Real-word spelling correction with trigrams: A reconsideration of the Mays, Damerau, and Mercer model

Amber Wilcox-O'Hearn, Graeme Hirst, and Alexander Budanitsky *

Department of Computer Science, University of Toronto
Toronto, Ontario, Canada M5S 3G4
amber, gh, abm@cs.toronto.edu

Abstract. The trigram-based noisy-channel model of real-word spelling-error correction that was presented by Mays, Damerau, and Mercer in 1991 has never been adequately evaluated or compared with other methods. We analyze the advantages and limitations of the method, and present a new evaluation that enables a meaningful comparison with the WordNet-based method of Hirst and Budanitsky. The trigram method is found to be superior, even on content words. We then show that optimizing over sentences gives better results than variants of the algorithm that optimize over fixed-length windows.

1 Introduction

Real-word spelling errors are words in a text that, although correctly spelled words in the dictionary, are not the words that the writer intended. Such errors may be caused by typing mistakes or by the writer's ignorance of the correct spelling of the intended word. Ironically, such errors are also caused by spelling checkers in the correction of non-word spelling errors: the "auto-correct" feature in popular word-processing software will sometimes silently change a non-word to the wrong real word (Hirst and Budanitsky 2005), and sometimes when correcting a flagged error, the user will inadvertently make the wrong selection from the alternatives offered. The problem that we address in this paper is the automatic detection and correction of real-word errors.

Methods developed in previous research on this topic fall into two basic categories: those based on human-made lexical or other resources and those based on machine-learning or statistical methods. An example of a resource-based method is that of Hirst and Budanitsky (2005), who use semantic distance measures in WordNet to detect words that are potentially anomalous in context — that is, semantically distant from nearby words; if a variation in spelling [1] results in a word that was semantically closer to the context, it is hypothesized that the original word is an error (a "*malapropism*")

* This research was supported by a grant from the Natural Sciences and Engineering Research Council of Canada. We are grateful to Bryce Wilcox-O'Hearn for his assistance.

[1] In this method, as in the trigram method that we discuss later, any consistent definition, narrow or broad, of what counts as the spelling variations of a word may be used. Typically it would be based on edit distance, and might also take phonetic similarity into account; see our remarks on Brill and Moore (2000) and Toutanova and Moore (2002) in section 5 below.

and the closer word is its correction. An example of a machine-learning method is that of Golding and Roth (1999), who combined the Winnow algorithm with weighted-majority voting, using nearby and adjacent words as features. (An extensive review of the prior research is given by Hirst and Budanitsky (2005), so we do not revisit it here. The problem of spelling correction more generally is reviewed by Kukich (1992).)

Typically, the machine learning and statistical approaches rely on pre-defined **confusion sets**, which are sets (usually pairs) of commonly confounded words, such as {*their, there, they're*} and {*principle, principal*}. The methods learn the characteristics of typical context for each member of the set and detect situations in which one member occurs in context that is more typical of another. Such methods, therefore, are inherently limited to a set of common, predefined errors, but such errors can include both content and function words. By contrast, the resource-based methods are not limited in this way, and can potentially detect a confounding of any two words listed in the resource that are spelling variations of one another, but these methods can operate only on errors in which both the error and the intended word are content words. The two methods are thus complementary; a complete system could use confusion sets to find common confounds and a resource-based method to look for other errors.

However, there is one method that is statistical and yet does not require predefined confusion sets: using word-trigram probabilities, which were first proposed for detecting and correcting real-word errors many years ago by Mays, Damerau, and Mercer (1991) (hereafter, *MDM*). Conceptually, the method is simple: if the trigram-derived probability of an observed sentence is lower than that of any sentence obtained by replacing one of the words with a spelling variation, then hypothesize that the original is an error and the variation is what the user intended.[2] In other words, relatively low probability of a sentence is taken as a proxy for semantic anomaly. Despite its apparent simplicity, the method has never, as far as we are aware, been applied in practice nor even used as a baseline in the evaluation of other methods. In this paper, we show why MDM's algorithm is more problematic than it at first seems, and why their published results cannot be used as a baseline. We present a new evaluation of the algorithm, designed so that the results can be compared with those of other methods, and then construct and evaluate some variations of the algorithm that use fixed-length windows.

2 The MDM Method and its characteristics

2.1 The Method

MDM frame real-word spelling correction as an instance of the noisy-channel problem: correcting the signal *S* (the observed sentence), which has passed through a noisy

[2] Trigram models have also been proposed for the simpler problem of correcting *non-word* spelling errors, most notably by Church and Gale (1991) and Brill and Moore (2000). Such models simply *presume* the presence of an error that has already been detected by another process (for example, by the failure of lexical look-up), and merely try to correct it within the trigram window. The real-word problem, by contrast, presumes the *absence* of an error, and the model is responsible not just for correcting errors but also for detecting them in the first place; this leads to considerations such as optimizing over sentence probabilities that have no counterpart in the simpler non-word trigram models. See also section 5 below.

channel (the typist) that might have introduced errors into it, by finding the most likely original signal S' (the intended sentence, generated by a language model). The probability that the typist types a word correctly is a parameter α, which is the same for all words.[3] A typical value for α could be .99. For each word, the remaining probability mass $(1-\alpha)$, the probability that the word is mistyped as another real word, is distributed equally among all its spelling variations.[4] So the probability that an intended word w is typed as x is given by

$$P(x\,|\,w) = \begin{cases} \alpha & \text{if } x = w \\ (1-\alpha)/|SV(w)| & \text{if } x \in SV(w) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $SV(w)$ is the set of spelling variations of word w (not including w itself).

The language model uses trigram probabilities; that is, the probability of an intended word w_i is given by $P(w_i\,|\,w_{i-1}w_{i-2})$, where $w_0 = w_{-1} = BoS$ (the beginning-of-sentence marker) and $w_{n+1} = w_{n+2} = EoS$ (the end-of-sentence marker). Thus the probability of an intended sentence $S' = w_1 \ldots w_n$ is given by

$$P(S') = \prod_{i=1}^{n+2} P(w_i\,|\,w_{i-1}w_{i-2}). \tag{2}$$

So given an observed sentence S, the corrected sentence S' is the one in the search space $\mathscr{C}(S) \cup \{S\}$ that maximizes the probability $P(S'\,|\,S) \propto P(S') \cdot P(S\,|\,S')$, where $P(S\,|\,S')$ is given by the model of the noisy channel, i.e., the typist, and the set $\mathscr{C}(S)$ of candidate corrections is the set of all sentences in which exactly one word in S has been replaced by one of its real-word spelling variations.

2.2 Discussion of the Method

MDM's method has an advantage over the resource-based "open-ended" methods in being able to detect errors in both content words and function words. But it also has the complementary disadvantage that effort is spent on errors that would also be found by a grammar checker (which would presumably be included in any writer's-aid system of which the spelling checker were a part), rather than concentrating on the errors that could not be thus detected. Another disadvantage is the size of the trigram model; a model covering a usefully large vocabulary might be impractically large. Data sparseness is also a serious problem: many correct trigrams that are observed will not occur in the model, even if it is built from a very large corpus.

An undesirable property of the method is that the likelihood that a real-word error x will be corrected depends on the number of spelling variations of the intended word w: the larger $SV(w)$ is, the smaller $P(w\,|\,x)$ is and hence the smaller the chance of correction

[3] No mention is made of words mistyped as non-words; but we can regard α as the probability that the word is either typed correctly or is typed as a non-word and then correctly amended.

[4] MDM refer to this as the word's confusion set; but unlike the confusion sets of, e.g., Golding and Roth (1999), it includes all spelling variations, not just those selected by a human as likely confounds.

is. This is a consequence of the division of the constant probability mass $(1-\alpha)$ among all members of $SV(w)$ in equation 1.

Because each member of $\mathscr{C}(S)$ contains exactly one changed word, the method is unable to correct more than one error per sentence. (Including in $\mathscr{C}(S)$ sentences with more than one change would be combinatorially explosive; but see section 4.2 below.) This limitation would usually not be a problem; that is, we expect that for most typists, α is considerably greater than the reciprocal of the mean sentence length, and so sentences would only very rarely contain more than one real-word error. Nonetheless, MDM seemingly violate their own assumption by considering typists with α values as low as .9 (one word in every ten is a real-word error); see section 2.3 below.

2.3 The Limitations of MDM's Evaluation

MDM's evaluation of their method used trigram probabilities for a 20,000-word vocabulary; they do not say what corpus the probabilities were derived from,[5] nor what smoothing method, if any, was used.[6] The test set was only 100 sentences, containing no words outside the 20,000-word vocabulary, chosen from newswire and English Canadian Hansard. For each sentence, a set of erroneous sentences was generated by replacing each word in turn with each of its possible spelling variations in the vocabulary; that is, each erroneous sentence contained exactly one error. There was an average of 86 erroneous sentences S for each original sentence S'.

In each set of sentences, each erroneous sentence was tested to determine whether, if it were observed, some other sentence in the set would be preferred, and if so whether that would be the original sentence; in addition, each original sentence was tested to see whether some erroneous variation would be preferred. The experiments were carried out with four different values of α, from .9 (an extremely error-prone typist) to .9999 (an extraordinarily accurate typist).

MDM did not present their results in terms of per-word accuracy or precision and recall, nor did they give the data necessary to calculate these values (true and false positives), so it is not possible to compare their results with other methods, such as those of Golding and Roth (1999) or Hirst and Budanitsky (2005), for which data are so presented. They do not include data on sentence lengths, and moreover, they classify their results according to *(a)* whether an erroneous sentence was detected as such and, if so, whether the appropriate correction was made, and *(b)* whether an actually correct sentence was wrongly selected for change. Thus, erroneous sentences in which the method incorrectly changes a true positive are conflated with those in which it chooses a false positive and a false negative. Hence only *per-sentence* accuracy, precision, and recall, incommensurate with other methods, can be derived from MDM's data; but in any case such measures are meaningless because of the extreme artificiality and bias of the test

[5] By a citation to Bahl, Jelinek, and Mercer (1983), MDM imply that the corpus they used was the IBM Laser Patent Corpus. But this cannot be so, as that corpus had a vocabulary of only 12,000 words (Bahl et al. 1978); and in any case trigram probabilities derived from such a corpus would be completely inappropriate for use with newswire and Hansard text.

[6] In their example data, MDM show the seemingly unlikely trigram *a submit that* as having a much higher probability than the trigram *what is happening*.

set. With the original sentences outnumbered by erroneous sentences 86 to 1, the number of false positives that are possible is extremely small compared to the number of true positives, with the consequence that per-sentence precision exceeds .99 in all cases and per-sentence recall varies from .618 for a very high value of α to .744 for a low value. Moreover, a model that performs well for MDM's test data may actually be prone to overcorrection in real data, which would translate into a loss of precision. There may be additional unpredictable effects of this bias too.

3 Re-evaluating the MDM Method

Because of these problems, we re-implemented and re-evaluated the MDM method in order to be able to make direct comparisons with other methods. As the original MDM data are not available, we followed Hirst and Budanitsky (2005) in using the 1987–89 Wall Street Journal corpus (approximately 30 million words), which we presume to be essentially free of errors. We reserved 500 articles (approximately 300,000 words) to create test data (see below). With the remainder of the corpus, using the CMU–Cambridge Statistical Language Modeling Toolkit (Clarkson and Rosenfeld 1997), we created a trigram model whose vocabulary was the 20,000 most frequent words in the corpus; all other words were mapped to the token *OOV* ("out of vocabulary"). We incorporated standard tokenization, and the Good–Turing smoothing and Katz backoff techniques of the toolkit.

To create more-realistic test sets, we automatically inserted real-word errors in the reserved articles by replacing one word in approximately every 200 with a random spelling variation — that is, we modeled a typist whose α value is .995; we chose this value simply to match the density of errors used by Hirst and Budanitsky (2005). And like both those authors and MDM, we defined a spelling variation to be a single-character insertion, deletion, or replacement, or the transposition of two characters that results in another real word. We created three test sets, each containing 15,555 sentences, which varied according to which words were candidates for replacement and for substitution:

T20: Any word in the 20,000-word vocabulary of the trigram model could be replaced by a spelling variation from the same vocabulary; this replicates MDM's style of test set.

T62: Any word in the 62,000 most frequent words in the corpus could be replaced by a spelling variation from the same vocabulary; this reflects real typing errors much better than **T20**.

Mal: Any content word listed as a noun in WordNet (but regardless of whether it was used as a noun in the text; there was no syntactic analysis) could be replaced by any spelling variation found in the lexicon of the *ispell* spelling checker; this replicates Hirst and Budanitsky's "malapropism" data.

Observe that in **T62** and **Mal**, the errors (the replacement words) are not limited to the vocabulary of the model. Thus one factor in our re-evaluation of the method is the adequacy of a 20,000-word vocabulary in the face of more-realistic data.

We ran our re-implementation of the MDM method with this data. Only test-data words that were in the 20,000-word vocabulary were candidates for correction, and

Table 1. Results of our replication of the MDM method on Wall Street Journal data with a 20,000-word vocabulary on three different test sets (see text for description), and the results of Hirst and Budanitsky (2005) on similar data (last row).

	Detection			Correction		
α	P	R	F	P	R	F
Test set **T20**:						
.9	.334	.847	.479	.327	.818	.467
.99	.574	.768	.657	.567	.747	.645
.995	.646	.736	.688	.639	.716	.675
.999	.794	.658	.719	.790	.643	.709
Test set **T62:**						
.9	.235	.537	.327	.229	.519	.318
.99	.447	.478	.462	.441	.466	.453
.995	.523	.460	.490	.517	.450	.481
.999	.693	.400	.508	.690	.395	.502
Test set **Mal**:						
.9	.145	.367	.208	.140	.352	.200
.99	.306	.320	.313	.299	.310	.304
.995	.371	.304	.334	.365	.296	.327
.999	.546	.261	.353	.543	.257	.349
Hirst and Budanitsky's best results (on **Mal**):						
–	.225	.306	.260	.207	.281	.238

words outside the vocabulary were mapped to *OOV* when determining trigram probabilities. We used four values of α, from .9 to .999, including the .995 value of the "typist" of our test data. We computed results in terms of per-word precision, recall, and F-measure, which we show separately for detection of an error and correction of an error; see Table 1.

The performance of the method is quite impressive. On the **T20** test set (all errors are in the vocabulary of the model) at $\alpha = .995$, which is perhaps the most realistic level, correction recall (the fraction of errors correctly amended) is .716 and correction precision (the fraction of amendments that are correct) is .639 ($F = .675$). On the **T62** test set (errors are not limited to the vocabulary of the model), performance naturally drops, but correction recall and precision are .450 and .517, respectively ($F = .481$), which is a level that would still be helpful to a user. Some examples of successful and unsuccessful corrections are shown in Table 2.

On the malapropism test set (all errors are in content words), the results are poorer; at $\alpha = .995$, correction recall is .296 and correction precision is .365 ($F = .327$). The difference between these results and those on **T62** shows that MDM's method performs better on function-word errors than on content-word errors. This is not surprising; intuitively, function-word errors are more likely to result in syntactic ill-formedness, and hence a much lower probability sentence, than the content-word errors. Nonethe-

Table 2. Examples of successful and unsuccessful corrections. Italics indicate observed word, arrow indicates correction, square brackets indicate intended word.

SUCCESSFUL CORRECTION:
Exxon has made a *loot* → lot [lot] of acquisitions of smaller properties, though the pace slowed last year after oil prices fell.
FALSE POSITIVE:
... Texaco's creditors *would* → could [would] breathe a sigh of relief ...
... the Conservative Party ... has been *last* → lost [last] in political polls.
FALSE NEGATIVE:
Like many schools, Lee's prospective kindergarten uses a readiness *teat* [test], designed to screen out children considered too immature.
TRUE POSITIVE DETECTION, FALSE POSITIVE CORRECTION:
"I'm uncomfortable *tacking* → talking [taking] a lot of time off work," he says.

less, these results are noticeably better than the best results of Hirst and Budanitsky's WordNet-based method, which achieved $F = .238$ on very similar data (last row of Table 1); in particular, the MDM method has superior correction precision.

4 Variations and Attempted Improvements on the MDM Method

4.1 A Better Language Model

Although MDM's method already does well compared to Hirst and Budanitsky's method, it is clear that it can be improved further. One obvious improvement is to increase the size of the language model. Table 3 shows that a 62,000-word model results in a large improvement over the 20,000-word model; for example, at $\alpha = .995$, correction F increases by 43% on test set **T62** and 45% on **Mal**. (Results on **T20** are roughly the same as before, of course; the slight reduction in performance is primarily due to the greater number of spelling variations that many words now have in the model.) The cost of the improvement is an increase in the size of the model from 17.9 million trigrams to 20.8 million. (Despite the exponential increase in the space of trigrams, the number actually observed in the corpus grows quite mildly.) Because of these results, we drop the 20,000-word model (and the **T20** test set) from further consideration.

4.2 Permitting Multiple Corrections

As we noted in section 2.2, the MDM algorithm can make at most one correction per sentence, because it would be combinatorially explosive to include sentences with more than one correction in the set $\mathscr{C}(S)$ of possible corrections of sentence S. We also noted that such an ability would, in any case, be of use only to very unskilled typists. Nonetheless, for the benefit of such typists, a possible method of making multiple corrections in a sentence while avoiding a combinatorial explosion is this: Instead of choosing the single sentence $S' \in \mathscr{C}(S) \cup S$ that maximizes the probability $P(S'|S)$, choose *all* sentences that give a probability exceeding that given by S itself, and then combine the

Table 3. Results of our replication of the MDM method on Wall Street Journal data with a 62,000-word vocabulary on three different test sets.

	Detection			Correction		
α	P	R	F	P	R	F
Test set **T20**:						
.9	.318	.828	.460	.311	.801	.448
.99	.532	.742	.619	.525	.724	.609
.995	.592	.708	.645	.587	.691	.635
.999	.738	.627	.678	.734	.614	.669
Test set **T62**:						
.9	.325	.846	.469	.318	.820	.458
.99	.544	.774	.639	.538	.758	.629
.995	.608	.750	.672	.603	.736	.663
.999	.756	.678	.715	.753	.667	.707
Test set **Mal**:						
.9	.212	.596	.313	.205	.571	.302
.99	.398	.536	.457	.390	.519	.445
.995	.459	.510	.483	.453	.497	.474
.999	.620	.444	.517	.616	.436	.510

corrections that each such sentence implies. (If conflicting corrections are implied then the one with the highest probability is chosen.) In other words, we apply all corrections that, taken individually, would raise the probability of the sentence as a whole, rather than only the single most probable such correction. It is important to note, however, the price that is paid here for avoiding the complete search space: The sentence that results from the combination of corrections might have a lower probability than others with fewer corrections — possibly even lower than that of the original sentence.

We experimented with this method using the 62,000-word model of section 4.1. We expected that the method would lead to improved correction only in the poor-typist condition where $\alpha = .9$ (one word in ten is mistyped). The results are shown in Table 4. Contrary to our expectations, despite an increase in recall compared to Table 3, F values were distinctly poorer for all values of α, *especially* the lower values, because the number of false positives went up greatly and hence precision dropped markedly. The number of sentences in which multiple corrections were hypothesized far exceeded the number of sentences with multiple errors; even for $\alpha = .9$ there were actually very few such sentences in the test data.

4.3 Using Fixed-Length Windows

The MDM method optimizes over sentences, which are variable-length and potentially quite long units. It is natural, therefore, to ask how performance changes if shorter, fixed-length units are used. In particular, what happens if we optimize a single word at a time in its trigram context? In this section, we consider a variation of the method that

Table 4. Results of the method permitting multiple corrections in the same sentence.

	Detection			Correction		
α	P	R	F	P	R	F
Test set **T62**:						
.9	.270	.869	.411	.263	.840	.400
.99	.505	.783	.614	.499	.765	.604
.995	.578	.756	.655	.573	.740	.646
.999	.739	.680	.708	.736	.668	.701
Test set **Mal**:						
.9	.179	.614	.277	.172	.586	.266
.99	.372	.543	.442	.364	.525	.430
.995	.437	.515	.473	.431	.502	.464
.999	.610	.448	.516	.605	.440	.510

optimizes over relatively short, fixed-length windows instead of over a whole sentence (except in the special case that the sentence is smaller than the window), while respecting sentence boundaries as natural breakpoints. To check the spelling of a span of d words requires a window of length $d+4$ to accommodate all the trigrams that overlap with the words in the span. The smallest possible window is therefore 5 words long, which uses 3 trigrams to optimize only its middle word.

Assume as before that the sentence is bracketed by two *BoS* and two *EoS* markers (to accommodate trigrams involving the first two and last two words of the sentence). The window starts with its left-hand edge at the first *BoS* marker, and the MDM method is run on the words covered by the trigrams that it contains; the window then moves d words to the right and the process repeats until all the words in the sentence have been checked.[7]

Observe that because the MDM algorithm is run separately in each window, potentially changing a word in each, this method as a side-effect also permits multiple corrections in a single sentence. In contrast to the method of section 4.2 above, the combinatorial explosion is avoided here by the segmentation of the sentence into smaller windows and the remaining limitation of no more than one correction per window. This limitation evaporates when $d = 1$, and the method becomes equivalent in its effect to that of section 4.2.

This, in turn, suggests a variation in which the window slides across the sentence, moving one word to the right at each iteration, overlapping its previous position, and then checking the words it contains in its new position. This would permit unrestricted

[7] If the number of words in the sentence is not an exact multiple of d, and the final window would contain no more than $d/2$ words, some preceding windows are enlarged to distribute these extra words; if the final window would contain more than $d/2$ but fewer than d words, then some preceding windows are reduced to distribute the extra space. For example, if $d = 5$ and the sentence is 22 words long, then the lengths of the windows are 6,6,5,5; if the sentence is 18 words long, then they will be 5,5,4,4.

Table 5. Results of adapting the MDM method to a fixed window of size $d+4$ that corrects d words.

	Detection			Correction		
α	P	R	F	P	R	F
Test set **T62**, $d=3$:						
.9	.275	.867	.418	.269	.838	.407
.99	.507	.783	.615	.501	.765	.605
.995	.579	.756	.656	.574	.740	.646
.999	.740	.680	.709	.737	.668	.701
Test set **Mal**, $d=3$:						
.9	.184	.614	.283	.177	.586	.272
.99	.373	.543	.442	.366	.525	.431
.995	.439	.515	.474	.432	.502	.465
.999	.611	.448	.517	.607	.440	.510

	Detection			Correction		
α	P	R	F	P	R	F
Test set **T62**, $d=6$:						
.9	.283	.864	.426	.276	.835	.415
.99	.512	.780	.618	.507	.762	.608
.995	.584	.755	.659	.579	.739	.649
.999	.743	.679	.710	.740	.668	.702
Test set **Mal**, $d=6$:						
.9	.188	.610	.287	.181	.583	.276
.99	.377	.541	.445	.370	.523	.433
.995	.442	.513	.475	.436	.500	.466
.999	.612	.446	.516	.607	.438	.509

	Detection			Correction		
α	P	R	F	P	R	F
Test set **T62**, $d=10$:						
.9	.292	.860	.436	.285	.832	.425
.99	.521	.780	.625	.515	.762	.615
.995	.593	.755	.664	.588	.739	.655
.999	.747	.679	.711	.744	.667	.703
Test set **Mal**, $d=10$:						
.9	.193	.609	.293	.186	.581	.282
.99	.384	.541	.449	.376	.524	.438
.995	.448	.514	.479	.442	.501	.470
.999	.614	.447	.518	.610	.439	.511

multiple corrections for values of d larger than 1, but at the price of rather more computation: If the sentence length is l words (plus the *BoS* and *EoS* markers), then $l-d+1$ iterations will be required to check the complete sentence instead of just $\lceil l/d \rceil$.[8]

We experimented with these methods for $d = 3$, 6, and 10, with the 62,000-word model. (We also tried $d = 1$, and verified that the results were identical to those of Table 4.) The performance of the simple fixed-window method is shown in Table 5. We observe that in most conditions, as with our first approach to multiple corrections, this method increases recall somewhat compared to the whole-sentence model (Table 3), but

[8] Some additional complexities arise in this method from the overlapping of the positions that the window takes. Except for the case when $d = 1$ (where this method becomes identical to the simple fixed-window method), words will be candidates for change in more than one window, with possibly conflicting results. We took a very simple approach: we never changed words in the middle of the analysis, and the opinion of the rightmost window always prevailed. For a discussion of the issues, see Wilcox-O'Hearn (2008).

precision drops markedly, especially for lower values of d and α, resulting in F values that are mostly poorer than, and at best about the same as, those of the whole-sentence model. Results are not shown for the sliding-window variation, whose performance in all conditions was the same as, or poorer than, the simpler method. We conclude that taking a unit of analysis smaller than the sentence is deleterious to the MDM method.

5 Related Work

As noted in footnote 2 above, noisy-channel trigram models have also been used in the simpler problem of non-word spelling correction. The emphasis in this work has generally been on the development of better channel models, i.e., better models of the typist. For example, at the level of keyboard errors, a substitution error involving keys that are adjacent on the keyboard is more likely than one involving two random keys; Church and Gale (1991) use complete character-based confusion matrices of typing errors. At the level of cognitive errors, the substitution of, for example, *a* for *e* is more likely (in English) in the context of *-ent* at the end of a word; Brill and Moore (2000) develop a model that accounts for this, which Toutanova and Moore (2002) extend to include phonetic similarity. Clearly, these channel models could also be used as the model of the typist in the MDM method; in equation (1), the probability mass $(1-\alpha)$ would be distributed among the spelling variations not equally but in accordance with their relative likelihood as given by the new model. We intend to do this in future work (Wilcox-O'Hearn 2008). However, such models will not account for errors introduced by miscorrection of non-word errors, for which our present equal-probability assumption is a better model.

The only other trigram-based method that we are aware of for real-word errors is that of Verberne (2002), who does not use (explicit) probabilities nor even localize the possible error to a specific word. Rather, her method simply assumes that any word trigram in the text that is attested in the British National Corpus (without regard to sentence boundaries!) is correct, and any unattested trigram is a likely error; when an unattested trigram is observed, the method then tries the spelling variations of all words in the trigram to find attested trigrams to present to the user as possible corrections. Her evaluation was carried out on only 7100 words of the Wall Street Journal corpus, with 31 errors introduced (i.e., a density of one error in every approximately 200 words, the same as used by Hirst and Budanitsky and the present study); she obtained a recall of .33 for correction and a precision of just .05 ($F = .086$).[9]

Since we began this research, Microsoft has released Office Word 2007, which includes a "contextual spelling checker" capable of detecting a number of real-word errors; the underlying method is proprietary and not disclosed. In future work, we will evaluate this system in comparison with the MDM model. An informal preliminary evaluation, with 5000 words of our **Mal** test data containing 25 errors, found a trade-off

[9] Verberne also tested her method on 5500 words of the BNC with 606 errors introduced (an average density of one word in nine) by inserting all possible instances from a pre-compiled list of 134 error types; this achieved correction recall of .68 and precision of .98. But this was a subset of her training data and the error density is quite unrealistic, so the results are not meaningful.

of low recall for high precision: Word 2007 found just 4 of the 25 errors and marked a fifth (*cation* for *nation*) as a non-word error, but it made no false-positive errors ($R = 0.2, P = 1.0, F = 0.33$).

6 Conclusion

We have shown that the trigram-based real-word spelling-correction method of Mays, Damerau, and Mercer is superior in performance to the WordNet-based method of Hirst and Budanitsky, even on content words ("malapropisms") — especially when supplied with a realistically large trigram model. Our attempts to improve the method with smaller windows and with multiple corrections per sentence were not successful. Rather, we found that there is little need for multiple corrections; indeed, the constraint of allowing at most one correction per sentence is useful in preventing false positives.

References

Bahl, Lalit R., J.K. Baker, P.S. Cohen, Frederick Jelinek, B.L. Lewis, and Robert L. Mercer 1978. Recognition of a continuously read natural corpus. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '78)*, Tulsa, vol. 3, 422–424.

Bahl, Lalit R., Frederick Jelinek, and Robert L. Mercer. 1983. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2), 179–190.

Brill, Eric and Moore, Robert C. 2000. An improved error model for noisy channel spelling correction. *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, Hong Kong, 286–293.

Church, Kenneth W. and William A. Gale. 1991. Probability scoring for spelling correction. *Statistics and Computing*, 1, 93–103.

Clarkson, Philip and Roni Rosenfeld. 1997. Statistical language modeling using the CMU–Cambridge Toolkit. *Proceedings of the 5th European Conference on Speech Communication and Technology (Eurospeech)*, Rhodes, 2707–2710.

Golding, Andrew R. and Dan Roth. 1999. A Winnow-based approach to context-sensitive spelling correction. *Machine Learning*, 34(1–3), 107–130.

Hirst, Graeme and Alexander Budanitsky. 2005. Correcting real-word spelling errors by restoring lexical cohesion. *Natural Language Engineering*, 11(1), March 2005, 87–111.

Kukich, Karen. 1992. Techniques for automatically correcting words in text. *Computing Surveys*, 24(4), 377–439.

Mays, Eric, Fred J. Damerau and Robert L. Mercer. 1991. Context based spelling correction. *Information Processing and Management*, 23(5), 517–522.

Toutanova, Kristina and Moore, Robert C. 2002. Pronunciation modeling for improved spelling correction. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, 144–151.

Verberne, Suzan. *Context-sensitive spell* [sic] *checking based on trigram probabilities*. Master's thesis, University of Nijmegen.

Wilcox-O'Hearn, L. Amber. 2008. *Applying trigram models to real-word spelling correction*. MSc thesis, Department of Computer Science, University of Toronto [forthcoming].