17th European Conference on Artificial Intelligence (ECAI 06)
Brewka, Coradeschi, Perini, Traverso (Eds.)
pp. 829-830, ISBN 1-58603-642-4, ©IOS Press 2006.

Phonetic Spelling and Heuristic Search

Benno Stein[1] and **Daniel Curatolo**[2]

Abstract. We introduce a new approach to spellchecking for languages with extreme phonetic irregularities. The spelling for such languages can be significantly improved if knowledge about pronunciation and sound becomes the central part of the spelling algorithm. However, given a weak phoneme-grapheme-correspondence the standard spelling algorithms, which are rule-based or edit-distance-based, are severely limited in their phonetic capabilities.

A production approach to spelling can overcome the limitations—but suffers from its search space size. We describe in this paper the main building blocks to tackle this problem with heuristic search. Our ideas have been operationalized in the SMARTSPELL algorithm, with impressive results related to spelling correction and runtime.

1 INTRODUCTION AND PROBLEM SETTING

Orthography is the study or practice of correct spelling or writing; spelling is the choice of which letters or symbols to combine to represent a word. A (single word) spelling algorithm takes a dictionary D and a possibly misspelled word w as input and returns the most similar words $w_1, \ldots, w_k$ from D with respect to w. The "quality" or "power" of a spelling algorithm depends on the operationalized similarity measure, implicating an inevitable tradeoff between runtime performance and spelling quality.

The difficulty of a spelling problem depends on both the type of the spelling error to be detected and the underlying language. Table 1 illustrates common types of spelling errors with respect to the single word spelling problem.

Spelling error type	Example
Permutations or dropped letters	hpantom → phantom
Misremembering spelling details	recieve → receive, remembering believe
Trying to spell out pronunciation	tuleboks → toolbox

Table 1. A hierarchy of spelling errors with increasing problem complexity.

A language's key impact to spelling is governed by its *phonemicity*, which describes the extent to which spelling is a guide to pronunciation: A word is called phonemic if its spelling corresponds to its pronunciation. However, Trost pointed out that written language is rarely a true phonemic description [10]. Table 2, taken from an IEA survey [3], arranges languages with respect to their degree of phonemicity.

One of the many examples for the high irregularity of the English language is the word "school" where among others the following writings produce the same sound: skool, scool, scule, skule. The sources of irregularity are positional spelling, i. e., the sound of letters varies according to the position in a word, and polyvalence, i. e., letters can produce different sounds.

		Spelling system		
Highly regular <——		——————		——> Irregular
Finnish	Spanish, Portuguese, Italian, Hungarian, Slovenian	German, Dutch, Greek Swedish, Norwegian, Icelandic	Danish, French	English

Table 2. Languages ordered with respect to their degree of phonemicity [3].

Considering the sound of a written word within a spelling process is by far more complex than simply computing edit distances, since extra combinatorics is introduced at two places: (*i*) Which letters form a group that produces a single sound? (*ii*) Which sound is produced? This paper shows how heuristic search can be used to significantly speed up the similarity analysis in the huge search space. The operationalization of the presented ideas led to an efficient algorithm, SMARTSPELL, which is used in real-world applications.

The paper is organized as follows. The remainder of this section classifies possible solutions to the spelling problem, and Section 2 outlines the main ideas of the heuristic search behind SMARTSPELL.

1.1 Handling Misspelling

To classify existing work on spelling correction for single words we developed the taxonomy shown in Figure 1.

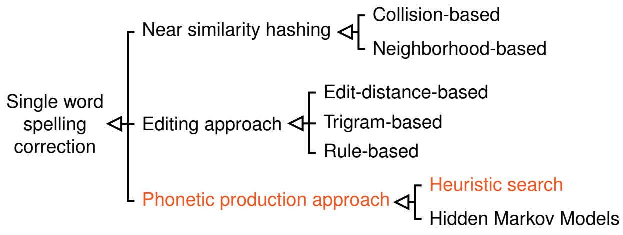

Figure 1. Classification of spelling correction methods.

At its top-level the taxonomy distinguishes three main approaches: (*i*) approaches that rely on the computation of hash keys for the words in D, (*ii*) editing approaches, which transform a word $w \notin D$ into a word from D, and (*iii*) phonetic production approaches, which construct a letter sequence that produces the observed sound.

The hashing approaches further subdivide into methods that exploit hash collisions (e. g. Soundex [6]) or evaluate neighborhoods in the list of the sorted hash keys (cf. SpeedCop [9] or [7]). Today, many approaches to handle misspelling are based on edit distances: The candidates for the correct spelling of some word w minimize the number of editing operations that transform w into some word of a dictionary D. The Levenshtein distance is a well-known measure to operationalize this idea; it associates each substitution, deletion, or insertion of a letter with certain cost [4]. This idea can be applied to trigrams [1] or become refined by phonetic transformation rules (cf. Aspell [2] or Correct [5]).

[1] Faculty of Media / Media Systems. Bauhaus University Weimar, Germany. benno.stein@medien.uni-weimar.de
[2] Art Systems Software GmbH. Paderborn, Germany.

2 PHONETIC SPELLING VIA SEARCH

We understand phonetic spelling as the generation of the most probable letter sequence with respect to phonological interpretation. Such a production approach may be realized with Hidden Markov Models or with heuristic search [8]; it is much more ambitious than an editing approach with respect to the explored search space. Note in this connection that editing approaches are too limited to be used as a guide to phonetic similarity. Examples:

- The Levenshtein distance between the words “eaecutiob / execution” and “aksekjushen / execution” is 2 and 8 respectively. Observe that the phonetic similarity in the second case is significantly higher than in the first case.
- The phonetically motivated edit rule “k ⟶ c” works fine to correct the misspelling “kat / cat”, but it fails in the case “cule / cool”.

The following three concepts—here applied to phonetic spelling—together form the elements of heuristic search:

1. *Search Space.* Defines all segmentations of all words in the dictionary D, along with operators to move between them.
2. *Evaluation Function* ϕ. Quantifies the quality of a segmentation of an unknown word with respect to the entire search space.
3. *Control Strategy.* Guides the exploration of the search space.

Search Space Since a hyphenation-based segmentation may not be optimum with respect to phonetics, the search space of SMARTSPELL contains for each word $w \in D$ the set Π_w of all possible segmentations, where each segment $s \in \pi_w$ in a segmentation $\pi_w \in \Pi_w$ is either a letter, 2-, 3-, or 4-gram. Hence, possible segmentations π_w for w = “execution” are:

e - x - e - c - u - t - i - o - n, ex - e - cut - i - on, e - x - ecu - tio - n

Altogether, “execution” has $|\Pi_w| = 773$ segmentations; a word of length n has $\sum_{i=1...4} seg(n-i)$ segmentations, with $seg(0) = 1$.

Evaluation Function ϕ $\phi(v, w, \pi_v, \pi_w)$ computes the similarity between two words, v, w, provided two segmentations, π_v, π_w; it is based on the phonetic a-priori similarity, $\varphi(s_i, s_j)$, for two segments s_i, s_j. φ is language-dependent and defines about 2500 segment similarities for the English language, among others:

s_i	s_j	$\varphi(s_i, s_j)$						
ac	ec	.95	ad	ed	.95	af	ef	.95
ac	ecc	.80	ad	edd	.90	af	eff	.90
ac	eck	.85	ad	eg	.57	af	es	.38

For the 24 essential sounds of RP English (received pronunciation) we developed a new concept to construct the function φ automatically: Based on so-called sound contexts, which are activated by the different consonant types like plosives, nasals, etc., raw estimates for the similarities are computed and statistically smoothed.

Let $\pi_v = s_{v,1}...s_{v,|\pi_v|}$ and $\pi_w = s_{w,1}...s_{w,|\pi_w|}$, then ϕ is defined as follows:

$$\phi(v, w, \pi_v, \pi_w) = \frac{\sum_{i=1}^{\min\{|\pi_v|,|\pi_w|\}} (|s_{v,i}| + |s_{w,i}|) \cdot \varphi(s_{v,i}, s_{w,i})}{|v| + |w|}$$

For example, the computation of ϕ for the two pairs of segmentations e-x-e-c-u-ti-o-n / a-ks-e-k-ju-sh-o-n and e-x-_-e-c-u-ti-o-n / a-k-s-e-k-ju-sh-o-n yields the following values:

		$\varphi(s_i,s_j)$			
e	a	.70	$\cdot$ 2	=	1.40
x	ks	.80	$\cdot$ 3	=	2.40
e	e	1.00	$\cdot$ 2	=	2.00
c	k	.95	$\cdot$ 2	=	1.90
u	ju	.75	$\cdot$ 3	=	2.24
ti	sh	.90	$\cdot$ 4	=	3.60
o	o	1.00	$\cdot$ 2	=	2.00
n	n	1.00	$\cdot$ 2	=	2.00
$\phi = 8.77/(9+11) \Rightarrow 88\%$					

e	a	.70	$\cdot$ 2	=	1.40
x	k	.30	$\cdot$ 2	=	.60
	s	0	$\cdot$ 2	=	0
e	e	1.00	$\cdot$ 2	=	2.00
c	k	.95	$\cdot$ 2	=	1.90
u	ju	.75	$\cdot$ 3	=	2.24
ti	sh	.90	$\cdot$ 4	=	3.60
o	o	1.00	$\cdot$ 2	=	2.00
n	n	1.00	$\cdot$ 2	=	3.00
$\phi = 7.87/(9+11) \Rightarrow 79\%$					

From all segmentations, Π_v, Π_w, for two words, v, w, one is interested in those that maximize ϕ; they are implicitly defined by ϕ^*:

$$\phi^*(v, w) = \max_{\pi_v \in \Pi_v, \pi_w \in \Pi_w} \phi(v, w, \pi_v, \pi_w)$$

ϕ fulfills the standard properties of a similarity measure; i. e., it is normalized, reflexive, and symmetric. In addition, it has a monotonic characteristic in word lengths: $|v| < |u| \Rightarrow \phi(w, wv) > \phi(w, wu)$

Control Strategy Checking the spelling of a word w regarding its phonetics means to identify a word $v \in D$ such that two segmentations $\pi_v \in \Pi_v$ and $\pi_w \in \Pi_w$ can be found that maximize ϕ. A typical value for $|D|$ is 10^6, a typical value for $|\Pi_w|, w \in D$, is $> 10^2$. Hence, the size of the search space for an ordinary spelling query is in $O(10^{10})$. To achieve an acceptable response time SMARTSPELL operationalizes a sophisticated control strategy that combines the following elements:

a) *Segmentation Heuristic.* Segmentations are constructed stepwise, striving for a minimum length difference of the residual strings.
b) *Early Pruning.* Exploitation of the monotonicity of ϕ for pruning.
c) *Iterative Deepening.* Candidates for a Depth-First Search are chosen from a pool whose pruning threshold is successively lowered.
d) *Nogood Construction.* Segments that are unlikely to match are stored in a special nogood table for φ.
e) *Memorization.* Strings with high ϕ-values are remembered.

Results Following examples illustrate the power of the phonetic production approach and the SMARTSPELL algorithm:

angenearing	⟶	engineering 92%
buysikel	⟶	physical 85%, bicycle 82%
dshungal	⟶	jungle 90%
hachhock	⟶	hedgehog 95%
jentelman	⟶	gentleman 85%
kompilayshon	⟶	compilation 89%
refridjeraitar	⟶	refrigerator 79%
siantifik	⟶	scientific 87%
tradishonell	⟶	traditional 93%
tshylt	⟶	child 97%
tulebogs	⟶	toolbox 93%

The function φ was constructed for English, German, and Spanish; moreover, SMARTSPELL has proven is usability in several real-word applications. On request we will provide Web-based access.

REFERENCES

[1] R. Angell, G. Freund, and P. Willett, ‘Automatic Spelling Correction Using a Trigram Similarity Measure’, *Information Processing and Management*, **19**(4), (1983).

[2] K. Atkinson. Aspell. aspell.sourceforge.net/, 2004.

[3] W. B. Elley, ‘How in the World do Students Read?’, Technical report, The Hague, International Association for the Evaluation of Educational Achievement (IEA), (1992).

[4] D. Gusfield, *Algorithms on Strings, Trees, and Sequencesy*, Cambridge University Press, 1997.

[5] B. Kessler. A spelling corrector incorporating knowledge of English orthography and pronunciation. www.artsci.wustl.edu/~bkessler/correct, 1998.

[6] D. E. Knuth, *The Art of Computer Programming, Volume 3: Sorting and Searching*, Addison-Wesley, 1973.

[7] K. Kukich, ‘Spelling correction for the Telecommunications Network for the Deaf’, *Com. of the ACM*, **35**(5), (1992).

[8] J. Pearl, *Heuristics*, Addison-Wesley, Massachusetts, 1984.

[9] J. Pollock and A. Zamora, ‘Automatic spelling correction in scientific and scholarly text’, *Com. of the ACM*, **27**(4), (1984).

[10] Harald Trost. Computational Morphology. www.ai.univie.ac.at/~harald/handbook.html, 2001.