# TRIPHONE ANALYSIS: A COMBINED METHOD FOR THE CORRECTION OF ORTHOGRAPHICAL AND TYPOGRAPHICAL ERRORS.

Brigitte van Berkel† and Koenraad De Smedt*

† Institute for Applied Computer Science (ITI), TNO
Schoemakerstraat 97, 2628 VK Delft, The Netherlands

*Language Technology Project, Psychology Dept., University of Nijmegen
Montessorilaan 3, 6525 HR Nijmegen, The Netherlands

## ABSTRACT

Most existing systems for the correction of word level errors are oriented toward either typographical or orthographical errors. Triphone analysis is a new correction strategy which combines phonemic transcription with trigram analysis. It corrects both kinds of errors (also in combination) and is superior for orthographical errors.

## 1. INTRODUCTION

### 1.1 Error types

Any method for the correction of word level errors in written texts must be carefully tuned. On the one hand, the number of probable corrections should be maximized; on the other hand, the number of unlikely corrections should be minimized. In order to achieve these goals, the characteristics of specific error types must be exploited as much as possible. In this article we distinguish two major types of word level errors: *orthographical* errors and *typographical* errors. They have some clearly different characteristics.

*Orthographical errors* are cognitive errors consisting of the substitution of a deviant spelling for a correct one when the author either simply doesn't know the correct spelling for a correct spelling, forgot it, or misconceived it. An important characteristic of orthographical errors is that they generally result in a string which is phonologically identical or very similar to the correct string (e.g. *indicies* instead of *indices*[1]). As a consequence, orthographical errors are dependent on the correspondence between spelling and pronunciation in a particular language. Another characteristic is that proper names, infrequent words and foreign words are particularly prone to orthographical errors.

[1]All examples of errors given in this article were actually found by the authors in texts written by native speakers of the language in question.

*Typographical errors* are motoric errors caused by hitting the wrong sequence of keys. Hence their characteristics depend on the use of a particular keyboard rather than on a particular language. Roughly eighty percent of these errors can be described as single deletions (e.g. *continous*) insertions (e.g. *explaination*), substitutions (e.g. *anyboby*) or transpositions (e.g. *autoamtically*) while the remaining twenty percent are complex errors (Peterson, 1980). Some statistical facts about typographical errors are that word-initial errors are rare, and doubling and undoubling (e.g. *succeeed, discusion*) are common. In general, typographical errors do not lead to a string which is homophonous with the correct string.

Most of the correction methods currently in use in spelling checkers are biased toward the correction of typographical errors. We argue that this is not the right thing to do. Even if orthographical errors are not as frequent as typographical errors, they are not to be neglected for a number of good reasons. First, orthographical errors are *cognitive* errors, so they are more persistent than typographical errors: proofreading by the author himself will often fail to lead to correction. Second, orthographical errors leave a worse impression on the reader than typographical errors. Third, the use of orthographical correction for standardization purposes (e.g. consistent use of either British or American spelling) is an important application appreciated by editors. In this context, our research pays special attention to Dutch, which has a preferred standard spelling but allows alternatives for a great many foreign words, e.g. *architect* (preferred) vs. *architekt* (allowed and commonly used in Dutch). Editors of books generally prefer a consistent use of the standard spelling.

Finally, we would like to point out that methods for orthographical error correction can not only be applied in text processing, but also in database retrieval. In fact, our research was prompted partly by a project proposal for a user interface to an electronic encyclopedia. One or our experiments involving a lists of some five thousand worldwide geographical

names (mainly in Dutch spelling, e.g. *Noordkorea, Nieuwzeeland*) has yielded very positive results. In this context, the correction of orthographical errors is obviously more important than the correction of typographical errors.

### 1.2 Correction strategies

Daelemans, Bakker & Schotel (1984) distinguish between two basic kinds of strategies: *statistical* and *linguistic* strategies. *Statistical* strategies are based on string comparison techniques, often augmented by specific biases using statistical characteristics of some error types, such as the fact that typographical errors do not frequently occur in the beginning of a word. Since these strategies do not exploit any specific linguistic knowledge, they will generally work better for typographical errors than for orthographical errors.

*Linguistic* strategies exploit the fact that orthographical errors often result in *homophonous* strings (*sound-alikes*. e.g. *consistancy* and *consistency*). They normally involve some kind of phonemic transcription. Typographical errors which do not severely affect the pronunciation, such as doubling and undoubling, may be covered as well, but in general, linguistic strategies will do a poor job on all other typographical errors.

Because each type of strategy is oriented toward one class of errors only, what is needed in our opinion is a combined method for orthographical *and* typographical errors. Our research has explored one approach to this problem, namely, the combination of a linguistic strategy with a statistical one.

The remainder of this document is structured as follows. First we will discuss and criticize some existing statistical and linguistic correction methods. Then we will introduce triphone analysis. Finally we will report some results of an experiment with this method.

## 2. SOME EXISTING CORRECTION METHODS

### 2.1 Spell

In Peterson's SPELL (Peterson, 1980), all probable corrections are directly generated from an incorrect string by considering the four major single error types. The program first makes a list of all strings from which the incorrect string can be derived by a single deletion, insertion, substitution or transposition. This list is then matched against the dictionary: all strings occuring in both the list and the dictionary are considered probable corrections. Although the number of derivations is relatively small for short strings, they often lead to several probable corrections because many of them will actually occur in the dictionary. For longer strings, many possible derivations are considered but most of those will be non-existent words.

An advantage of SPELL with respect to all other methods is that short words can be corrected equally well as long ones. A disadvantage is that all complex errors and many orthographical errors fall outside the scope of SPELL.

### 2.2 Speedcop

SPEEDCOP (Pollock & Zamora, 1984) uses a special technique for searching and comparing strings. In order to allow a certain measure of similarity, strings are converted into *similarity keys* which intentionally blur the characteristics of the original strings. The key of the misspelling is looked up in a list of keys for all dictionary entries. The keys found in the list within a certain distance of the target key are considered probable corrections.

The blurring of the similarity keys must be carefully finetuned. On the one hand, if too much information is lost, too many words collate to the same key. If, on the other hand, too much information is retained, the key will be too sensitive to alterations by misspellings. Two similarity keys are used in SPEEDCOP: a *skeleton key* and an *omission key*. These keys are carefully designed in order to *partially* preserve the characters in a string and their interrelationships. The information contained in the key is ordered according to some characteristics of typographical errors, e.g. the fact that word-initial errors are infrequent and that the sequence of consonants is often undisturbed.

The *skeleton key* contains the first letter of a string, then the remaining consonants and finally the remaining vowels (in order, without duplicates). E.g. the skeleton key of *information* would be *infrmtoa*. The advantage of using this key is that some frequent error types such as doubling and undoubling of characters as well as transpositions involving one consonant and one vowel (except for an initial vowel) results in keys which are identical to the keys of the original strings.

The most vulnerable aspect of the skeleton key is its dependence on the first few consonants. This turned out to be a problem, especially for omissions. Therefore, a second key, the *omission key*, was developed. According to Pollock & Zamora (1984), consonants are omitted in the following declining or-

der of frequency: RSTNLCHDPGMFBYWVZXQKJ. The omission key is construed by first putting the consonants in increasing order of omission frequency and adding the vowels in order of occurrence. E.g. the omission key for *information* is *fmntrioa*.

SPEEDCOP exploits the statistical properties of typographical errors well, so it deals better with frequent kinds of typographical errors than with infrequent ones. Because of this emphasis on typographical errors, its performance on orthographical errors will be poor. A specific disadvantage is its dependence on the correctness of initial characters. Even when the omission key is used, word-initial errors involving e.g. *j* or *k* do not lead to an appropriate correction.

### 2.3 Trigram analysis: Fuzzie and Acute

Trigram analysis, as used in FUZZIE (De Heer, 1982) and ACUTE (Angell, 1983), uses a more general similarity measure. The idea behind this method is that a word can be divided in a set of small overlapping substrings, called *n*-grams, which each carry some information about the identity of a word. When a misspelling has at least one undisturbed *n*-gram, the correct spelling spelling can still be traced. For natural languages, *tri*grams seem to have the most suitable length. E.g., counting one surrounding space, the word *trigram* is represented by the trigrams *#tr*, *tri*, *rig*, *igr*, *gra*, *ram*, and *am#*. *Bi*grams are in general too short to contain any useful identifying information while *tetra*grams and larger *n*-grams are already close to average word length.

Correction using trigrams proceeds as follows. The trigrams in a misspelling are looked up in an *inverted file* consisting of all trigrams extracted from the dictionary. With each trigram in this inverted file, a list of all words containing the trigram is associated. The words retrieved by means of the trigrams in the misspelling are probable corrections.

The difference between FUZZIE and ACUTE is mainly in the criteria which are used to restrict the number of possible corrections. FUZZIE emphasizes frequency as a selection criterium whereas ACUTE also uses word length. Low frequency trigrams are assumed to have a higher identifying value than high frequency trigrams. In FUZZIE, only the correction candidates associated with the *n* least frequent trigrams, which are called *selective trigrams*, are considered. ACUTE offers the choice between giving low frequency trigrams a higher value and giving all trigrams the same value.

Taking trigram frequency into account has advantages as well as disadvantages. On the one hand, there is a favorable distribution of trigrams in natural languages in the sense that there is a large number of low frequency trigrams. Also, the majority of words contain at least one selective trigram. On the other hand, typographical errors may yield very low frequency trigrams which inevitably get a high information value.

In general, trigram analysis works better for long words than for short ones, because a single error may disturb all or virtually all trigrams in a short word. Some advantages of this method are that the error position is not important and that complex errors (e.g. *differenent*), and, to a certain extent, orthographical errors, can often be corrected. A disadvantage which is specific to this method is that transpositions disturb more trigrams than other types of errors and will thus be more difficult to correct.

Trigram analysis lends itself well to extensions. By first selecting a large group of intermediate solutions, i.e. all words which share at least one selective trigram with the misspelling, there is a lot of room for other factors to decide which words will eventually be chosen as probable corrections. ACUTE for example uses word length as an important criterium.

### 2.4 The PF-474 chip

The PF-474 chip is a special-purpose VLSI circuit designed for very fast comparison of a string with every entry in a dictionary (Yianilos, 1983). It consists of a DMA controller for handling input from a data base (the dictionary), a *proximity computer* for computing the proximity (similarity) of two strings, and a *ranker* for ranking the 16 best solutions according to their proximity values.

The *proximity value* (PV) of two strings is a function of the number of corresponding characters of both strings counted in forward and backward directions. It is basically expressed as the following ratio:

$$PV=\frac{2*(AB_{forward}+AB_{backward})}{AA_{forward}+AA_{backward}+BB_{forward}+BB_{backward}}$$

This value can be influenced by manipulating the parameters *weight*, *bias* and *compensation*. The parameter *weight* makes some characters more important than others. This parameter can e.g. be manipulated to reflect the fact that consonants carry more information than vowels. The parameter *bias* may correct the weight of a character in either word-initial or word-final position. The parameter *compensation* determines the importance of an occurrence of a certain character within the word. By using a high

compensation/weight ratio, for example, substitution of characters will be less severe than omission. One may force two characters to be considered identical by equalizing their compensation and weight values.

An advantage of the PF-474 chip, apart from its high speed, is that it is a general string comparison technique which is not biased to a particular kind of errors. By carefully manipulating the parameters, many orthographical errors may be corrected in addition to typographical errors.

### 2.5 Spell Therapist

SPELL THERAPIST (Van Berkel, 1986) is a linguistic method for the correction of orthographical errors. The misspelling is transcribed into a phonological code which is subsequently looked up in a dictionary consisting of phonological codes with associated spellings. The phonemic transcription, based on the GRAFON system (Daelemans, 1987), is performed in three steps. First the character string is split into syllables. Then a rule-based system converts each syllable into a phoneme string by means of *transliteration rules*. These syllabic phoneme strings are further processed by *phonological rules* which take the surrounding syllable context into account and are finally concatenated.

The transliteration rules in SPELL THERAPIST are grouped into three ordered lists: one for the onset of the syllable, one for the nucleus, and one for the coda. Each rule consists of a graphemic selection pattern, a graphemic conversion pattern, and a phoneme string. The following rules are some examples for Dutch onsets:

(( sc ( ~ h i e y )) c /k/ )

(( qu ) qu ( /k/ /kw/ ))

(( a ( consonantp )) a /a/ )

The first rule indicates that in a graphemic pattern consisting of *sc* which is *not* followed by either *h*, *i*, *e* or *y*, the grapheme *c* is to be transcribed as the phoneme /k/.

The transcription proceeds as follows. The onset of a syllable is matched with the graphemic selection patterns in the onset rule list. The first rule which matches is selected. Then the characters which match with the conversion pattern are converted into the phoneme string. The same procedure is then performed for the nucleus and coda of the syllable.

The result of the transcription is then processed by means of phonological rules, which convert a sequence of phonemes into another sequence of phonemes in a certain phonological context on the level of the word. An example for Dutch is the *cluster reduction* rule which deletes a /t/ in certain consonant clusters:

((( obstruent-p ) /t/ ( obstruent-p )) /t/ // )

Such rules account for much of the power of SPELL THERAPIST because many homophonous orthographic errors seem to be related to rules such as *assimilation* (e.g. *inplementation*) or cluster reduction and degemination (e.g. Dutch *kunstof* instead of *kunststof*).

This method is further enhanced by the following refinements. First, a spelling may be transcribed into more than one phonological code in order to account for possible pronunciation variants, especially those due to several possible *stress* patterns. Second, the phonological code itself is designed to intentionally blur some finer phonological distinctions. E.g. in order to account for the fact that short vowels in unstressed syllables are prone to misspellings (e.g. *optomization, incoded*) such vowels are always reduced to a *schwa* /ə/. As a result, misspellings of this type will collocate.

It is clear that this method is suited only for errors which result in completely homophonous spellings (e.g. *issuing, inplementation*). A somewhat less stringent similarity measure is created by using a coarse phonological coding, as mentioned above. Still, this method is not suitable for most typographical errors. Moreover, orthographical errors involving 'hard' phonological differences (e.g. *managable, recommand*) fail to lead to correction.

## 3. AN INTEGRATED METHOD

### 3.1 Combining methods

Of the methods described in the previous chapter, no single method sufficiently covers the whole spectrum of errors. Because each method has its strengths and weaknesses, it is advantageous to combine two methods which supplement each other. Because orthographical errors are the most difficult and persistent, we chose to take a linguistic method as a starting point and added another method to cover its weaknesses. SPELL THERAPIST has two weak points. First, most typographical errors cannot be corrected. Second, even though the phonological codes are somewhat blurred, at least one possible transcription of the misspelling must match exactly with the phonological code of the intended word.

A possible solution to both problems consists in applying a general string comparison technique to phonological codes rather than spellings. We decided

to combine SPELL THERAPIST with trigram analysis by using sequences of three phonemes instead of three characters. We call such a sequence a *triphone* and the new strategy *triphone analysis*.

### 3.2 Triphone analysis

Triphone analysis is a fast and efficient method for correcting orthographical and typographical errors. When carefully implemented, it is not significantly slower than trigram analysis. The new method uses only one dictionary in the form of an inverted file of triphones. Such a file is created by first computing phonological variants for each word, then splitting each code into triphones, and finally adding backpointers from each triphone in the file to each spelling in which it occurs. Also, a frequency value is associated with each triphone.

The way this inverted file is used during correction is virtually the same as in FUZZIE, except that first all phonological variants of the misspelling have to be generated. The grapheme-to-phoneme conversion is similar to that of SPELL THERAPIST, except that the phonological code is made even coarser by means of various simplifications., e.g. by removing the distinction between tense and lax vowels and by not applying certain phonological rules.

The easiest way to select probable corrections from an inverted file is the method used by FUZZIE, because the similarity measure used by ACUTE requires that the number of triphones in the possible correction be known in advance. The problem with this requirement is that phonological variants may have different string lengths and hence a varying number of triphones.

Using the FUZZIE method, each phonological variant may select probable corrections by means of the following steps:

1. The phonological code is split into triphones.
2. Each triphone receives an information value depending on its frequency. The sum of all values is 1.
3. The selective triphones (those with a frequency below a certain preset value) are looked up in the inverted file.
4. For all correction candidates found in this way, the similarity with the misspelling is determined by computing the sum of the information values of all triphones shared between the candidate and the misspelling.

If a certain candidate for correction is found by more than one phonological variant, only the highest information value for that candidate is retained. After candidates have been selected for all variants, they are ordered by their similarity values. A possible extension could be realized by also taking into account the difference in string length between the misspelling and each candidate.

Because processing time increases with each phonological variant, it is important to reduce the number of variants as much as possible. A considerable reduction is achieved by not generating a separate variant for each possible stress pattern. The resulting inaccuracy is largely compensated by the fact that a perfect match is no longer required by the new method.

Although this method yields very satisfactory results for both orthographical and typographical errors and for combinations of them, it does have some shortcomings for typographical errors in short words. One problem is that certain deletions cause two surrounding letters to be contracted into very different phonemes. Consider the deletion of the *r* in *very*: the pronunciation of the vowels in the resulting spelling, *vey*, changes substantially. Counting one surrounding space, the misspelling does not have a single triphone in common with the original and so it cannot be corrected.

A second problem is that a character (or character cluster) leading to several possible phonemes carries more information than a character leading to a single phoneme. Consequently, an error affecting such a character disturbs more triphones.

### 3.3 An experiment

The triphone analysis method presented here has been implemented on a Symbolics LISP Machine and on an APOLLO workstation running Common LISP. After the programs had been completed, we decided to test the new method and compare its qualitative performance with that of the other methods.

For a first, preliminary test we chose our domain carefully. The task domain had to be very error-prone, especially with respect to orthographical errors, so that we could elicit errors from human subjects under controlled circumstances. Given these requirements, we decided to choose Dutch surnames as the task domain. In Dutch, many surnames have very different spellings. For example, there are 32 different names with the same pronunciation as *Theyse*, and even 124 ways to spell *Craeybeckx*! When such a name is written in a dictation task (e.g. during a telephone conversation) the chance of the right spelling being chosen is quite small.

For our experiment, we recorded deviant spellings of Dutch surnames generated by native speakers of Dutch in a writing-to-dictation task. A series of 123 Dutch surnames was randomly chosen from a telephone directory. The names were dictated to 10 subjects via a cassette tape recording. A comparison of the subjects' spelling with the intended spellings showed that on the average, subjects wrote down 37.6% of the names in a deviant way. The set of 463 tokens of misspellings contained 188 different types, which were subsequently given as input to implementations of each of the methods[2]. The dictionary consisted of 254 names (the 123 names mentioned above plus 131 additional Dutch surnames randomly selected from a different source). The results of the correction are presented in Tables 1 and 2.

Table 1. Results of the evaluation study. The numbers refer to percentages of recognized (first, second or third choice) or not recognized surnames ($n$ = 188).

| | 1st choice | 2nd or 3rd | not found |
|---|---|---|---|
| SPELL | 58.5 | 1.1 | 40.4 |
| SPEEDCOP | 53.7 | 1.1 | 45.2 |
| FUZZIE | 86.2 | 9.6 | 4.2 |
| ACUTE | 89.9 | 6.9 | 3.2 |
| PF-474 | 84.0 | 14.9 | 1.1 |
| SPELL THERAPIST | 86.2 | 1.1 | 12.8 |
| TRIPHONE ANALYSIS | 94.1 | 5.9 | 0.0 |

Table 2. Results of the evaluation study. The numbers refer to percentages of recognized (first, second or third choice) or not recognized surnames multiplied by their frequencies ($n$ = 463).

| | 1st choice | 2nd or 3rd | not found |
|---|---|---|---|
| SPELL | 63.7 | 2.2 | 34.1 |
| SPEEDCOP | 55.7 | 2.2 | 42.1 |
| FUZZIE | 87.7 | 8.4 | 3.9 |
| ACUTE | 90.3 | 6.7 | 3.0 |
| PF-474 | 85.5 | 14.1 | 0.4 |
| SPELL THERAPIST | 90.5 | 2.2 | 7.3 |
| TRIPHONE ANALYSIS | 95.2 | 4.8 | 0.0 |

[2] The PF-474 method was simulated in software instead of using the special hardware.

### 3.4 Discussion

The experiment was designed in order to minimize typographical errors and to maximize orthographical errors. Hence it is not surprising that SPELL and SPEEDCOP, which are very much dependent on the characteristics of typographical errors, do very poorly. What is perhaps most surprising is that SPELL THERAPIST, a method primarily aiming at the correction of orthographical errors, shows worse results than FUZZIE, ACUTE and the PF-474 method, which are general string comparison methods. The reason is that a certain number of orthographical errors turned out to involve real phonological differences. These were probably caused by *mishearings* rather than *misspellings*. Poor sound quality of the cassette recorder and dialectal differences between speaker and hearer are possible causes. As expected, triphone analysis yielded the best results: not a single misspelling could not be corrected, and only about one out of twenty failed to be returned as the most likely correction.

## 4. CONCLUSION

We have demonstrated that an integration of complementary correction methods performs better than single methods. With respect to orthographical errors, triphone analysis performs better than either grapheme-to-phoneme conversion or trigram analysis alone. Its capacity to correct typographical errors is still to be evaluated, but it is already clear that it will be better than that of SPELL THERAPIST although somewhat worse than trigram analysis in those cases where a typographical error drastically alters the pronunciation. In practice, however, one always finds both kinds of errors. Therefore, it would be interesting to compare the various methods in actual use.

Future research will go into a number of variants on the basic ideas presented here. From a linguistic point of view, it is possible to make the phonological matching less stringent. One way to do this is to use a comparison at the level of phonological features rather than phonemes. However, greater emphasis on orthographical errors may deteriorate performance on the correction of typing errors.

An area of current research is the extension of triphone analysis toward the correction of compounds. In languages like Dutch and German, new compounds such as *taaltechnologie* (*language technology*) are normally written as one word. Correction of errors in such compounds is difficult because the constituting words should be corrected separately but there is no

easy way to find the right segmentation. We have developed some heuristics to solve this problem.

Of course, other combinations of methods are possible. One possibility which looks promising is to combine phonemic transcription with the PF-474 chip. Although triphone analysis is fairly fast, use of the PF-474 chip might further increase the speed. For the correction of large quantities of word material, speed is an essential factor. However, it should be kept in mind that there is a linear correlation between the size of the dictionary and the required processing time, and that the correlation curve is steeper for the PF-474 chip than for triphone analysis. This means that triphone analysis will still be faster for very large dictionaries.

With an eye to commercial applications, TNO-ITI is extending the basic method with data compression techniques and an improved formalism for grapheme-to-phoneme conversion.

## ACKNOWLEDGEMENTS

A prototype of triphone analysis was implemented at the Dept. of Psychology of the University of Nijmegen under ESPRIT project OS-82. Parts of the experiment and the port to the APOLLO were carried out at TNO-ITI, which also developed FUZZIE.

We are indebted to Prof. Dr. Gerard Kempen (University of Nijmegen) and to Adriaan van Paassen (TNO-ITI) for their stimulation of the research and for the helpful comments, and to Hil Weber for typing the paper.

## REFERENCES

Angell, R.C., Freund, G.E. & Willett, P. (1983) Automatic spelling correction using a trigram similarity measure. *Information Processing & Management*, 19, 255-261.

Barendregt, L.G., Benschop, C.A. & De Heer, T. (1985) Subjective trial of the performance of the information trace method. *Information Processing & Management*, 21, 103-111.

Daelemans, W. (1987) *Studies in Language technology*. Ph.D. Dissertation, Linguistics Dept., University of Leuven.

Daelemans, W., Bakker, D. & Schotel, H. (1984) Automatische detectie en correctie van spelfouten. *Informatie*, 26, 949-1024.

Damerau, F.J. (1964) A technique for computer detection and correction of spelling errors. *CACM*, 7, 171-177.

De Heer, T. (1982) The application of the concept of homeosemy to natural language information retrieval. *Information Processing & Management*, 18, 229-236.

Peterson, J.L. (1980) Computer programs for detecting and correcting spelling errors. *CACM*, 23, 676-687.

Pollock, J.J. & Zamora, A. (1984) Automatic spelling correction in scientific and scholarly text. *CACM*, 27, 358-368.

Van Berkel, B. (1986) *SPELTERAPUIT: een algoritme voor spel- en typefoutencorrectie gebaseerd op grafeem-foneemomzetting*. Master's thesis, Dept. of Psychology, University of Nijmegen.

Yianilos, P.N. (1983) A dedicated comparator matches symbol strings fast and intelligently. *Electronics*, December 1983, 113-117.