as the tuples in LLC are the nearest ones to LEX. R5 sanctions the spreading (and gemination) of consonants. R6 sanctions the spreading of the first vowel. Spreading examples appear in (9).

(9)

DERIVATION OF M 1- M 3

a. /katab/ =

	a				+	*VT*
k		t		b	+	*RT*
c_1	v_1	c_2	v_1	c_3	+	*PT*
1	2	1	6	1	4	
k	a	t	a	b		*ST*

b. /kattab/ =

	a					+	*VT*
k		t			b	+	*RT*
c_1	v_1	c_2	c_2	v_1	c_3	+	*PT*
1	2	1	5	6	1	4	
k	a	t	t	a	b		*ST*

c. /kaatab/ =

	a					+	*VT*
k			t		b	+	*RT*
c_1	v_1	v_1	c_2	v_1	c_3	+	*PT*
1	2	6	1	6	1	4	
k	a	a	t	a	b		*ST*

The following rules allow for the different possible orthographic vocalisations in Semitic texts:

R7 $\begin{matrix}(\overline{V}, \varepsilon,\varepsilon) & - & (V, \varepsilon, \varepsilon) & - & (\overline{V}, \varepsilon, \varepsilon) & \Rightarrow \\ * & - & \varepsilon & - & * & \end{matrix}$

R8 $\begin{matrix}(P_{c1}, C1, \varepsilon) & - & (P, \varepsilon, V) & - & (P_{c2}, C2, \varepsilon) & \Rightarrow \\ * & - & \varepsilon & - & * & \end{matrix}$

R9 $\begin{matrix}\lambda & - & (v_1,\varepsilon,\varepsilon) & - & \rho & \Rightarrow \\ * & - & \varepsilon & - & * & \end{matrix}$

where $\lambda = (v_1,\varepsilon,V)\cdots(P_{c1},C1,\varepsilon)$ and $\rho = (P_{c2},C2,\varepsilon)$.

R7 and R8 allow the optional deletion of short vowels in non-stem and stem morphemes, respectively; note that the lexical contexts make sure that long vowels are not deleted. R9 allows the optional deletion of a short vowel what is the cause of spreading. For example the rules sanction both /katab/ (M 1, active) and /kutib/ (M 1, passive) as interpretations of ⟨ktb⟩ as showin in (10).

3.2 Error Rules

Below are outlined error rules resulting from peculiarly Semitic problems. Error rules can also be constructed in a similar vein to deal with typographical Damerau error (which also take care of the issue of

wrong vocalisms).

(10)

TWO-LEVEL DERIVATION OF M 1

a. /katab/ =

	a					+
k		t			b	+
c_1	v_1	c_2	v_1	c_3		+
1	8	1	9	1	4	
k		t		b		

VT *RT* *PT* *ST*

b. /kutib/ =

	u		i			+
k		t			b	+
c_1	v_1	c_2	v_1	c_3		+
1	8	1	9	1	4	
k		t		b		

VT *RT* *PT* *ST*

3.2.1 Vowel Shift

A vowel shift error rule will be tried with a partition on a (short) vowel which is not an expected (lexical) vowel at that position. Short vowels can legitimately be omitted from an orthographic representation - it is this fact which contributes to the problem of vowel shifts. A vowel is considered shifted if the same vowel has been omitted earlier in the word. The rule deletes the vowel from the surface. Hence in the next pass of (normal) analysis, the partition is analysed as a legitimate omission of the *expected* vowel. This prepares for the next shifted vowel to be treated in exactly the same way as the first. The expectation of this reapplication is allowed for in reap = y.

(11)

E0: X ⇒ ε **where** reap = y
{ [om_stmv,ε,(*,*,X)] ··· - * }

E1: X ⇒ ε **where** reap = y
{ [*,*,(v1,ε,X)] ··· [om_sprv,ε,(*,*,ε)] ··· - * }

In the rules above, 'X' is the shifted vowel. It is deleted from the surface. The partition contextual tuples consist of [RULE NAME, SURF, LEX]. The LEX element is a tuple itself of [PATTERN, ROOT, VOCALISM]. In E0 the shifted vowel was analysed earlier as an omitted stem vowel (om_stmv), whereas in E1 it was analysed earlier as an omitted spread vowel (om_sprv). The surface/lexical restrictions in the contexts could be written out in more detail, but both rules make use of the fact that those contexts are analysed by other partitions, which check that they meet the conditions for an omitted stem vowel or omitted spread vowel.

For example, *⟨dħruĵi⟩ will be interpreted as ⟨duħrij⟩. The 'E0's on the rule number line indicate where the vowel shift rule was applied to replace an error surface vowel with ε. The error surface vowels are written in italics.

(12)

TWO-LEVEL ANALYSIS OF *⟨dħruĵi⟩

	u			i				+	*VT*
d		ħ	r			ĵ		+	*RT*
c_1	v_1	c_2	c_3		v_2	c_4		+	*PT*
1	8	1	1	E0	8	1	E0	4	
d		ħ	r	*u*		ĵ	*i*		*ST*

3.2.2 Deleted Consonant

Problems resulting from phonetic syncopation can be treated as accidental omission of a consonant, e.g. *⟨mdītâ⟩, ⟨mdīntâ⟩.

(13)

E2: ε ⇒ X **where** cons(X),reap = n
{ * - * }

3.2.3 Deleted Long Vowel

Although the error probably results from a different fault, a deleted long vowel can be treated in the same way as a deleted consonant. With current transcription practice, long vowels are commonly written as two characters - they are possibly better represented as a single, distinct character.

(14)

E3: ε ⇒ XX **where** vowel(X),reap = n
{ * - * }

The form *⟨tuktib⟩ can be interpreted as either ⟨tukuttib⟩ with a deleted consonant (geminated 't') or ⟨tukuutib⟩ with a deleted long vowel.

(15)

TWO-LEVEL ANALYSIS OF *⟨tuktib⟩

a. M 5 =

	u						i			+	*VT*
		k			t				b	+	*RT*
t	v_1	c_1	v_1	c_2		c_2	v_2	c_3		+	*PT*
0	2	1	9	1	E2	1	2	1	4		
t	u	k		t		t	i	b			*ST*

b. M 6 =

	u						i			+	*VT*
		k				t			b	+	*RT*
t	v_1	c_1		v_1	v_1	c_2	v_2	c_3		+	*PT*
0	2	1	E3	6	6	1	2	1	4		
t	u	k		u	u	t	i	b			*ST*

3.2.4 Substituted Consonant

One type of morphographemic error is that consonant substitution may not take place before appending a suffix. For example /samaaʔ/ 'heaven' + {iyy} 'relative adjective' surfaces as ⟨samaawiyy⟩, where ʔ→ w in the given context. A common mistake is to write it as *⟨sammaʔiyy⟩.

(16)

E4: ʔ ⇒ w where reap = n
{ * - [glottal_change, w,(P_c,ʔ,ε)] }

The 'glottal_change' rule would be a normal morphological spelling change rule, incorporating contextual constraints (e.g. for the morpheme boundary) as necessary.

4 Broken Plurals, Diminutive and Deverbal Nouns

This section deals with morphosyntactic errors which are independent of the two-level analysis. The data described below was obtained from Daniel Ponsford (personal communication), based on (Wehr, 1971).

Recall that a Semitic stems consists of a root morpheme and a vocalism morpheme arranged according to a canonical pattern morpheme. As each root does not occur in all vocalisms and patterns, each lexical entry is associated with a feature structure which indicates *inter alia* the possible patterns and vocalisms for a particular root. Consider the nominal data in (17).

(17)

BROKEN PLURALS

Singular	Plural Forms
kadiš	kudš, *kidaaš
kaafil	kuffal, *kufalaaʔ, *kuffaal
kafiil	kufalaaʔ
sahm	*ʔashaam, suhuum, ʔashum

Patterns marked with * are morphologically plausible, but do not occur lexically with the cited nouns. A common mistake is to choose the wrong pattern.

In such a case, the two-level model succeeds in finding two-level analyses of the word in question, but fails when parsing the word morphosyntactically: at this stage, the parser is passed a root, vocalism and pattern whose feature structures do not unify.

Usually this feature-clash situation creates the problem of which constituent to give preference to (Langer, 1990). Here the vocalism indicates the inflection (e.g. broken plural) and the preferance of vocalism pattern for that type of inflection belongs to the root. For example *(kidaaš)would be analysed as root {kdš} with a broken plural vocalism. The pattern type of the vocalism clashes with the broken plural pattern that the root expects. To correct, the morphological analyser is executed in generation mode to generate the broken plural form of {kdš} in the normal way.

The same procedure can be applied on diminutive and deverbal nouns.

5 Conclusion

The model presented corrects errors resulting from combining nonconcatenative strings as well as more standard morphological or spelling errors. It covers Semitic errors relating to vocalisation, diacritics, phonetic syncopation and morphographemic idiosyncrasies. Morphosyntactic issues of broken plurals, diminutives and deverbal nouns can be handled by a complementary correction strategy which also depends on morphological analysis.

Other than the economic factor, an important advantage of combining morphological analysis and error detection/correction is the way the lexical tree associated with the analysis can be used to determine correction possibilities. The morphological analysis proceeds by selecting rules that hypothesise lexical strings for a given surface string. The rules are accepted/rejected by checking that the lexical string(s) can extend along the lexical tree(s) from the current position(s). Variables introduced by error rules into the surface string are then instantiated by associating surface with lexical, and matching lexical strings to the lexicon tree(s). The system is unable to consider correction characters that would be lexical impossibilities.

Acknowledgements

The authors would like to thank their supervisor Dr Stephen Pulman. Thanks to Daniel Ponsford for providing data on the broken plural and Nuha Adly Atteya for discussing Arabic examples.

References

Abduh, D. (1990). *ṣuʕūbat tadqīq ʔal-ʔimlāʔ ʔāliyyan fī ʔal-ʕarabiyyah* [Difficulties in automatic spell checking of Arabic]. In *Proceedings of the Second Cambridge Conference: Bilingual Computing in Arabic and English.* In Arabic.

Damerau, F. (1964). A technique for computer detection and correction of spelling errors. *Comm. of the Assoc. for Computing Machinery*, 7(3):171–6.

Goldsmith, J. (1976). *Autosegmental Phonology.* PhD thesis, MIT. Published as *Autosegmental and Metrical Phonology*, Oxford 1990.

Kay, M. (1987). Nonconcatenative finite-state morphology. In *Proceedings of the Third Conference of the European Chapter of the Association for Computational Linguistics*, pages 2–10.

Kiraz, G. Computational analyses of Arabic morphology. Forthcoming in Narayanan, A. and Ditters, E., editors, *The Linguistic Computation of Arabic*. Intellect. Article 9408002 in `cmp-lg@xxx.lanl.gov` archive.

Kiraz, G. (1994). Multi-tape two-level morphology: a case study in Semitic non-linear morphology. In *COLING-94: Papers Presented to the 15th International Conference on Computational Linguistics*, volume 1, pages 180–6.

Kiraz, G. and Grimley-Evans, E. (1995). Compilation of n:1 two-level rules into finite state automata. Manuscript.

Langer, H. (1990). Syntactic normalization of spontaneous speech. In *COLING-90: Papers Presented to the 14th International Conference on Computational Linguistics*, pages 180–3.

McCarthy, J. (1981). A prosodic theory of nonconcatenative morphology. *Linguistic Inquiry*, 12(3):373–418.

Pollock, J. and Zamora, A. (1983). Collection and characterization of spelling errors in scientific and scholarly text. *Journal of the American Society for Information Science*, 34(1):51–8.

Pulman, S. and Hepple, M. (1993). A feature-based formalism for two-level phonology: a description and implementation. *Computer Speech and Language*, 7:333–58.

Wehr, H. (1971). *A Dictionary of Modern Written Arabic.* Spoken Language Services, Ithaca.