“A Statistical Grammar Checker”

Philip S. Kernick and David M.W. Powers

Dept. of Computer Science
Flinders University of SA
GPO Box 2100 Adelaide SA 5001

August 1996

1. Introduction

In conversation, we use English every day, and most of us manage to be understood most of the time. The same cannot be said for our written language. I have many friends who are fluent even gifted speakers, but whose written English is appalling. Much of this can be attributed to spelling errors, which by definition do not occur in speech, but there seems to be a general inability to write proper grammatical sentences.

Is this merely being pedantic, by defining a prescriptive structure that everyone must follow, or does it represent a real problem? I propose that it really is a problem, one caused by a modern device and not an inherent limitation in our way of thinking. The problem is that we type almost everything now.

This report is being typed directly from a loose set of abstract notes - there is no hand-written copy that has been transcribed. A result of touch-typing is that our thoughts “automagically” appear on the screen - or nearly so. To correct our errors most of us rely on some sort of automated spelling and grammar checker. This is completely reasonable as it is well known that people are poor proof-readers of their own work - they tend to read what they thought they wrote, rather than what is actually present. When we finally start using automated tools as our principal proof-reading technique, their true value becomes apparent - they really are inadequate.

Much work has been done in the field of spell checking, but less so in grammar checking. This paper will be approaching the task of automated grammar checking in such a way that it is useful. We make the default assumption that all tested text contains no spelling errors in the sense of non-words.

2. Possible techniques

The first question that requires an answer is “how do we attempt grammar checking?” There are many possible techniques, and four of them will be considered here.

2.1 Use a complete English grammar and part-of-speech dictionary

If these were available, it would be trivial to parse each sentence for grammatical correctness [1, 2]. Many years of research has failed to produce a complete grammar for English. Part of the problem is that “English” is a very broad term. Do we mean written or spoken; casual or formal; British or Australian or American? Each of these options is a subtly different language.

Another problematic factor is that English is a live language - it is constantly evolving. The evolution pressures are not academic, there are no formal review panels who outlaw unacceptable usage as has been attempted in other languages. English evolves as the speakers’ requirements change.

As an example, consider changes of meaning of the word “target” over the last 200 years:

1797.	A kind of buckler or shield borne on the left arm.
1890.	A small buckler or shield; a mark to fire at.
1942.	Shooting mark, esp. a round or rectangular butt divided by concentric circles; butt for scorn, etc.
1985.	Mark or person or object fired at, esp. round or rectangular object divided by concentric circles; objective or result aimed at; butt for scorn, etc.

The evolution of the word is clear, but it is notable that each of these usages has “target” only as a noun. Even as recently as ten years ago it was not defined in the dictionary as a transitive verb as in “I am going to target the leader”.

More and more nouns are being “verbised”, and it is getting to the stage where it is difficult to clearly define a word as being only one or the other.

Furthermore, we coin new words to describe concepts that did not previously exist. A pertinent example in the field of computer science is “hypertext” which was invented as recently as 1966.

For these reasons, it seems folly to attempt to rely only on grammars and dictionaries, as they are almost impossible to produce, and will rarely if ever map to current usage.

2.2 Learn English grammar by induction

Given no knowledge of English and a sufficiently large corpus, it should be possible for a system to learn English by induction, and then use the generated parser to check the correctness of any text.

It could be claimed that an existence proof that this is possible is you! All young children manage to learn their native language with no apparent pre-knowledge [3, 4, 5].

Some AI attempts have been made to do this [6], especially with small, well defined subsets of natural language [7], but these are not appropriate for a general grammar checker.

2.3 Use a set of constraints

Rather than trying to define a grammar for English, it is possible instead to define a set of constraints that bind the words together. This technique is notable because it does not use either a grammar or a dictionary - all the information is taken from the words themselves.

The use of word endings to define these constraints has been attempted, and the results look promising [8]. If we were to look at left and right contexts around a potentially confused word, it might be enough to find errors.

2.4 Tailor made to find errors that writers actually make

This seeks to define the sorts of errors that we make, and then find a way to discover them.

While this sounds like a catch-all, what it really means is to produce a grammar checker that successfully detects certain classes of errors, without trying to be completely general.

2.5 Statistical induction methods

The grammar checking kernel that has been developed uses the principal of 2.4 and some of the ideas from 2.3.

It has been discovered that there is one class of commonly made error that is infrequently highlighted by grammar checkers. These errors are categorised by word substitution of a “near-homonym” or “near-homograph”. The archetypal example is the substitution of “from” for “form” and vice-versa.

Let us consider which parts-of-speech these words represent:

from	preposition
form	noun, transitive verb, intransitive verb

While they represent different parts of speech, we still cannot easily use a grammar to differentiate them. In trying to do so, we just have a reduced version of the “whole of English grammar” problem.

Is it possible to make the attempt without a grammar, using only a complete part-of-speech dictionary? Experimentation suggests that it is not. Analysis of such a dictionary containing 214,100 words, lists two words that can be used in seven different parts-of-speech! They are:

like	adjective, preposition, adverb, conjuction, noun, intransitive verb, transitive verb
down	adverb, preposition, adjective, noun, transitive verb, intransitive verb, interjection

It also lists 94 other words which can be used as all of the major content parts of speech - noun, verb, adjective and adverb. If these words were obscure, it would not significantly increase the difficulty of mapping words to parts-of-speech. The problem is that they aren't - ten of them[1] are in the 150 most commonly used words in English.

This suggests that we are approaching the problem from the wrong end. Instead of asking the question "does this phrase represent grammatically correct English?", we should ask, "which of these two phrases is more likely to be correct English?" The distinction is subtle, but significant.

We need to look for word patterns in the text that will correctly differentiate between commonly confused words, rather than determine objectively whether the text is grammatically correct.

[1] They are: back, down, even, in, just, like, out, right, up & well.

3. Difficulties and solutions

3.1 Where do we look for information

Since we will not be working with a pre-existing grammar, all the required statistical information must be extracted from a set of corpora. For this project I am using a corpus of more than 100 million words [9], composed of articles from Ziff-Davis, the Wall Street Journal and AP Newswire. The choice was made to use non-fiction texts, as they are less thematic and will contain fewer non-words; and to use text that approximates semi-formal spoken English. It is hoped that in these texts, Australian and American will be indistinguishable.

Let us define "target class" as meaning "a set of words that are commonly substituted"; and "target words" as meaning "the words in the target class".

The first question is "now we have a corpus, where do we look for the statistics?" It is obvious that we must look in the vicinity of the target words, but at what range?

3.1.1 Semantic associations

There is experimental evidence that syntactic associations will be found in the range {1 .. 5} words, and semantic associations in the range {1 .. 20} words [10]. This rule-of-thumb appears reasonable, as the syntax defines the structure of a sentence, while the semantics defines its content.

3.1.1.1 Attempt 1 - counting occurrences

If we scan a corpus for each of the target words, and note the frequency with which all other words within a fixed range occur, we can compare these occurrence statistics as a method for differentiating between the words. By using the rule-of-thumb syntactic and semantic diameters, we can have a two dimensional comparison space.

This technique is good at identifying the syntactically binding closed class words, but has little success in discovering open class semantic associations. This result can be attributed to the fact that the frequency of the common words far outweighs the less frequent interesting words.

3.1.1.2 Attempt 2 - restrict diameter based on frequency

To refine this method, and to try and remove the flaw in the technique, the next logical step is to see what information theory tells us about the relationship between information content and frequency.

Information theory states that information content is directly proportional to the negative log of frequency. Hence the more common words have less information content.

We can use this to define the "diameter" of a word as a measure of the range of its influence.

$D_w = -\log_2 f_w$ where D_w = word diameter
f_w = word relative frequency

Using a sample corpus of approximately 100,000 words [11], a word that occurs only once will have a diameter of approximately 17. This matches well with our rule-of-thumb about the semantic diameter of a word. At the other end of the scale, we find problems. Table 1 shows the diameter of the ten most frequently occurring words.

Word	Count	Rate	Diameter
the	5768	18	4
of	3293	31	5
and	2965	34	5
a	2497	41	5
to	2239	45	6
in	1820	56	6
was	1128	90	6
it	852	119	7
is	820	124	7
for	811	126	7

Table 1: Word diameters

The word “the” is by far the most common word in English, and in this corpus it has a frequency of approximately 1/18. This gives it a diameter of 4 words. If we look at the tenth most common word, “for”, it has a frequency of 1/126, and therefore a diameter of 7 words. These large diameters will still overwhelm any interesting results we might hope to find.

We could redefine the “diameter” using offsets and scaling:

$D_w = -\alpha \log_2 f_w + \beta$ where α, β = scaling constants

but then we need an objective definition of the scaling constants, and no such definition that does not seem arbitrary is forthcoming.

3.1.1.3 Attempt 3 - ignore all syntax words

A final attempt is to exclude all the common words, as information theory tells us that their information content is low. It is not a coincidence that these words are the so called “closed class” structure words that function as syntactic binders for the “open class” content words.

We are now considering only semantic information, and are intentionally trying to disregard syntactic information. This is harder than it sounds, as determining which words are structure and which content is almost as grey an area as the distinction between nouns and verbs.

Despite this limitation, it does improve our ability to distinguish between target words in some cases. In our prototypal target class of “form” and “from” we start to see some distinguishing features. This is to be expected as “form” is a content word, and there are contextually related words; whereas “from” is a structure word and is uncorrelated with it’s surrounding content words.

It does not in any way help us to differentiate between “there”, “they’re” and “their” as they are all structure words. Consider: “They’re there with their friends.” If no more than one of the target words is closed class, this technique might be made to work.

In essence we are trying to develop a reverse thesaurus. This differs from a standard thesaurus, in that if we look up “milk” in Roget [12], we find “food: eating and drinking”, “extraction” and “taking”. Thus the standard thesaurus provides an “is-a” mapping. What we are generating is a “has-a” mapping which returns “bottle”, “dairy” and “cow”.

The problems and disappointing results achieved with this method caused us to abandon it and search for another technique.

3.1.2 Syntactic contexts

Since looking at semantics did not give us the results that we were looking for, it seems sensible next to ignore the semantics, and concentrate only on the syntax.

At first inspection this looks like an attempt to induce a grammar. It isn’t - a grammar is a set of rules that can be used to generate all sentences in a language. We are looking for a minimal set of contexts that can be used to distinguish between words.

3.1.2.1 Eigen-words

The term eigen-word is borrowed loosely from the mathematical concept of the eigenvalue. They represent the characteristic syntactic words in English.

These words are not closed class words by definition, but are the most frequent 150 words used in English[2]. Their exact composition is not important, but by their nature they tend to define commonly occurring syntactic structures.

In addition to these words, we define several commonly occurring word classes, and treat them as first-class eigen-words. The classes are shown in Table 2.

The words are classified by their endings, with exception lists kept for each class. For example “sing” is not an “`-ING`” word. These classes are neither exhaustive nor exclusive. In the case of “`ADJ`” there are certainly many other word endings that form adjectives, and there are many other word classes that could have been used. Their purpose, like the other eigen-words is to define commonly occurring grammatical structures. This idea is related to that described in section 2.3, the fundamental difference being that the constraint approach uses *only* the word ending information.

[2] Actually they are the words from `/usr/lib/eign` on SunOS 4.1.3 - `troff` uses them for hyphenation purposes.

Any corpus can now be tokenised entirely into a list of eigen-words and target words.

PUNCT	punctuation marks
-ED	past tense, verb past participle, noun based adjective
-ING	verb present participle
-S	plural nouns, singular verbs
ADJ	adjectives (-al, -ic, -er, -est, -ble, -ous, -ive)
ADV	adverbs (-ly)
CLOSED	closed class words that are not eigen-words
ZERO	any word that does not fit in one of the other classes

Table 2: Word Classes

3.1.2.2 Contextual extension

The smallest possible context is the unencumbered target word. This is defined as having a diameter of zero. In the case of “form” and “from” it is a very significant context, as “from” occurs approximately thirty times more frequently than “form”. If a grammar checker did no more that suggest “from” every time it saw “form”, it would perform quite well. Clearly this will not be the case for every pair of commonly confused words.

We must define an arbitrary acceptance level, below which we will assume that we have a definitive answer. This level also serves as a pruning factor in our search tree. This parameter is tunable, and experience suggests a choice of 5%. If we find a context in which one word occurs greater than 95%, we will claim that the correct word is the most frequent one. This has a direct implication on the false error rate - it now has an upper bound of 5%. If it is outside this range, we must extend the diameter of the context and continue the search. There is one exception to this rule - no matter what the numbers, we will always extend the root of the search tree.

To extend the diameter of the context, we add words alternately to the right and left, creating a larger context for which we also gather statistics. We must gather the ngram statistics in this manner as we cannot directly determine the frequency of an ngram directly from an (n-1)gram. More specifically, the frequency distribution about an ngram is highly dependent on the ngram.

The diameter of such a context is defined to be the number of words it has been extended from the root context. This means that there are two diameter one contexts, one generated by extending the root one word to the right, and the other generated by extending the root one word to the left. The maximum diameter of a context is ten words - five on each side - as this is assumed to extend to the limits of the syntactic association of a word. In practice this has not limited the system in any way, as the largest context that has been induced has diameter nine.

3.1.2.3 Statistical significance

It has been said that “there are lies, damned lies and statistics” [13]. It is vitally important that any statistics we generate and use are significant [14] - that is, the probability of rejecting a true hypothesis is smaller than a fixed upper bound. We use two different significance tests to guarantee this.

The first test is a binary laplacian [15]. This tells us the minimum number of samples that we require before we can claim any significance to the statistics. If insufficient samples are found, we disregard the result. Initially this is set to the same 5% acceptance value as above, but if the target word ratio is smaller than this, the laplacian estimator is reduced to that ratio.

Initially this was the only test that was done, but it was found to be insufficient. Further investigation demonstrated that statistics such as that shown in Table 3 were causing the rejection of correct sentences.

`freq(n, “from”) = 47`	`freq(n, “form”) = 1`
`freq(n-1, “from”) = 1105`	`freq(n-1, “form”) = 5`

Table 3: Aberrant Statistics

If we use a comparison of relative frequency, we are comparing:

$$\frac{{}^{47}\!/_{1105}}{{}^{47}\!/_{1105}+{}^{1}\!/_{5}} \leftrightarrow \frac{{}^{1}\!/_{5}}{{}^{47}\!/_{1105}+{}^{1}\!/_{5}} = 18\% \leftrightarrow 82\%$$

Hence the use of “form” was considered more than four times more likely than “from”. Clearly this is in error. The reason for this error is that a small change in the “form” statistics affect the result far more than a small change in the “from” statistics.

For this reason, the second test that is applied is a first order derivative. We calculate a value of the likelihood that the context is acceptably defined as above, based on the ratio of relative frequency of occurrence. The partial derivative of this likelihood with respect to the number of samples in each class is then calculated.

$$L(\chi^t) = \frac{{}^{\chi_0^t}\!/_{\chi_{-1}^t}}{\sum_i {}^{\chi_0^i}\!/_{\chi_{-1}^i}} \Rightarrow \frac{\partial L}{\partial \chi^t} = \frac{\sum_{i\neq t} {}^{\chi_0^i}\!/_{\chi_{-1}^i}}{\chi_{-1}^t\left(\sum_i {}^{\chi_0^i}\!/_{\chi_{-1}^i}\right)^2}$$

If the rate of change is too large, the sample is ignored. In the case above:

$$\frac{\partial L}{\partial \text{form}} \approx \frac{1}{7}$$

$$\frac{\partial L}{\partial \text{from}} \approx \frac{1}{325}$$

Since “from” skews the data too much, it’s statistics are ignored, and the result is treated as if the statistics were:

$$\frac{{}^{47}\!/_{1105}}{{}^{47}\!/_{1105} + {}^{0}\!/_{5}} \leftrightarrow \frac{{}^{0}\!/_{5}}{{}^{47}\!/_{1105} + {}^{0}\!/_{5}} = 100\% \leftrightarrow 0\%$$

This stops the data being skewed by an anomalous comparison of a very large class and a very small class, which is the desired result.

3.1.2.4 Differential grammars

When we have completely traversed the search space available, and noted all the significant defining contexts (and those non-defining contexts for which no further information is available), we are left with a grammar of sorts. Rather than being a standard generating grammar, we have a *differential* grammar.

An interesting comparison can be made with a *distuent* grammar [16], one which simply lists the tokens which cannot occur together in a language. A distuent grammar has been shown to be capable of inducing most of noun-phrase and prepositional-phrase structures within a corpus, but its accuracy is insufficient to be useful in grammar checking. It has been suggested that by giving the distuent parser more “linguistic” information, it may be able to significantly increase it’s accuracy. This is just the sort of information that has been intentionally ignored in the case of the differential grammar - and has been shown to be unnecessary for effective word discrimination..

A differential grammar is defined as “a set of contexts that differentiate statistically between target words”. An example of a context from a differential grammar for “form” and “from” is:

[3, 97]					-ING	↔	a				

This context with a diameter of two is read as: if you see a list of words that looks like “a word ending in ‘ing’”, followed by a word from the target class, followed by “a”, then 3% of the time, the correct choice is the first target word, and 97% of the time, the correct choice is the second target word.

As these statistics have been generated from a very large corpus, and are therefore assumed ergodic, we can claim that the false error rate should be less than 5%. This is significantly better than the 30-40% deemed acceptable by Wampler [17]. If it is significantly above this, then we should retrain the system with a sample of the text to be checked.

This induced differential grammar, is sufficient to distinguish between target words.

3.2 How can we test a piece of text

Testing a piece of text now becomes simple.

The text must be tokenised into eigen-words and target words. In this list of tokens we search for target words. When a target word is found we compare the surrounding context with the defining contexts in the differential grammar, and select the largest one that matches. This presents us with a statistical choice as to whether the word is correct or incorrect. However this is still insufficient, as we must remember that there is less information in a diameter one context than a diameter ten context. As a result, the function that flags an error is dependent on context diameter.

$$\theta = \frac{20}{\delta} + 65, \text{ where, } \delta = \text{context diameter; } \theta = \text{certainty threshold}$$

Initially we always extended the context first right then left, but there are some very strong diameter one left contexts that were not being discovered[3]. As a result, both first left, then first right alternating context is generated, and the resulting differential grammars merged.

In some cases this may lead to more than one context that matches. This can occur when for example, we get a match with a diameter one context to the left and to the right. In this case we can have three results, (a) the word is correct; (b) the word is wrong; (c) the left and right contexts disagree. In this final case there is probably something significantly wrong with the phrase.

A real example of this is “I don’t know where you’re coming from on this” - the diameter one contexts strongly suggest both “coming from” and “form on”. This is an example of slang expression that was not present in the training text, hence we must decide that the assumption of ergodicity of the training test text is false, or that the sentence is inherently grammatically incorrect. This will always be the case for genre or sublanguage texts, and is the fundamental reason that they were *not* used in the training corpora. A corollary to this is that accuracy of testing will likely be low on these sort of texts.

[3] Consider “the from” vs “the form”.

4. Success of the differential grammar

4.1 Techniques used in other grammar checkers

The Microsoft Word 6™ grammar checker was applied to a set of test sentences with only one rule enabled: “Commonly Confused Words”. Note that each sentence is wrong. The results are shown in Table 4.

Test sentence	Correct?
He was tired form running.	✗
He was wibbled form running.	✗
He was orange form running.	✔
He was red form running.	✔
He was slow form running.	✔
He was smiling form running.	✗
He was glinking form running.	✗
He was glinking form rungling.	✔
He was smiling form rungling.	✔
He was smiling form yawning.	✔
He was wibbling form laughing.	✗
He was hurting form bleeding.	✗
He was hurting form blessing.	✔
He was hurting form catching.	✗
He was hurting form coughing.	✔
He was hurting form crying.	✔
He was hurting form falling.	✗
He was hurting form laughing.	✗
He was hurting form looking.	✗
He was hurting form reading.	✔
He was hurting form running.	✗
He was hurting form skiing.	✔
He was hurting form smiling.	✔
He was hurting form sneezing.	✔
He was hurting form sniffing.	✔
He was hurting form wibbling.	✔
He was hurting form writing.	✔

Table 4: What is Word doing?

Well might we ask “What is Word doing?” Of the 27 sentences, only 11 are correctly detected as being in error. Furthermore it is not consistent - occasionally it knows what to do with non-words, but it also gets a large number of real words wrong. Results like this make it very difficult to determine what algorithm, if any, the Word grammar checker is using.

Using a differential grammar, each of these sentences is detected as being wrong.

4.2 Initial comparison with commercial grammar checkers

An attempt has been made to quantitatively test how well the differential grammar checker functions. To do this a test corpus was checked, and the number of errors counted. These errors can be classified using the into one of two classes.

Statistics calls these errors "type I" in the case when an error is reported when the word is correct; and "type II" in the case when an error is not reported when the word is incorrect.

The same text was checked with the differential grammar checker, the Microsoft Word 6 grammar checker and the Word Perfect 6 grammar checker[4]. The results are shown in Table 5.

	type I	type II
Microsoft Word	4.4%	0%
WordPerfect	0.0%	100%
Differential Checker	1.9%	0%

Table 5: Comparison of grammar checkers

It will be interesting to see if these results are consistent across a large range of test corpuses.

The recognisor has been tested on a large selection my own text, and has found several errors that I had not previously detected.

5. Future work

While some quite impressive results have been achieved, there are still a number of areas require more detail.

1. Additional eigen-word classes.

   Experimentation has suggested that additional classes of punctuation and quotation marks would be useful in increasing accuracy. The possessive case also needs to be dealt with specially. It would also be useful to have a larger class that are defined to be "closed" rather than "open".

2. Determination of additional commonly confused words.

[4] Note that for the commercial grammar checkers only the "commonly confused words" rule was enabled.

Other sets of commonly confused words will be found, and differential grammars induced for each set. As each target class is independent, additional classes can be added at any time.

3. Implementation of the grammar checker user interface.

   A user interface will be added to the grammar checker such that probable grammar errors will be flagged by a colour coding mechanism. Correct text will be coloured black, and suspected incorrect text will be coloured on a scale from blue to red. It will allow a user to customise the checking preferences.

4. Detailed comparison with commercial grammar checkers.

   The grammar checker will be compared to two commercially available programs, both on known good text and known bad text. The known good text will be a selection of journal articles or books, and the known bad text will be a selection of Usenet news articles. A corpus of my own work will then be tested to see how well it is written.

6. References

[1] Webber B, “*Natural Language Processing*”, McGraw Hill Encyclopedia of Science and Technology, vol 11, pp 523-526, McGraw Hill, 1987.

[2] Obermeier K.K., “*Natural Language Processing*”, Byte, pp 225-228, Dec. 1987.

[3] Hetherington E.M. & Parke R.D., “*Child Psychology, A Contemporary Viewpoint*”, pp 271-273, McGraw Hill, 1979.

[4] Mussen P.H., “*Psychological Development: A Life-Span Approach*”, pp 129-140, Harper and Row, 1979.

[5] Crain S, “*Language Acquisition in the Absence of Experience*”, Behavioural and Brian Sciences, vol 14, no 4, pp 597-612, Dec 1991.

[6] Powers D.M.W., “*Hierarchical Self-Organization of Corpora: The conflict between Theory and Practice*”, ACSC 1996 (rejected).

[7] Huijsen W, “*Introduction to Controlled Languages*”, `http://wwwots.let.ruu.nl/Controlled-languages/faq.html`

[8] Entwistle J & Groves M, “*A Method of Parsing English Based on Sentense Form*”, Proceedings of International Conference on New Methods in Language Processing, 1994.

[9] TIPSTER Information Retrieval, “*Text Research Collection Vol. 2*”, University of Pennsylvania, March 1994.

[10] Powers D, “*Unsupervised learning of linguistic structure: An empirical evaluation*”, Journal of Corpus Linguistics, vol 2, no 1, 1996 (to appear).

[11] Sterling B, “*The Hacker Crackdown: Law and Disorder on the Electronic Frontier*”, Bantam Books, 1994.

[12] Lloyd S, “*Roget’s Thesaurus of English Words and Phrases*”, pg 625, Penguin Books, 1984.

[13] Disraeli B, “*Autobiography*”.

[14] Hines W & Montgomery D, “*Probability and Statistics in Engineering and Management Science*”, 2nd ed, pg 269, John Wiley & Sons, 1980.

[15] Webb G, “*Further Experimental Evidence against the Utility of Occam’s Razor*”, Journal of Artificial Intelligence Research 4, pp 397-417, 1996.

[16] Magerman D.M. & Marcus M.P., "*Distuent Parsing and Grammar Induction*", Proceedings of MLNLO'91, pp 122a-122e, 1991.

[17] Wampler B.E., *"Risks of grammar checkers"*, `comp.risks`, 17.54, 1995.