Statistics and Phonotactical Rules in Finding OCR Errors

Stina Nylander

Uppsala University & SICS

stina@stp.ling.uu.se

Abstract

This report describes two experiments in finding errors in optically scanned Swedish without lexicon. First, statistics were used to find unexpectedly frequent trigrams and correction rules were created. Second, Bengt Sigurds model of Swedish phonotax was used to detect words with phonotactically illegal beginning or end.

The phonotax did not perform as well as the statictic rules did on their training material, but outscored them by far on new text.

A correction tool was created with the phonotax as means of error detection. The tool displays every occurrence of an error string at the same time and gives the user the possibility to give different corrections to each occurrence.

This work shows that it is possible to find errors in optically scanned text without relying on a lexicon, and that word structure can provide useful information to the correction process.

1. Introduction

Optical character recognition (OCR) is a technique for moving text resources from paper medium to electronic form, something that is often needed in our computerised society. Companies and authorities want to make old material machine readable or searchable. Unfortunately, it does not get us all the way. With good paper originals, OCR can achieve 99% of the characters correctly recognised but the result will still contain in average one error word per 20 words which means 5% incorrect words or about one error per sentence (Kukich, 1992). Depending on the application of the optically scanned text, large post processing efforts can be necessary. Since OCR is often used to move large amounts of text to electronic form, the proofreading is a task both demanding and dull. This makes the need for good tools of spell checking and correction large and urgent.

Most spell checkers and OCR post processing systems are lexicon based. A lexicon of reasonable size is used to match against the text, and any word token not in the text is presented as a possible error. Probability scores or similarity measures are then used to generate correction suggestions.

I will concentrate on the error finding process and not try to generate correction suggestions. I want to find ways of proofreading text without relying on a lexicon. Instead I will try to define rules that identify character sequences that are unlikely to be correct word tokens. I made two experiments: using statistical methods to find unexpectedly frequent character sequences, and using phono- or graphotactical rules to find unlikely character combinations. Obviously these results can be generalised for all kinds of proofreading tasks: e.g. handwriting recognition or dictation tasks.

The work described in this report has been done within a Master thesis at the Language Engineering Programme at Uppsala University. The work has been carried out at SICS and was funded by the Digital Library project.

1.1 OCR errors

Many recognition errors are caused by graphical similarity : arguinent (argument), teature (feature), rnean (mean), sernantics (semantics), systernet (systemet), textförstäelse (textförståelse), disambiqueras (disambigueras).

Proofreading by hand is difficult. The graphical errors are by definition difficult to detect by ocular scanning through the text: the visual difference between *bodv* and *body* is very small. Other problems are print quality, font and the age of the original that sometimes produce errors that make it impossible to guess the original word like appTijdo-tiL-s (approaches.) or Umt (that).

Another group of errors that occurs in optical scanning of text is split errors; spaces are inserted in a word and produces a number of strings, many of them incorrect: pronunc i at ion (*pronunciation*), i nt e (*inte*), öre I i gger (*föreligger*).However, many or even most of these errors still produce string tokens that are unlikely or impossible words in the language under consideration.

1.2 Approaches to Error Correction

Most approaches to correction of scanning errors are lexicon based. A lexicon of reasonable size is used to match against the text, and any word token not in the lexicon is defined as incorrect. This leads to many false alarms, since a lexicon never can cover everything. Many correct words and proper names will be presented as errors by the system. To find real word errors -- i.e. errors that result in another correct word -- sequences of parts of speech are evaluated for likelihood of occurrence, and unlikely sequences are marked as possible errors.(Meknavin et al., 1998; Tong & Evans 1996; Huismann, 1999).

The research made shows that OCR post processing problems are highly language specific, Meknavin et al. show that one of the biggest problems working with Thai is to find the word boundaries (1998), while those working with English put the largest effort in spotting the errors (Tong & Evans, 1996; Golding & Shabes, 1996), or providing good correction suggestions (Takahashi et al., 1990). Lee et al. argues that the Korean writing system is syllable based and that recognition and error correction therefore should be syllable based rather than character based (1997), and Hogan points out that if you work with minority languages, in his case Haitian Creole, it is not likely that you even use OCR software developed for your language, which makes post processing even more necessary.

We want to find ways of proofreading text without relying on a lexicon by finding character sequences that are unlikely to be correct word tokens. We tried two experiments: using statistical methods to find unexpectedly frequent character sequences, and using phono- or graphotactical rules to find unlikely character combinations.

2. Statistics

The hypothesis is that differences in observed frequency between correct text and optically scanned text for a character n-gram would indicate that the n-gram in question was incorrectly recognized by the scanning process.

The NoDaLiDa conference proceedings were selected as experimental material. The proceedings contain both correct text as provided by the author in machine readable form and optically scanned text. Two optically scanned papers were handcorrected to obtain testing material: *How Close Can We Get to the Ideal of Simple Transfer in Multilingual Machine Translation (MT)?* (Andersen, 1989) of about 2500 words, henceforth NODA89-09, and the first part of *A self-extending lexicon: description of a word learning program* (Ejerhed & Bromley, 1985) of about 1800 words, henceforth NODA85-06. The statistical experiment was made on English text material, since we do not have enough Swedish text provided in machine readable form to establish frequencies for the correct text.

Since it is necessary to be able to count the number of errors automatically in studies like this, proofreading and error counting by hand is simply too costly in time, a simple error measure was defined for the experiment: the number of errors is the difference in number of word tokens between the optically scanned text and the corrected text plus the number of strings that only appeared in the optically scanned text. The real word errors would not be included in the resulting number of errors and split errors would be counted as the number of parts the original words was split into.

The n-gram frequencies of the corrected text and the optically scanned text were compared and n-grams that showed large frequency differences between text versions were displayed to the editor, together with a concordance of all the occurrences of the n-gram. This allowed the editor to formulate a correction rule for the n-gram under consideration.

Two sets of rules were formulated for each article, one with rules that replaced a character trigram with another string of optional length, the other with rules that replaced a string of optional length with another. The rules that rewrite trigrams were generated with the support of a graphical tool that generated a list of suspect trigrams, for each trigram showed a concordance of all the occurrences of the trigram, and, with a correction given by the user, could generate a correction rule. The rules that rewrites longer strings were generate by hand.

The rules were then used to correct both the article that had been used to generate the rules and the other, to see if the rules were useful in another context than the one they had been generated in.

The number of errors were counted by means of a perl program before and after correction to estimate the performance of the rules. The number of errors generated in the correction process were counted separately to keep track of over correction. As generated errors were considered strings that appeared only in the version corrected with the rules, neither in the correct version nor in the optically scanned version. Over corrections that result in correct words (or already present errors) will thus not be counted.

The tests described above show that while rules based on observed frequencies of character sequences do provide a noticeable improvement on the training material and presumably will be useful for proofreading a given text, they are too specific for use on other material. When it comes to the trigram rules the problem could be that the trigram as context is too small and that many errors, and even more corrections, affect more than three characters, but even if the rules that treated longer strings worked a little better on unknown errors, they were much too text specific too. All the rules deal more or less with a given error in a given word, which is even more true for the rules that rewrote strings of optional length: the longer string that is rewritten, the less generic is the rule. This approach needs a very large text material to generate the rules from and a huge number of rules to be of any significant use in correcting new texts.

3. Graphotactical Rules

Instead of trying to find potential error strings by computing frequency data and comparing correct text with optically scanned text, we used Bengt Sigurd's model of Swedish phonotactics (1965). The model was adapted to graphemes, i.e. the phoneme /ʃ/ was replaced by all the different Swedish spellings (sk, sj, stj, skj etc.) and the same for the phonemes /j/ and /ç/. The SUC corpus (Källgren, 1990) was used to check which vowels that followed the different spellings of these phonemes, *stj* being followed only by *ä*, *ch* by *a,o,e,i*, etc. This added 63 initial consonant combinations to Sigurds 55. The legal initial vowel clusters were listed empirically by extracting all words from SUC that started or ended in a sequence of vowels and added 22 initial sequences to the model. Altogether this gives around 530 different initial consonant sequences + vowel or initial vowel sequences.

Sigurd has 102 primary final consonant combinations. To this has been added 11 combinations to cover the Swedish final doubling of consonants after long vowel, *-x* and *-xt* as final consonant clusters since *x* is realized as two phonemes, /ks/, and is thus as letter not a part of Sigurds description (Benny Brodda even says that the letter *x* has no place in a phonological description of Swedish (Brodda, 1979)). The legal final vowel clusters were listed empirically by extracting all words from SUC that ended in a sequence of vowels. With these adaptions made, the model could be used to find strings beginning or ending with illegal consonant clusters.

In addition rules were added to find strings with consonants only, strings with mixed alphanumeric characters, strings with mixed case and strings with punctuation characters in non-final positions. With these adaptations made, the model could be used to find strings beginning or ending with illegal consonant clusters.

Breaches of the graphotactical rules were extracted from the text and marked as potential errors. The user was displayed the list of possible errors, asked to suggest a correction, and to mark which occurrences should be corrected.

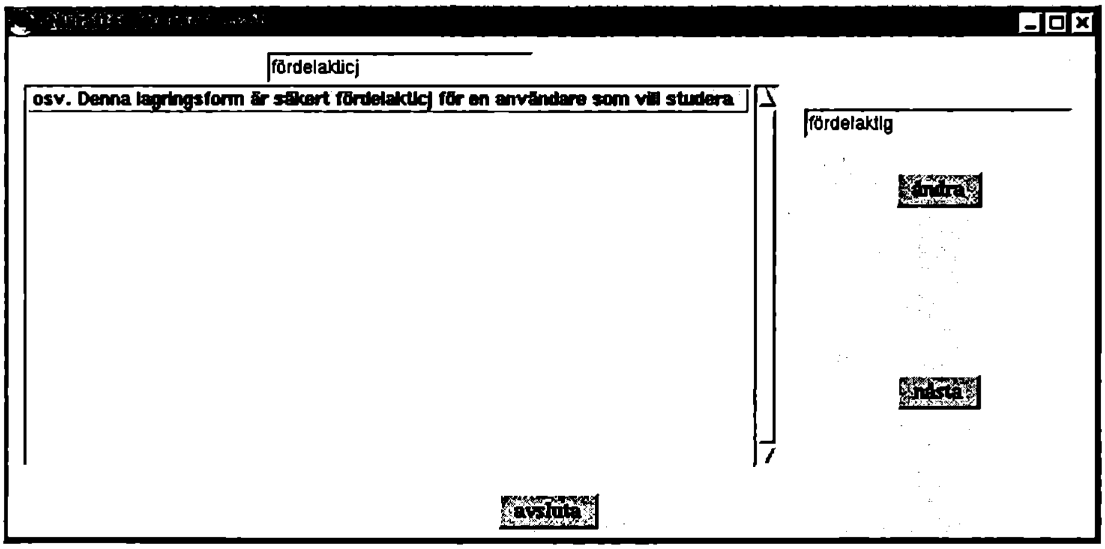
A corpus of optically scanned Swedish text containing 71 000 words was scanned for graphotactical clashes. We found 2495 words with possible errors (822 words with illegal prefixes, 737 with illegal suffixes, 336 words containing punctuation marks or other special characters and 600 words that mixed letters and digits or upper and lower case characters). Of these, many are abbreviations, foreign words, or correct Swedish words with unusual spelling.

State-of-the-art OCR systems give a result of up to 99% correct character recognition, which gives on average one error per 20 words (Kukich, 1992). One error per 20 words would for our corpus give 3550 errors, and while we found 2495 possible errors, about 370 of them were abbreviations (which could easily be filtered out), about 75 were correct non-compound words, and about 100 were acceptable alphanumeric combinations. This leaves us about 1900 likely errors -- a precision of around 75% at a recall of more than 50% if the error estimate holds! And the graphotactic rules can be improved -- at the moment they only deal with initial and final clusters.

The rules were also tested on a single article, *Inte bara idiom* containing 2200 words (Allén, 1983), corrected by hand. The article contained 89 errors, 4% of the total number of words, of which 24 were real word errors and 1 was a split error, both error types that the rules can not handle. The rules presented 42 possible errors of which 19 were errors, 1 was a correct Swedish word, 14 were correct foreign words and 8 were abbreviations. This means that, when the abbreviations have been filtered out, the rules managed to find 29% of the non word errors and only presented one correct Swedish as an error.

4. Implementation

The implementation of the graphotactical experiment described above is a correction tool for Swedish (html) text written in Perl/Tk. The program detects non word errors with the phonotactical rules described above and enables the user to proofread a text in a non linear way. All occurrences for each possible error are displayed to the user at the same time, with a small context. This gives the user the possibility to decide if all the occurrences really are to be corrected, or if one occurrence is correct (maybe an unusual abbreviation or an acronym). Each occurrence can then be given a different correction if necessary.



fördelaklicj

osv. Denna lagringsform är säkert fördelaklicj för en användare som vill studera

fördelaklig

4.1 Possible Improvements

At the moment there is no possibility of undoing a made correction, since the program does not keep track of where the correction is made in the text. The program can not find the position in the text again, and thus can not undo the correction. Another consequence of this is that the user can do only one correction per occurrence.

It is not possible to go back to the previous error word and to see the concordance over that word again. The program does not keep the error words and can thus not go back and reconstruct the concordance.

When correcting an error the user should be able to change the scope of the error. If the program presents *översättn* as a possible error and the context looks like *översättn ing*, the user should be able to mark *översättn ing* as the error string and replace it with the correct string *översättning* without space. At the moment the user can not change the scope of the error string, thus split errors cannot be corrected even when observed by the user.

5. Discussion

The above experiments show that it is possible to find errors in text without relying on a lexicon, and without the large numbers of false alarms we have learnt to expect from such systems. And -- which should not be surprising to those of us who are linguists! -- it is also clear that knowledge of the structure of words improves the results. The phonotactical rules might not reach the same recall as a lexical error finding approach, strings that do not violate the Swedish phonotactics might still be non words, for example *sernantik* (semantik) and *systernet* (systemet). The precision of this method although, will spare the user many of the false alarms and still clean up the text from a substantial part of the recognition errors.

References

Allén, S. 1983. Inte bara idiom. In Proceedings of the 4th Nordic Conference on Computational Linguistics. Uppsala.

Andersen, P. 1989. How Close Can We Get to the Ideal of Simple Transfer in Multilingual Machine Translation (MT)? In Proceedings of the 7th Nordic Conference on Computational Linguistics. Reykjavik.

Ejerhed, E. & Bromley, H. 1985. A self-extending lexicon: description of a word learning program. In Proceedings of he 5th Nordic Conference on Computational Linguistics. Helsinki.

Hogan, C. 1999. OCR for Minority Languages. In Proceedings of the 1999 Symposium on Document Image Understanding Technology. Annapolis, Maryland.

Huismann, G. 1999. OCR Post Processing. Groningen University. Groningen.

Kukich, K. 1992. Techniques for Automatically Correcting Words in Text. In ACM Computing Surveys, Vol. 24, No. 4, 377-439.

Källgren, G. 1990: ``The first million is hardest to get": Building a Large Tagged Corpus as Automatically as Possible. In Proceedings of COLING 90. Helsinki.

Lee, G., Lee, J-H., Yoo, J. 1997. Multi-level post processing for Korean character recognition using morphological analysis and linguistic evaluation. Pattern Recognition 30(8): 1347 - 1360.

Meknavin, S., Kijsirikul, B., Chotimonkol, A. Nuttee, C. 1998. Combining Trigram and Winnow in Thai OCR Error Correction. In Proceedings of COLING 1998. Montréal.

Sigurd, B. 1965. *Phonotactic Structures in Swedish.* Lund University. Lund.

Takahashi, H., Itoh, N., Amano, T. & Yamashita, A. 1990. A Spelling Correction Method and Its Application to an OCR System. Pattern Recognition vol 23 3/4.

Tong, X. & Evans, D. 1996. A Statistical Approach to Automatic OCR Error Correction in Context.