Natural Language Engineering **11** (1): 87–111. © 2005 Cambridge University Press
DOI: 10.1017/S1351324904003560 Printed in the United Kingdom

Correcting real-word spelling errors by restoring lexical cohesion

GRAEME HIRST and ALEXANDER BUDANITSKY
Department of Computer Science, University of Toronto, Toronto, Ontario, Canada M5S 3G4
e-mail: {gh,abm}@cs.toronto.edu

(*Received 26 August 2003; revised 29 December 2003*)

Abstract

Spelling errors that happen to result in a real word in the lexicon cannot be detected by a conventional spelling checker. We present a method for detecting and correcting many such errors by identifying tokens that are semantically unrelated to their context and are spelling variations of words that would be related to the context. Relatedness to context is determined by a measure of semantic distance initially proposed by Jiang and Conrath (1997). We tested the method on an artificial corpus of errors; it achieved recall of 23–50% and precision of 18–25%.

1 Real-word spelling errors

Conventional spelling checkers detect typing errors simply by comparing each token of a text against a dictionary of words that are known to be correctly spelled. Any token that matches an element of the dictionary, possibly after some minimal morphological analysis, is deemed to be correctly spelled; any token that matches no element is flagged as a possible error, with near-matches displayed as suggested corrections. Typing errors that happen to result in a token that is a correctly spelled word, albeit not the one that the user intended, cannot be detected by such systems. Such errors are not uncommon; Mitton (1987, 1996) found that "real-word errors account for about a quarter to a third of all spelling errors, perhaps more if you include word-division errors". *A fortiori*, it is now common for real-word errors to be *introduced* by auto-correction mechanisms[1] and by conventional spelling checkers

[1] An auto-correction mechanism watches out for certain pre-defined "errors" as the user types, replacing them with a "correction" and giving no indication or warning of the change. Such mechanisms are intended for undoubted typing errors for which only one correction is plausible, such as correcting *accomodate* to *accommodate*; deliberate misspellings (as in this footnote) are precluded. However, the 'AutoCorrect' feature of Microsoft Word contains by default many "corrections" for which other possibilities are also plausible. For example (in Microsoft Word v.X for Macintosh, with all updates to December 2003), *wierd* is changed to *weird*, although *wired* and *wield* are also plausible; *eyt* is changed to *yet*, although *eye* is plausible; and *Herat* is changed to *Heart*, although *Herat* is plausibly correct as it stands. Thus, a typing error that could have been subsequently detected by a spelling-checker may be replaced by a real-word error that can't be.

when the user carelessly accepts an incorrect recommendation by the system or inadvertently chooses the wrong correction from a menu. By contrast, a human proofreader, using linguistic and world knowledge, will usually notice an error of this kind because it will cause the text to be set somehow awry. If the erroneous token is of a different part of speech from that intended, then the sentence that it is in might not be parsable:

Example 1
The instrumental parts were recorded at different times and *them [then]* later combined on the master tape to produce special effects.

If the erroneous token is semantically unrelated to the intended one, then the sentence might not make sense:

Example 2
It is my sincere *hole [hope]* that you will recover swiftly.

Some typing errors will cause both conditions to occur:

Example 3
We all *hole [hope]* that you will recover swiftly.

And, of course, some errors result in a perfectly well-formed text, even if they produce a meaning other than that intended, and hence cannot be detected without knowledge or inference of the writer's intention:

Example 4
The committee is *now [not]* prepared to grant your request.

See Kukich (1992) or Mitton (1996) for an extensive survey of types of spelling errors and early approaches to the problem.

In this paper, we will discuss the detection and correction of errors that result in semantic anomaly, as in example 2. To distinguish these errors from the other kinds, we will refer to them, somewhat loosely, as *malapropisms*. Strictly speaking, a malapropism is an amusing substitution, due to ignorance and pretentiousness on the part of the writer or speaker, of one word for another of similar spelling or sound:[2]

Example 5
She has reached the *pinochle [pinnacle]* of success.

For our purposes in this paper, it is immaterial whether the cause of the error is ignorance, pretentiousness, poor typing, or careless use of a conventional spelling checker (and whether or not the error is cause for amusement). Our goal is thus

[2] The term is sometimes used even more loosely; for example, many of the spoken "malapropisms" attributed to George W. Bush, while possibly both amusing and due to ignorance, are actually non-word errors (*They misunderestimated me; This issue doesn't seem to resignate with the people*) or other kinds of linguistic or non-linguistic error (*Families is…where wings take dream; Our nation must come together to unite*); see `http://slate.msn.com/Features/bushisms/bushisms.asp`.

considerably broader than that of other recent work on real-word errors that aims simply at detecting occurrences of any of a small, pre-defined set of common errors; see section 7 below for discussion of this work.

2 Malapropisms as perturbations of cohesion

By their nature, naturally occurring coherent, meaningful texts contain many instances of the mechanisms of *linguistic cohesion*, such as word repetitions, coreference, and sets of semantically related words (Halliday and Hasan 1976; Hoey 1991). A coherent text will naturally refer to various concepts that are related to its topic or topics and hence are related to one another. The recurrence in a text of lexemes related to a particular concept is often characterized metaphorically as a 'chain' of words running through the text, linked by lexical and semantic relationships such as literal repetition, coreference, synonymy, and hyponymy (Halliday and Hasan 1976; Morris and Hirst 1991; Hoey 1991). A coherent text will have many such *lexical chains*, each running through part or all of the text and pertaining to some aspect of the topic or topics of the text; and, conversely, most content words of the text will be members of one or more chains. Because they are indicative of the structure and content of a text, lexical chains have been applied in computational linguistics for tasks such as text segmentation (Morris and Hirst 1991; Okumura and Honda 1994), lexical disambiguation (Okumura and Honda 1994), automatic creation of hypertext (Green 1999), and text summarization (Barzilay and Elhadad 1999; Silber and McCoy 2002); see Budanitsky (1999) for a detailed review. However, it remains an open research question as to just what kinds and degrees of semantic relationship should qualify as links for a lexical chain; we discuss this issue in a separate paper (Budanitsky and Hirst submitted).

A malapropism is a perturbation of the cohesion (and coherence) of a text. By definition, it is semantically inappropriate in its immediate context, and is probably therefore also semantically inappropriate in the broader context of the text itself. It is therefore unlikely that a malapropism can be linked into any of the lexical chains of the nearby text; it will probably bear no semantic relationship to any other word in the text.[3] On the other hand, it is likely (though not assured) that the intended word *would* fit into a lexical chain in the text. Thus the problem of detecting and correcting malapropisms can be cast as the problem of detecting tokens that fit into no lexical chain of the text and replacing them with words for which they are plausible mistypings that do fit into a lexical chain.

This idea was first tried by Hirst and St-Onge (1998), who reported modest success; while the system's performance in both detecting and correcting malapropisms was

[3] There are two qualifications to this statement. First, the malapropisms that we are considering are primarily performance errors – slips in typing. So if the malapropism is instead a competence error and is repeated within the text – consistently typing *pinochle* for *pinnacle*, for example, in the belief that the former is the correct spelling of the latter – then the malapropisms will form a lexical chain by their repetition. Such a text would be incoherent but cohesive, and the methods to be discussed below will not apply. Second, there is actually a mild cognitive bias in performance errors to words that are indeed related to the intended word or its context (Fromkin 1980), but we ignore this effect here.

well above baseline, it was nonetheless prone to far too many false alarms: for every true malapropism that it found, it would flag about ten other tokens that were not malapropisms at all. It was especially vulnerable to confusion by proper names and by common words of minimal topical content; for example, it suggested that *year* was a mistyping of *pear* in the context of *Lotus Development Corporation*, because *lotus* and *pear* are both hyponyms of *fruit* and hence could form a lexical chain. One of the serious problems underlying the system was an inadequate account of semantic relatedness in its construction of lexical chains. Two non-identical lexemes were considered to be related if and only if a short path of an allowable shape could be found between their synsets in the noun portion of WordNet. Paths could follow any of the WordNet relationships. (The details of 'allowable shape' and requisite shortness are not necessary here; the interested reader may see Hirst and St-Onge (1998).)

3 A new algorithm for detecting and correcting malapropisms

We have developed a new algorithm for malapropism detection and correction that, like Hirst and St-Onge's, is based on the idea of detecting and eliminating perturbations of cohesion in text. However, the new algorithm does not use lexical chains per se; rather, it treats a text as a bag of words (more precisely, as a list of paragraph-sized bags of words). Forgoing the chain structures enables the search to be bidirectional instead of left-to-right and to wrap around from the end of the text to the start, thereby recognizing the potential cohesion between introduction and conclusion. In addition, the measure of semantic relatedness that the algorithm employs can be varied independently; the scope of search is an additional parameter; distances in the text are measured in paragraphs rather than sentences; and disambiguation of words may be only partial. The new algorithm also includes proper-name recognition and addresses problems of ambiguity in inflectional morphology.

In accordance with the discussion above, the algorithm makes the following assumptions:

- A real-word spelling error is unlikely to be semantically related to the text.
- Usually, the writer's intended word will be semantically related to nearby words.
- It is unlikely that an intended word that is semantically unrelated to all those nearby will have a spelling variation that *is* related.

In addition, the algorithm requires the definition of two independent mechanisms. First, it requires a mechanism that, given a word (or any string), returns a list of all the words in the lexicon for which that word is a plausible misspelling – its *spelling variations*. Such a mechanism can be found in any conventional spelling checker. In the system to be described in section 4, we define the spelling variations of a word *w* to be those words in the lexicon that are derived from *w* by the insertion, deletion, or replacement of a single character, or the transposition of two adjacent characters. However, broader or narrower definitions are possible. For example, one might allow only substitutions of characters that are close to one another on the keyboard (Al-Mubaid and Truemper 2004), or take into account the probability of each particular typing error, using the data of Kernighan, Church and Gale (1990). At the risk of becoming dialect-dependent, the definition might permit homophones and other phonetic near-matches such as *kettle–cattle* and *pour–poor* (Al-Mubaid and Truemper 2004).[4]

Second, the algorithm requires a mechanism that, given two words, determines whether or not those words are *semantically related* (or *semantically close*). It's important to observe that semantic relatedness is not just similarity; similar entities are usually assumed to be semantically related by virtue of their likeness (*bank–trust company*), but dissimilar entities may also be semantically related by lexical relationships such as meronymy (*car–wheel*) and antonymy (*hot–cold*), or just by any kind of functional relationship or frequent association or co-occurrence of ideas (*soap–wash, penguin–Antarctica*). Here, we require relatedness in the broadest sense – pertaining to or associated with the same topic. Nonetheless, taking relatedness too broadly will result in failing to detect malapropisms; they will be spuriously found to be related to their context. We will discuss a constrained measure of semantic relatedness in section 4.1. A more-general discussion of such measures and the theoretical issues that they raise is given by Budanitsky (1999) and Budanitsky and Hirst (submitted).

In outline, the algorithm for detecting and correcting malapropisms is as follows: Words are (crudely) disambiguated where possible by accepting senses that are semantically related to possible senses of other nearby words. If all senses of any open-class, non–stop-list word that occurs only once in the text are found to be semantically unrelated to accepted senses of all other nearby words, but some sense of a spelling variation of that word is related (or is identical to another token in the context), then it is hypothesized that the original word is an error and the variation is what the writer intended; the user is warned of this possibility. For example, if no nearby word in a text is related to *diary* but one or more are related to *dairy*, it is suggested to the user that it is the latter that was intended. The exact window size implied by "nearby" is a parameter to the algorithm.

[4] Some commercial spelling checkers are extremely liberal in their notion of spelling variation, allowing multiple insertions, deletions, and transpositions – a strategy that taken to extremes could propose any word for any other. For example, the spelling checker in Microsoft Word, given a list of uncommon names, suggests implausible changes such as these: *Procopia* to *Porkpie* or *Preoccupied*, *Prunella* to *Runnels*, and *Philena* to *Phalanx* or *Hyena*. (This is in contrast with the auto-correction mechanism in the same software, whose definition of spelling variation is much too narrow; see footnote 1.) Overly broad definitions will reduce the precision of our algorithm, as it becomes more likely that some spelling variation will be wrongly preferred to the original word – see section 6. Nonetheless, in practical use, the definition of spelling variation used with the algorithm should be the same as that used with any associated non-word spelling corrector or auto-correct mechanism so that the errors that they make can be undone.

Pedler (2001a, 2001b) studied the performance of four spelling checkers on a corpus of writing by dyslexics (in which the error rate was greater than one word in every five) and found that while 47% of the errors involved more than one addition, deletion, or substitution, Word's broad definition of spelling variation gave it no practical advantage over other spelling checkers with more-conservative definitions.

0a. Look for non-word errors in the text, and make corrections (with help from user).

0b. Identify for consideration all words in the text that are in the lexicon but are not on the stop-list nor used as (part of) the name of a named entity.

1. Mark a word as *confirmed* if it occurs more than once in the text, if it occurs in the text as part of a known phrase, or if, within a window of n paragraphs (wrapping around to the start or end of the text if necessary), there are one or more words with a sense that is semantically related to at least one sense of the word under consideration. When a word is confirmed, remove from consideration all of its senses that were not involved in its confirmation.
2. If an unconfirmed word w (a *suspect*) has a spelling variation w' that would have been confirmed if it had appeared in the text instead of w, alert the user to the possibility that w' was intended where w appears (*raise an alarm*).

Fig. 1. Algorithm for malapropism detection and correction.

A statement of the algorithm is given in Figure 1. We now explain each step in detail.

Step 0: Preprocessing

Steps 0a and 0b of the algorithm are preprocessing. The first substep is correction of non-word spelling errors (perhaps by a conventional spelling checker). This should occur before malapropisms are sought (rather than after or in parallel), in order to maximize the number of words in the text available to check for semantic relatedness to each word considered by the algorithm. Moreover, as we observed earlier, it is not unusual for malapropisms to be *introduced* during conventional spelling checking, so malapropism detection should follow that – but see also our remarks in section 8 on integration of the two processes.

The second substep identifies words in the document that are to be checked for error by removing from further consideration those that are not in the lexicon at all and those that are on a stop-list. The algorithm, by its nature, applies only to words whose meaning or meanings are known and have content that is likely to be topical. We therefore exclude closed-class words and common non-topical words. Closed-class words are excluded as their role in a text is almost always purely functional and unrelated to content or topic. It is of course possible that a typing error could turn a highly contentful word into a closed-class word or vice versa; but the former case will not be considered by the algorithm and the latter will be considered but not detected. The exclusion of 'untopical' open-class words, such as *know, find*, and *world*, is well-precedented in information retrieval. Here, there is a trade-off between making the list as short as possible, in order to let as many words as possible be checked, and making the list as long as possible in order to avoid spurious relationships, such as the *year–pear–Lotus* example mentioned above.

Step 1: Suspicion

The first step of the algorithm itself is to confirm as correct any word found to be related to at least one other word in the text. This relationship can be identity – another occurrence of a word with the same lemma – or it can be a semantic relationship to another word, as discussed above. In searching for an identical token, the entire text is scanned; but in searching for a semantically related word, the *scope* of the search may be limited to words that are physically not too far away. The rationale for this is that in a large text with many topics, there is too high a chance of finding a spurious semantic relationship between a malapropism and a word in some other part of the text on a different topic; but this is less likely in the case of identity. (Pollock and Zamora (1983) found that, with the exception of a handful of frequently misspelled words, misspellings rarely tend to be repeated in a document.) Of course, the risk of finding a spurious relationship depends on the nature and length of the text, and different kinds of text could be treated differently. (In the system to be described in section 4, we experimented with search scopes ranging from a single paragraph to the entire text of a newspaper article.)

When the word under consideration has more than one sense, semantic relationships are sought between all its senses and all the senses of other words in the search scope. If any relationships are found between the word under consideration and any others, then only the senses that participate in those relationships are retained for subsequent searches. Thus words are, rather roughly, disambiguated or at least partially disambiguated. For example, if relationships are sought for (senses of) the word *file*, and a relationship to the tool sense of the word *plane* is the only one found, then only the tool sense of *file* will be retained.

In addition to identity and semantic relatedness, we follow St-Onge's (1995) intuition that the probability of accidentally forming a multiword compound that can be found in the lexicon (e.g. *abdominal cavity*, *chief executive officer*, *automated teller machine*, *withdrawal symptom*) is so low that the words of any such phrase occurring in the text can be regarded as mutually confirming.

Any word that cannot be confirmed in this step thus appears unrelated to its context, and might therefore be a real-word spelling error. We refer to such words as *suspects*; but it should be understood that this, by itself, is not sufficient cause to flag the word as a likely malapropism. It is not at all unusual for a text to contain such words, especially if the search scope – the context – is limited to a single paragraph or little more than that.

Step 2: Detection

To determine whether a suspect is a likely real-word spelling error, we look for positive evidence: a spelling variation of the suspect that would fit better into context than the suspect itself does. We therefore generate all spelling variations and for each one, attempt to confirm it as in Step 1. If at least one spelling variation is confirmed, then we take this as indicating that the variation is a better fit and hence more likely to be the intended word. The user is then alerted to this possibility.

4 A system for detecting and correcting malapropisms

We have built and evaluated a prototype system to detect and correct malapropisms by means of the algorithm above. In this section, we explain the components of the system, and in the following section, we describe an evaluation of the system.

4.1 Semantic relatedness measure

We tried five different measures of semantic relatedness in our system, all of which rely on a WordNet-like hierarchical thesaurus (Fellbaum 1998) as their lexical resource. The measures were those of Hirst and St-Onge (1998), Jiang and Conrath (1997), Leacock and Chodorow (1998), Lin (1997, 1998), and Resnik (1995). By comparing the performance of the different measures, which varied widely, we were able to study theories of semantic relatedness, and we describe this work in a separate paper (Budanitsky and Hirst submitted). Because these issues are orthogonal to malapropism detection, we report here only our experiments with the best-performing measure, which was that of Jiang and Conrath.

Jiang and Conrath's (1997) measure of semantic relatedness (strictly speaking, of semantic distance, the inverse of relatedness) is based on both the hierarchical structure of a taxonomy and the information content (IC) of its nodes. Given a node c in the hierarchy (a synset in the case of WordNet), the information content of c is the negative logarithm of the probability $\mathrm{p}(c)$ of encountering an *instance* of concept c in a corpus – that is, any lexeme that maps to c (the words of the synset) or its hyponyms. Then the relatedness of two lexemes that map to nodes c_1 and c_2 in the hierarchy is computed from the information content of those nodes and that of their *lowest superordinate* (or *most specific subsumer*), $lso(c_1, c_2)$, the lowest node in the hierarchy that is an ancestor to both. Specifically, Jiang and Conrath define the semantic distance between a child-node c and its parent-node $par(c)$ as:

$$\mathrm{dist_{JC}}(c, par(c)) = \mathrm{IC}(c \mid par(c)) = \mathrm{IC}(c) - \mathrm{IC}(par(c)).$$

Then the semantic distance between two arbitrary nodes c_1 and c_2 is the sum of the child-parent distances along the shortest path that connects them, $\mathrm{path}(c_1, c_2)$. Let $N(c_1, c_2)$ be the set of nodes in $\mathrm{path}(c_1, c_2)$, including c_1 and c_2 themselves. Then we have:

$$\begin{aligned} \mathrm{dist_{JC}}(c_1, c_2) &= \sum_{c \in N(c_1,c_2) \setminus lso(c_1,c_2)} \mathrm{dist_{JC}}(c, par(c)) \\ &= \mathrm{IC}(c_1) + \mathrm{IC}(c_2) - 2 \times \mathrm{IC}(lso(c_1, c_2)) \\ &= 2\log(\mathrm{p}(lso(c_1, c_2))) - (\log(\mathrm{p}(c_1)) + \log(\mathrm{p}(c_2))). \end{aligned} \tag{1}$$

Observe that, as a special case, the distance between two words in the same synset is zero. For a detailed explication and interpretation of the derivation and justification of Jiang and Conrath's measure, see Budanitsky (1999).

For example, in WordNet 1.5,[5] the concepts number ('a sum or total or indefinite quantity of units or individuals') and limit/bounds/boundary ('the greatest possible

[5] We began this work with WordNet 1.5, and stayed with this version despite newer releases in order to maintain strict comparability. Our experiments were complete before WordNet 2.0 was released. See section 6 for further comments.

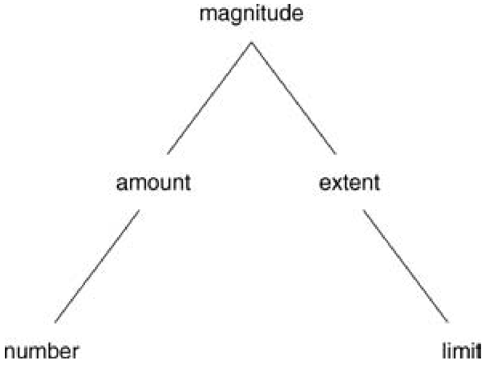

Fig. 2. Shortest path from number to limit in WordNet 1.5.

extent or degree of something') are related through their *lso*, magnitude ('relative size or extent'), as shown in Figure 2. The probability of encountering an instance of these concepts in the Brown Corpus (see below) is, respectively, 1.746986×10^{-3}, 9.889191×10^{-4}, and 3.748222×10^{-2}, implying respective information content of 9.160916, 9.98186, and 4.73765. Hence $\text{dist}_{\text{JC}}(\text{number}, \text{limit}) = 9.160916 + 9.98186 - 2 \times 4.73765 = 9.667477$.

Corpus Following Resnik (1995), we obtained information content values for each node in WordNet from frequency counts of words in the complete, untagged Brown Corpus. In their original experiments, Jiang and Conrath used SemCor (Miller, Leacock, Tengi and Bunker 1993), a sense-tagged subset of the Brown Corpus. Choosing the Brown Corpus over SemCor essentially means trading away accuracy for size, but, like Resnik, we believe that using a non-disambiguated corpus constitutes a more general approach. The availability of disambiguated texts such as SemCor is highly limited, due to the fact that automatic sense-tagging of text remains an open problem and manual sense-tagging of large corpora is prohibitively labor-intensive. On the other hand, the volume of raw textual data in electronic form is steadily growing.

Calibrating the measure Because the Jiang–Conrath function returns a numerical measure of distance on an essentially arbitrary scale, and not the boolean *related–unrelated* judgment required by the malapropism-detection algorithm, we needed to set a threshold distance below which two lexemes would be deemed close enough to be related. We did this by calibrating the measure against human judgments of semantic relatedness.

The data that we used were obtained and published by Rubenstein and Goodenough (1965), who asked 51 human subjects to make "synonymy judgments" on 65 pairs of words. The pairs ranged from "highly synonymous" (*gem–jewel*) to "semantically unrelated" (*noon–string*). Subjects were asked to rate them on the scale of 0.0 to 4.0 according to their "similarity of meaning" and ignoring any other observed semantic relationships (such as in the pair *journey–car*). Rubenstein and Goodenough's results are shown in Figure 3; the *y*-axis shows average similarity

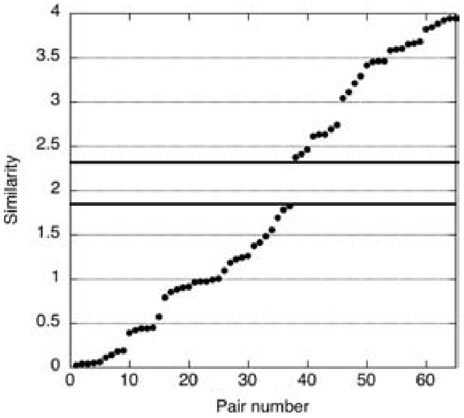

Fig. 3. Human judgments of similarity on 65 pairs of words, in order of increasing similarity (Rubenstein and Goodenough 1965).

rating, and the *x*-axis shows pairs enumerated by increasing similarity. Observe the broad gap between pair numbers 37 and 38 that separates the pairs into a more-similar group and a less-similar group.

To calibrate the Jiang–Conrath measure with this data, we evaluated each of the 65 Rubenstein–Goodenough pairs with the measure. The correlation between the measure and the human similarity judgments was —0.781. (The correlation is negative because semantic distance is inversely related to similarity.) We therefore set the Jiang–Conrath measure's threshold of relatedness at the point at which it best separates the two Rubenstein–Goodenough groups.

4.2 Other components of the system

Lexicon and thesaurus Because the Jiang–Conrath measure requires a hierarchical thesaurus, we used the noun portion of WordNet 1.5 as our lexicon and thesaurus. However, we did not limit use of the system to nouns in the text; semantic relatedness is independent of part of speech, and the algorithm applies to any non–stop-list word found in the lexicon. For our prototype system, we relied on the fact that, in English, verb lemmas and adjectives that are orthographically identical to a noun are almost always closely semantically related to that noun and so it was immaterial to our algorithm what the actual part of speech of the token was.[6] For example, for the sentence *Nadia hoped for a miracle*, we use the noun synset for *hope* even though it occurs as a verb in the sentence. While this leads to an obvious limitation on the prototype system – it simply cannot deal with words that do not have a noun form – there is no reason to think that this will not be resolved by the advent of a more-integrated hierarchical thesaurus that connects all parts of speech, such as the

[6] There are exceptions to this heuristic, of course. For example, the verb *to spell* 'to set down the orthographic form of a word' is not related to the noun *spell* 'magic incantation' or 'period of time'.

wordnets of EuroWordNet (Vossen 1998), the recently released WordNet 2.0, or an on-line *Roget*-structured thesaurus. See section 6 for additional discussion.

Stop-list and proper name recognition We use St-Onge's (1995) stop-list of 221 closed-class and high-frequency words, which is rather small compared to the lists used in systems for information retrieval and other applications in natural language processing. Thus we have opted for wider coverage over higher precision.

Proper names are filtered out by a module based on a lexical analyzer that was generously made available to us by Dekang Lin and Nalante Inc.

Spelling variations To generate spelling variations in Step 2 of the algorithm, we used code from *ispell*, an open-source spelling checker for non-word errors,[7] whose definition of spelling variation is as described in section 3: a single insertion, deletion, replacement, or transposition.

Lemmas and surface forms Although different occurrences of the same word in a text are recognized through their having the same lemma, the original surface forms must also be stored in order to generate spelling variations. For example, if the lemma *lie* surfaces in the text as the token *lain*, its spelling variations are *gain*, *lair*, *loin*, *lawn*, *plain*, etc., and not *die*, *lee*, *life*, *lieu*, or *pie*.

5 Evaluation of the system

5.1 Test data

To test the algorithm, we need a sufficiently large corpus of malapropisms in their context, each identified and annotated with its correction. Since no such corpus of naturally occurring malapropisms exists, we created one artificially. Following Hirst and St-Onge (1998), we took 500 articles from the 1987–89 *Wall Street Journal* corpus, with lengths ranging from 90 to 2763 tokens (an average of just over 600 words each), and replaced one word in every 200 with a spelling variation. To be a candidate for replacement, a word had to be present in our lexicon (see section 4.2), have at least one spelling variation that was also in the lexicon, and not be a stop-list word or proper noun. The corpus contained 107,233 such words, of which 1408 (1.31%) were replaced by malapropisms – an average of 2.8 malapropisms per article. In 19 articles that contained few malapropizable words, no word was replaced; these articles were removed from the data. We generated the spelling variations with the same code from *ispell* that we used in the implementation of the algorithm. (This does not lead to circularity, but rather to a consistent definition of what constitutes a spelling variation.)

In evaluating the system, we tried four different search scopes in determinations of semantic relatedness: just the paragraph containing the target word (scope = 1),

[7] *ispell* is a program that has evolved in PDP-10, Unix, and Usenet circles for more than 20 years, with contributions from many authors. Principal contributors to the current version include Pace Willisson and Geoff Kuenning.

that paragraph plus one or two adjacent paragraphs on each side (scope = 3 and 5), and the complete article (scope = MAX).

The baseline algorithm for malapropism detection is random choice ("chance"), flagging words as real-word spelling errors in the same proportion as they are expected to occur in the data. In addition, we compare our results to those of Hirst and St-Onge (1998).

5.2 Example results

In this section, we give examples of situations in which the algorithm succeeded and those in which it failed. In the subsequent section, we analyze the results quantitatively.

The malapropism in the following example was detected and corrected in all search scopes:

Example 6
Maybe the reasons the House Democrats won't let the contras stand and fight for what they believe in is because the Democrats themselves no longer stand and fight for their beliefs. The House's liberals want to pull the plug on the rebels but, lacking the courage to hold a straight up or down vote on that policy and expose its consequences to the U.S. electorate, they have to disguise their *intension [intention]* as a funding "moratorium."

No relationship was found between *intension* and any other word in the search scope; *intention* was the only possible spelling variation, and it was found to be related to *reason, want, policy, vote, stand,* and *belief.*

This malapropism was detected, but in some conditions was wrongly corrected:

Example 7
American Express says... it doesn't know what the *muss [fuss]* is all about.

Although no connections could be found for *muss* in most search scopes, connections were found not only for *fuss* but also for other spelling variations, such as *mass* (connected to *number* in an adjacent sentence) and *mugs* (in the sense of gullible people, connected to *group* nearby). And with scope = 1, *fuss* itself was not among the candidate corrections but *mugs* was.

The algorithm failed completely on this example:

Example 8
Mr. Russell argues that usury *flaw [law]* depressed rates below market levels years ago . . .

The word *flaw* was found to be related to the word *state* in a nearby sentence (*flaw* IS-A *imperfection* IS-A *state*), although *state* was used in that sentence in the sense of 'nation'. This example shows the limitations of the very rough disambiguation method in the algorithm.

Last, we illustrate two particular problems for the algorithm. The first is idiomatic expressions that use words unrelated to the topic:

Example 9
Banks need to realize that there is a *fox* in the *henhouse*...

The word *fox* had no relationships in narrower search scopes; *box* was suggested in its place. (In broader search scopes, *fox* was accepted because of a spurious relationship with *American* nearby: in WordNet, both *Fox* and *American* are hyponyms of *natural language*.) The word *henhouse* was also suspected of being a malapropism, but had no spelling variations. The second problem is rare words that do not appear in the corpus that was used to generate the word probabilities for the method:

Example 10
Charles T. Russell used to play *trombone* in Pittsburgh burlesque houses and with big bands in the Southeast.

The word *trombone* does not appear in the Brown Corpus, and so has zero probability, which leads to taking the logarithm of zero in equation 1. (There is no smoothing in Jiang and Conrath's measure.)

5.3 Quantitative results

We view malapropism detection as a retrieval task and present our results below in terms of precision, recall, and F-measure for each different search scope. In the first step of the algorithm, we say that a suspect is a *true suspect* if it is indeed a malapropism and a *false suspect* if it isn't. In the second step, if an alarm word is indeed a malapropism, we say that the alarm is a *true alarm* and that the malapropism has been *detected*; otherwise, it is a *false alarm*. Then we can define precision (P), recall (R), and F-measure (F) for suspicion ($_S$), involving only the first step, and for detection ($_D$), involving both steps, as follows:

Suspicion

$$P_S = \frac{\text{number of true suspects}}{\text{number of suspects}}, \tag{2}$$

$$R_S = \frac{\text{number of true suspects}}{\text{number of malapropisms in text}}, \tag{3}$$

$$F_S = \frac{2 \times P_S \times R_S}{P_S + R_S}. \tag{4}$$

Detection

$$P_D = \frac{\text{number of true alarms}}{\text{number of alarms}}, \tag{5}$$

$$R_D = \frac{\text{number of true alarms}}{\text{number of malapropisms in text}}, \tag{6}$$

$$F_D = \frac{2 \times P_D \times R_D}{P_D + R_D}. \tag{7}$$

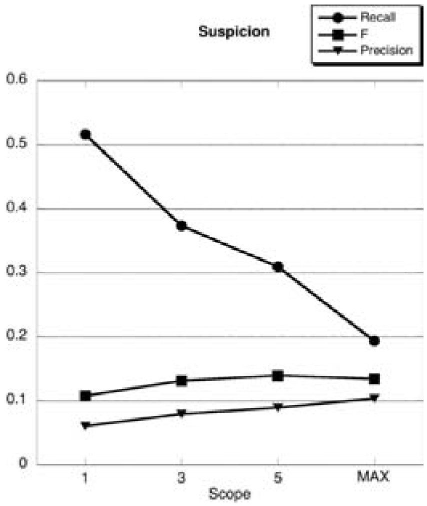

Fig. 4. Suspicion precision (P_S), recall (R_S), and F-measure (F_S), by scope.

Table 1. *Precision, recall, and F after the first step (suspicion) and second step (detection) of the algorithm, varying the scope of the search for related words to 1, 3, or 5 paragraphs or the complete news article (*MAX*)*

	Suspicion			Detection		
Scope	P_S	R_S	F_S	P_D	R_D	F_D
1	0.064	0.536	0.112	0.184	0.498	0.254
3	0.086	0.383	0.135	0.205	0.372	0.245
5	0.097	0.326	0.141	0.219	0.322	0.243
MAX	0.111	0.233	0.137	0.247	0.231	0.211
Chance	0.0129	0.0129	0.0129	0.0129	0.0129	0.0129

5.3.1 Suspicion

We look first at the results for suspicion – just identifying words that have no semantically related word nearby. Obviously, the chance of finding some word that is judged to be related to the target word will increase with the size of the scope of the search (with a large enough scope, e.g. a complete book, we would probably find a relative for just about any word). So we expect recall to decrease as scope increases, because some relationships will be found even for malapropisms; that is, there will be more false negatives. But we expect that precision will increase with scope, as it becomes more likely that (genuine) relationships will be found for non-malapropisms; that is, there will be fewer false positives, and this factor will outweigh the decrease in the overall number of suspects found.

Figure 4 and the left-hand side of Table 1 show suspicion precision, recall, and F for each of the search scopes, computed as the mean values of these statistics across our collection of 481 articles (which constitute a random sample from the

Table 2. *Overall (single-point) precision, recall, and F after the first step (suspicion) and second step (detection) of Hirst and St-Onge's (1998) system and of our system, varying the scope of the search for related words to 1, 3, or 5 paragraphs or the complete news article (*MAX*)*

	Suspicion			Detection		
Scope	P_S	R_S	F_S	P_D	R_D	F_D
Hirst–St-Onge	0.055	0.314	0.094	0.125	0.282	0.174
1	0.060	0.516	0.107	0.157	0.484	0.237
3	0.079	0.373	0.131	0.199	0.365	0.258
5	0.089	0.309	0.139	0.225	0.306	0.260
MAX	0.103	0.193	0.134	0.274	0.192	0.226

population of all *WSJ* articles). The values of precision range from 6.4% to 11.1%, increasing significantly from scope 1 to the larger scopes[8] and those of recall range from 23.3% to 53.6%, decreasing, as expected, with scope (significantly everywhere except from 3 to 5). The value of *F* ranges between 11.2% and 14.1%, with a significant performance improvement from scope 1 to scope 5. All these values are significantly ($p < 0.001$) better than chance, for which all measures are 1.29% (and, of course, this is merely the first stage of a two-stage algorithm).[9] Moreover, the value for precision is inherently limited by the likelihood, as mentioned above, that, especially for small search scopes, there will be words other than our deliberate malapropisms that are genuinely unrelated to all others in the scope.

Although Hirst and St-Onge used their own, custom-made, measures of system performance (see St-Onge (1995)), we can use their figures to compute overall precision, recall, and *F* for their system; this is shown in the top row of the left side of Table 2. These can then be compared with the corresponding quantities computed for our system (shown in the remaining rows of the left side of the table), which are seen to be far superior – with the crucial qualification that this comparison bears no statistical significance because, unlike the figures in Table 1, these are single-point figures, not per-article means (which is why they differ from those in Table 1).

5.3.2 Detection

We now turn to the results for malapropism detection, after the second step of the algorithm. In the detection step, the suspects are winnowed by checking the spelling variations of each for relatedness to context. Since (true) alarms can only

[8] All the comparisons presented, except those with the baseline, were performed with the Bonferroni multiple-comparison technique (Agresti and Finlay 1997), with an *overall* significance level of .01.

[9] To make statistically meaningful comparisons possible, we calculated a separate proportion for each *WSJ* article in the test data, by analogy with the method used to compute the performance of our system. The *mean* was 1.29%, slightly different from the *overall proportion* of 1.31%.

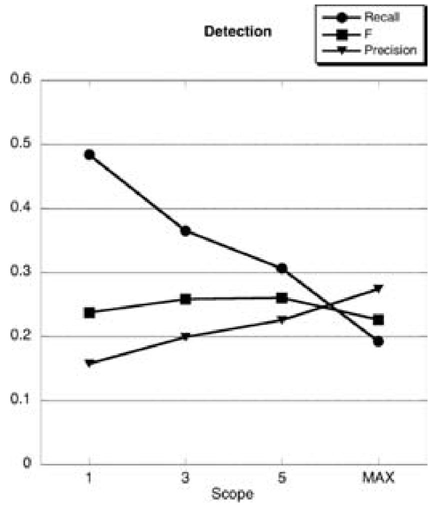

Fig. 5. Detection precision (P_D), recall (R_D), and F-measure (F_D), by scope.

result from (true) suspects, recall can only decrease (more precisely, not increase) from that for suspicion (cf. equations (3) and (6)). However, if the system is any good, the proportion of false alarms will reduce more considerably – far fewer false suspects will become alarms than true suspects – thus resulting in higher precision for detection than for suspicion (cf. equations (2) and (5)).

Figure 5 and the right-hand side of Table 1 show precision, recall, and F for detection, determined by the same method as those for suspicion. The values of recall range from 23.1% to just under 50%. While these values are, as expected, lower than those for suspicion recall, the decline (of 0.3–3.7 percentage points) is *not* statistically significant. The values of precision range from 18.4% to 24.7%, increasing, as expected, from suspicion precision – and the increase (of between 11.9 and 13.6 percentage points) *is* statistically significant at each scope. Furthermore, the increase in precision outweighs the decline in recall, and F, which now ranges from 21.1% to 25.4%, increases by 10.7 percentage points on average; this increase is also statistically significant for all scopes. Again, even the lower ends of the precision, recall, and F ranges are significantly ($p < 0.001$) better than chance (which again is 1.29%), and the results are quite impressive (e.g. 18% precision, 50% recall for scope = 1, which had the highest F_D). The right-hand side of Table 2 shows that this performance again far exceeds that of Hirst and St-Onge.

Scope differences As in the suspicion stage, detection recall goes down with scope (statistically significantly, except from 3 to 5); precision appears to go up, but the increase is in fact statistically significant only between 1 and MAX. These overall flatter precision and recall graphs explain the picture for F_D: there are no significant differences among the scopes, and so the F graph is not significantly different from being flat. Thus we can choose scope = 1 – the smallest, most efficient search – as the optimal scope for our malapropism detector.

5.3.3 Correction

Last, we look at how often detection of a malapropism led to correction of the error. Our algorithm is founded on the assumption that a spelling variation that is more related to context than a malapropism is will be the correct word that the writer intended (or, if there is more than one such spelling variation, then the correct word will be a member of the set). But it is nonetheless possible that a true malapropism could be detected and yet the spelling variation (or set of variations) responsible for this detection is not correct either – in effect, it would be another malapropism, and the detection of the initial malapropism would have been just a lucky accident. We saw this in the *fuss–mugs* example earlier (Example 7 in section 5.2).

In our experiments, we observed only a few such "lucky accidents"; as expected, in almost all cases the correct word was the spelling variation, or one of the variations, responsible for detection of the malapropism, and thus would be suggested to the user in an interactive system. Specifically, the proportion of detected malapropisms for which the correct replacement was found ranged from 92% for scope = 1 to 97.4% for scope = MAX.

6 Discussion

With a recall of 50% and precision of nearly 20%, our system approaches what we believe is a level of practical usability for malapropism detection and correction. It is not realistic to expect absolute correctness, 100% precision and recall, nor is this level of performance necessary for the system to be useful. In conventional interactive spelling correction, it is generally assumed that very high recall is imperative but precision of 25% or even less is acceptable – that is, the user may reject more than three out of four of the system's suggestions without deprecating the system as 'dumb' or not worth using. Very high recall is not yet achievable in unconstrained, open-ended real-word spelling correction, but it is presently unknown just what a typical user would consider to be an acceptable performance level. Nonetheless, we believe that the performance of our system is competitive, especially in light of the constraints under which it presently operates.

Limitations of WordNet In particular, the performance of our prototype is constrained by limitations that arise from the use of WordNet, many of which are likely to be eliminated or attenuated in the future. For example, we have already mentioned in section 4.2 that our prototype uses only the noun portion of WordNet (though adjectives and verb lemmas are taken as equivalent to any noun to which they are identical in spelling) and that a complete system would require links between synsets for different parts of speech, as WordNet 2.0 now permits. Another limitation arises from the fine-grainedness of WordNet; its fine division of word-senses and its inclusion of obscure and metaphoric senses (not labeled as such) are more likely to lead our algorithm astray than to help it. The coarse-grained WordNet presently under development by Mihalcea and Moldovan (2001) could help alleviate this.

Efficiency In our present system, we perform a search in WordNet for each pair of words whose semantic relatedness we wish to know. If semantic-relatedness measures were pre-compiled for all possible pairs of synsets in WordNet, this search could be replaced by table look-up. While the integrated WordNet of the future might contain, say, 100,000 distinct synsets, implying 10^{10} ordered pairs, in practice the table would be symmetric and very sparse; if each word has an above-threshold semantic relationship to no more than a couple of hundred others, and probably fewer, the size of the table might not need to be significantly greater than that of WordNet itself.

Similarity versus semantic relatedness Although we have spoken throughout the paper of semantic relatedness, the Jiang–Conrath measure that we have used is actually a measure of similarity (expressed as its inverse, semantic distance) rather than semantic relatedness per se: the only links of WordNet that it uses are hypernymy, hyponymy, and the implicit link of synonymy. This is in contrast to the measure of Hirst and St-Onge, which used all WordNet noun synset links, including antonymy, meronymy, and holonymy. In a separate paper (Budanitsky and Hirst submitted), in which we compare and evaluate a number of measures of semantic relatedness, we explain in detail why the Jiang–Conrath measure is superior overall to the Hirst–St-Onge measure, even though it will not find non-hypernymic semantic relationships (such as *yacht–keel*) that would clearly be helpful in our task. But regardless of which WordNet relationships are and aren't used, "non-classical" semantic relationships (Morris, Beghtol and Hirst 2003) that do not appear in WordNet at all, especially those that are merely matters of typical association (e.g. *penguin–Antarctica*) will not be found. EuroWordNet (Vossen 1998) employs additional lexical relationships, such as non-factitive causality (*search–find*), that might begin to help with this. In Budanitsky and Hirst (submitted), we discuss this matter, including consideration of the effects of using a *Roget*-style thesaurus in place of WordNet. (We are presently experimenting with the *Macquarie Thesaurus* (Bernard 1986).) Here, it suffices to point out that because the essence of our method lies in a sensitivity to cohesion and perturbations of cohesion, the success of its performance depends on having an excellent measure of semantic relatedness. The Jiang–Conrath measure for WordNet has done well, but there is still much room for improvement.

Setting the threshold of relatedness The algorithm treats semantic relatedness as boolean: two words are either related or they aren't. So if the underlying measure of relatedness is a continuous function (and most of the measures that have been proposed are – see Budanitsky (1999)), then it is necessary to find the breakpoint at which relatedness is separated from unrelatedness. We calibrated the Jiang–Conrath measure with data on human judgments of "similarity of meaning" from Rubenstein and Goodenough's experiments, taking the breakpoint to correspond to that observed in their data. This was justified by the strong correlation that we found between the measure and the human judgments. But the correlation was by no means perfect; the Rubenstein–Goodenough dataset is very small; and of course, similarity of meaning is not the same thing as semantic relatedness. However, there is at present no large dataset of human judgments of semantic relatedness and no better way to calibrate a computational measure.[10]

Proper names Last, our method is limited by its inability to use proper nouns in its considerations of semantic relatedness. As we pointed out in section 2, misleading proper names (such as *Lotus Development Corporation*) could get Hirst and St-Onge's system into serious difficulty, and for that reason we simply excluded them from consideration in our system. But this is also unsatisfactory; many company names contain words used with their ordinary meaning that are potentially helpful in our task – for example, *United Parcel Service* – and it would be helpful to have some method of identifying such words. In addition, many widely known brand names carry meaning that we could usefully relate to other words – for example, *McDonald's–hamburger; Visa–MasterCard–credit* – and the same is true of the names of well-known places and people, some of which are indeed listed in WordNet (e.g. *New York, Statue of Liberty, Bill Clinton*). But any thesaurus that we choose will contain comparatively few proper nouns, and a topical supplement would be desirable.

Limitations of our method of evaluation In addition to limitations in the algorithm itself, our method of evaluation also has its limitations: the use of artificial test data and our somewhat narrow definition of "spelling variation". The need for artificial data is obvious: there is no large-enough, naturally occurring annotated corpus of malapropisms. But, following Hirst and St-Onge, we chose the *Wall Street Journal* as the basis for our corpus merely as a matter of convenience. Thus our results do not necessarily hold for other genres of text. The malapropism-insertion rate of one word in 200 was an arbitrary choice that seemed "sparse enough" to prevent the malapropisms from interacting with one another. Inserting too many malapropisms (one word in ten, say) would be unrealistic and would not just perturb the cohesion of the text but completely destroy it, undermining the very basis of the algorithm. Thus there is an underlying assumption that humans, likewise, do not normally make malapropisms so frequently as to render their text wholly incoherent.

Our results are in part dependent upon the definition of "spelling variation" that we chose – that used by the open-source spelling checker *ispell*. Clearly, the broader the definition, the greater the chance of false alarms and the less well the algorithm will perform. (In the limit, any word could be a spelling variation of any other, so a spurious connection could always be found.) We could not try our algorithm with the extremely liberal definition of spelling variation that is used in Microsoft Word (see footnote 4) as this is proprietary information, but our results would almost certainly be poorer. However, this must be seen as a weakness of Word's overly broad definition, not of our algorithm.

[10] Evgeniy Gabrilovich has recently made available a dataset of similarity judgments of 353 English word pairs that were used by Finkelstein, Gabrilovich, Matias, Rivlin, Solan, Wolfman and Ruppin (2002). This is still very small, but we plan nonetheless to try recalibrating with this dataset in future work.

7 Related research

Kukich (1992) reviews early approaches to the detection of real-word spelling errors; such techniques included looking for unlikely part-of-speech bigrams (Atwell and Elliott 1987), and looking for unlikely word trigrams (Mays, Damerau and Mercer 1991). Some of the more recent work on spelling correction has focused on smarter identification of non-word errors (Zhao and Truemper 1999), the use of syntax (Vosse 1994; Zhao and Truemper 1999), and methods for improving the suggested corrections offered for non-word errors (Mc Hale and Crowter 1996; Agirre, Gojenola, Sarasola and Voutilainen 1998).

The word-trigram method of Mays, Damerau and Mercer (1991) used ideas from a project in speech recognition (Bahl, Jelinek and Mercer 1983). They attempted to apply a statistical language model in which "syntactic, semantic, and pragmatic knowledge is conflated into word trigram conditional probabilities…derived from statistics gathered from large bodies of text". By itself, the model can be used "to judge the relative well-formedness of sentences". This model was combined with the noisy-channel model, thereby making it possible to express the "a priori belief that the observed input word is correct". Briefly, their approach was to consider all variant sentences $s' = w'_1 w'_2 \dots w'_k$ of a given sentence $s = w_1 w_2 \dots w_k$, where w'_i is a spelling variation of w_i for each i, choosing the sentence (possibly s itself) with the highest likelihood.

Mays, Damerau and Mercer (1991) report the results of a preliminary experiment, intended to "assess the viability" of their approach, in which it was applied to just 100 sentences from "the AP newswire and transcripts of the Canadian Parliament", with 8628 spelling variations; the trigram probabilities were borrowed from the IBM speech recognition project (Bahl *et al.* 1983). All of these contained only words from the 20,000 word vocabulary of the IBM corpus, and, to avoid a combinatorial explosion, each s' was a result of a single-word perturbation – that is, for each variant sentence, $w'_i \neq w_i$ for exactly one i. Unfortunately, the idiosyncratic terms used to express their results, coupled with the unavailability of their training corpus, preclude a direct comparison with our work. We are presently reconstructing and re-implementing their method, and will report on a comparison of word-trigram and coherence-based methods in a future paper.

More recently, Verberne (2002) developed a word-trigram method that, instead of using probabilities, considered a trigram to be probably wrong if and only if it does not occur in the British National Corpus. Her evaluation of the method was both small and, on her own test data, methodologically problematic. On a 7000-word sample of our *Wall Street Journal* test data, the method showed a recall of 33% for correction at the price of a precision of only 5%.

Much recent work specifically on real-word spelling correction, especially that of Golding and colleagues (Golding and Roth 1996, 1999; Golding and Schabes 1996) on methods for what they call "context-sensitive spelling correction", has viewed the task as one of "word disambiguation" (Golding and Roth 1996). Ambiguity among words is modeled by pre-specified *confusion sets*: a confusion set $C = \{w_1, \dots, w_n\}$ means that each word $w_i \in C$ "could mistakenly be typed" (Golding and Schabes

1996) when another word $w_j \in C$ was intended – for example, {*principal, principle*}. Given an occurrence of a word from C in the text, then, the task is to decide, from the context, which $w_k \in C$ was actually intended. The specific techniques of addressing the issue are what distinguish the methods. *WinSpell* (Golding and Roth 1996, 1999) uses a machine-learning algorithm in which weights are updated multiplicatively and members of confusion sets are represented as *clouds* of "simple and slow neuron-like" nodes that correspond to *co-occurrence* and *collocation* features. *Tribayes* (Golding and Schabes 1996) combines a part-of-speech trigram method and a Bayesian hybrid method from Golding (1995), both statistical in nature: the trigram method relies on probabilities of part-of-speech sequences and fires for confusion sets whose members would differ as parts of speech when substituted in a given sentence (e.g. {*hear, here*}, {*cite, sight, site*}, and some cases of {*raise, rise*}); the Bayesian hybrid method relies on probabilities of the presence of particular words, as well as collocations and sequences of part-of-speech tags, within a window around a target word and is applied in all the other cases (e.g. for confusion sets like {*country, county*} and (most cases of) {*peace, piece*}). When tested on 21 confusion sets (taken mostly from the list of commonly confused words that is given as an appendix of the *Random House Unabridged Dictionary* (Flexner 1983)), these methods were correct, on average, 93% to 96.4% of the time, compared with a baseline of 74.8% by choosing the most frequent member of the confusion set (Golding and Roth 1999). Carlson, Rosen and Roth (2001) subsequently scaled the method up to 265 confusion sets with up to 99% accuracy.

Other researchers have also used the confusion set model of correction, but with other disambiguation methods. Mangu and Brill (1997), put off by the idea of extracting "large sets of opaque features and weights", as in the Golding methods, applied *data-driven transformation-based learning* to "automatically learn simple, small…sets" of rules for correcting probable instances of confusion-set errors. The rules acquired were intended to account for transformations that correspond to *co-occurrences*, *collocations*, and *collocations with wildcards*. An example of a rule is "Change *except* to *accept* if the word three before is *he* and the immediately preceding word is *not*." The method was tested on 14 confusion sets, with results that were "comparable" to those of Golding and colleagues despite the relative simplicity of the method. Jones and Martin (1997) applied *latent semantic analysis* (Landauer, Foltz and Laham 1998) to the task of discriminating members of confusion sets. Treating sentences as *documents* and words and word bigrams (stemmed and weighted) as *terms*, they constructed a separate predictor space for each confusion set, then formed *projections* of a test sentence onto the space (by computing a weighted average of its term vectors), and chose the member of the confusion set whose vector is closest (in the Euclidean sense). Testing the method with 18 of the confusion sets that Golding and colleagues used, they found the results to be "competitive" with those of Tribayes.

One advantage of these machine-learning-based confusion-set methods over semantic methods such as ours is that they can handle function-word and low-semantic-content–word errors with apparent ease, simply by considering confusion sets such as {*than, then*} and {*to, too*}. Furthermore, they are not restricted to spelling variations: {*amount, number*} is a perfectly valid confusion set. Their principal drawback, however, is that all the confusion sets must be defined in advance: they can look only for specific errors that they know about ahead of time. Thus the process of *detection* is reduced to what might be termed *verification*: a word will be checked for being an error only if it belongs to a confusion set; moreover, every occurrence of such a word will undergo an attempt to be *corrected* (i.e. its confusions will be considered in its place every time the word is encountered). We therefore see confusion-set methods as complementary to our own; each is suitable for a particular kind of error, and a complete spelling checker should draw on both.

Al-Mubaid and Truemper (forthcoming) present a method that is based on the classification of lexical context. Their system aims to find slips in typing or performance errors rather than competence errors; like our system, it will be misled by consistent mistakes. The method is quite complex, but in outline their idea is as follows. In the training phase, deliberate real-word spelling errors are introduced into a "training text". Each word in the text is characterized by a vector that represents its immediate context (±2 tokens), and a classifier then derives rules to distinguish words that are erroneous in their context, as exemplified in the training text, from words that are correct in their context, as exemplified by a separate, unaltered "history text" from the same domain. The method is limited by the fact that both the target word and each of its spelling variations must occur at least three times in each of the training and history texts. It is therefore unable to derive rules for many words, though it performs fairly well on those words that it is able to check. Moreover, a separate set of classification rules must be derived for each domain (with training times of many hours), and the original history text for each domain must be present along with the classification rules whenever the system is used to check a text.

8 Conclusion

The method of detecting and correcting real-word spelling errors that we have presented in this paper is, of course, a research prototype that still awaits integration with a conventional spelling checker for non-word errors and a suitable user interface in a word-processor (or similar software) in order to be tested in a realistic setting. While we speculated above that the performance of our system, in terms of precision and recall, approaches practical usability, only a trial in an integrated system could test this and perhaps determine just what level of performance users would find sufficient in order to gladly use such a system.

We do not claim that our method by itself is sufficient for finding real-word errors. As we remarked above, a practical spelling checker would also employ a confusion set method on words for which it was appropriate, and would probably, in addition, use syntactic methods to detect errors, especially those in closed-class words, that result in syntactic ill-formedness.

We have not attempted to address issues in the user interface. Conventional spelling checkers have very spare interfaces; typically, the suspect word is highlighted in some way and a list of alternatives is presented; it is the user's job to recognize whether the original word or one of the alternatives is what was intended. An integrated system must be careful to distinguish possible malapropisms from non-word errors, or else the user is likely to too-rapidly recognize the highlighted word as correctly spelled and move on. So some message relating to the meaning must be presented; for example:

> Wrong word?
> *Ontologist* means someone who studies the nature of existence.
> Did you mean *oncologist*, someone who studies or treats cancer?

If WordNet is present in the system anyway, its glosses and other words in the synset can be used as the basis for such messages. However, the precise nature of the message is a matter for study.

By recognizing that malapropisms will usually perturb the lexical cohesion of a text, we have demonstrated a practical method for detecting and correcting real-word spelling errors by looking for spelling variations that restore cohesion. Further development of the approach will depend, in turn, upon the development of more-appropriate lexical resources and better models of semantic relatedness.

Acknowledgements

This research was supported financially by the Natural Sciences and Research Council of Canada, the Ontario Graduate Scholarship Program, and the University of Toronto. For discussions, comments, and advice, we are grateful to Mark Chignell, Stephen Green, Jay Jiang, Keith Knight, Claudia Leacock, Dekang Lin, Radford Neal, Manabu Okumura, Gerald Penn, Philip Resnik, David St-Onge, Suzanne Stevenson, and Michael Sussna.

References

Agirre, E., Gojenola, K., Sarasola, K. and Voutilainen, A. (1998) Towards a single proposal in spelling correction. *Proceedings 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-98)*, pp. 22–28. Montreal, Canada.

Agresti, A. and Finlay, B. (1997) *Statistical Methods for the Social Sciences* (3rd ed). Prentice-Hall.

Al-Mubaid, H. and Truemper, K. (forthcoming) Learning to find context-based spelling errors. In: Triantaphyllou, E. and Felici, G., editors, *Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques*. Kluwer.

Atwell, E. and Elliott, S. (1987) Dealing with ill-formed English text. In: Garside, R., Leech, G. and Sampson, G., editors, *The Computational Analysis of English: A Corpus-Based Approach*, pp. 120–138. Longman.

Bahl, L. R., Jelinek, F. and Mercer, R. L. (1983) A maximum likelihood approach to continuous speech recognition. *IEEE Trans. Patt. Anal. Machine Intell.* **5**(2): 179–190.

Barzilay, R. and Elhadad, M. (1999) Using lexical chains for text summarization. In: Mani, I. and Maybury, M. T., editors, *Advances in Automatic Text Summarization*, pp. 111–121. MIT Press.

Bernard, J. R. L. (editor) (1986). *The Macquarie Thesaurus*. Macquarie Library, Sydney.

Budanitsky, A. (1999) *Lexical Semantic Relatedness and its Application in Natural Language Processing*, Technical report CSRG-390, Department of Computer Science, University of

Toronto. *http://www.cs.toronto.edu/compling/Publications/Abstracts/Theses/Budanitsky-thabs.html*

Budanitsky, A. and Hirst, G. (submitted) Evaluating WordNet-based measures of semantic relatedness.

Carlson, A. A., Rosen, J. and Roth, D. (2001) Scaling up context-sensitive text correction. *Proceedings 13th Innovative Applications of Artificial Intelligence Conference*, pp. 45–50. Seattle, WA.

Fellbaum, C. (editor) (1998) *WordNet: An Electronic Lexical Database.* MIT Press.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G. and Ruppin, E. (2002) Placing search in context: The concept revisited. *ACM Trans. Infor. Syst.* **20**(1): 116–131.

Flexner, S. B. (editor) (1983) *Random House Unabridged Dictionary* (2nd ed). Random House.

Fromkin, V. A. (1980) *Errors in Linguistic Performance: Slips of the tongue, ear, pen, and hand.* Academic Press.

Golding, A. R. (1995) A Bayesian hybrid method for context-sensitive spelling correction. *Proceedings Third Workshop on Very Large Corpora*, pp. 39–53. Boston, MA.

Golding, A. R. and Roth, D. (1996) Applying Winnow to context-sensitive spelling correction. In: Saitta, L., editor, *Machine Learning: Proceedings 13th International Conference*, pp. 182–190. Bari, Italy.

Golding, A. R. and Schabes, Y. (1996) Combining trigram-based and feature-based methods for context-sensitive spelling correction. *Proceedings 34th Annual Meeting of the Association for Computational Linguistics*, pp. 71–78. Santa Cruz, CA.

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

Green, S. (1999) Building hypertext links by computing semantic similarity. *IEEE Trans. Knowl. & Data Eng.* **11**(5): 713–731.

Halliday, M. A. K. and Hasan, R. (1976) *Cohesion in English.* Longman.

Hirst, G. and St-Onge, D. (1998) Lexical chains as representations of context for the detection and correction of malapropisms. In: Fellbaum, C. (editor) *WordNet: An Electronic Lexical Database*, pp. 305–332. MIT Press.

Hoey, M. (1991) *Patterns of Lexis in Text.* Oxford University Press.

Jiang, J. J. and Conrath, D. W. (1997) Semantic similarity based on corpus statistics and lexical taxonomy. *Proceedings International Conference on Research in Computational Linguistics*, Taiwan.

Jones, M. P. and Martin, J. H. (1997) Contextual spelling correction using latent semantic analysis. *Proceedings Fifth Conference on Applied Natural Language Processing*, pp. 166–173. Washington, DC.

Kernighan, M. D., Church, K. W. and Gale, W. A. (1990) A spelling correction program based on a noisy channel model. *Proceedings 13th International Conference on Computational Linguistics*, vol. 2, pp. 205–210. Helsinki, Finland.

Kukich, K. (1992) Techniques for automatically correcting words in text. *ACM Comput. Surv.* **24**(4): 377–439.

Landauer, T. K., Foltz, P. W. and Laham, D. (1998) An introduction to latent semantic analysis. *Discourse Processes*, **25**(2–3): 259–284.

Leacock, C. and Chodorow, M. (1998). Combining local context and WordNet similarity for word sense identification. In: Fellbaum, C. (editor) *WordNet: An Electronic Lexical Database*, pp. 265–283. MIT Press.

Lin, D. (1997) Using syntactic dependency as local context to resolve word sense ambiguity. *Proceedings 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the ACL*, pp. 64–71. Madrid, Spain.

Lin, D. (1998) An information-theoretic definition of similarity. *Proceedings 15th International Conference on Machine Learning*, Madison, WI.

Mangu, L. and Brill, E. (1997) Automatic rule acquisition for spelling correction. *Proceedings 14th International Conference on Machine Learning*, pp. 734–741. Nashville, TN.

Mays, E., Damerau, F. J. and Mercer, R. L. (1991) Context based spelling correction. *Infor. Process. Manage.* **27**(5): 517–522.

Mc Hale, M. L. and Crowter, J. J. (1996) Spelling correction for natural language processing systems. *Proceedings Conference on Natural Language Processing and Industrial Applications*, Moncton, Canada.

Mihalcea, R. and Moldovan, D. (2001) Automatic generation of a coarse grained WordNet. *Proceedings Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*, Second Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 35–40. Pittsburgh, PA.

Miller, G. A., Leacock, C., Tengi, R. and Bunker, R. T. (1993) A semantic concordance. *Proceedings ARPA Human Language Technology Workshop*, pp. 303–308. San Francisco, CA.

Mitton, R. (1987) Spelling checkers, spelling correctors, and the misspellings of poor spellers. *Infor. Process. Manage.* **23**(5): 495–505.

Mitton, R. (1996) *English Spelling and the Computer*. Longman.

Morris, J. and Hirst, G. (1991) Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, **17**(1): 21–48.

Morris, J., Beghtol, C. and Hirst, G. (2003) Term relationships and their contribution to text semantics and information literacy through lexical cohesion. *Proceedings 31st Annual Conference of the Canadian Association for Information Science*, Halifax, Canada.

Okumura, M. and Honda, T. (1994) Word sense disambiguation and text segmentation based on lexical cohesion. *Proceedings Fifteenth International Conference on Computational Linguistics (COLING-94)*, pp. 755–761. Kyoto, Japan.

Pedler, J. (2001a) Computer spellcheckers [*sic*] and dyslexics—a performance survey. *Br. J. Educ. Technol.* **32**(1): 23–37.

Pedler, J. (2001b) The detection and correction of real-word spelling errors in dyslexic text. *Proceedings 4th Computational Linguistics UK Colloquium*, pp. 115–119. Sheffield, UK.

Pollock, J. J. and Zamora, A. (1983) Collection and characterization of spelling errors in scientific and scholarly text. *J. Am. Soc. Infor. Sci.* **34**(1): 51–58.

Resnik, P. (1995) Using information content to evaluate semantic similarity. *Proceedings 14th International Joint Conference on Artificial Intelligence*, pp. 448–453. Montreal, Canada.

Rubenstein, H. and Goodenough, J. B. (1965) Contextual correlates of synonymy. *Comm. ACM*, **8**(10): 627–633.

St-Onge, D. (1995) Detecting and correcting malapropisms with lexical chains. Master's thesis, Department of Computer Science, University of Toronto. Published as Technical Report CSRI-319.

Silber, H. G. and McCoy, K. F. (2002) Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics*, **28**(4): 487–496.

Verberne, S. (2002) *Context-sensitive spell* [sic] *checking based on trigram probabilities.* Master's thesis, University of Nijmegen.

Vosse, T. G. (1994) *The Word Connection.* Doctoral dissertation, University of Leiden.

Vossen, P. (1998) *EuroWordNet*. Kluwer.

Zhao, Y. and Truemper, K. (1999) Effective spell [*sic*] checking by learning user behavior. *Appl. Artif. Intell.* **13**(8): 725–742.