Large Scale Experiments on Correction of Confused Words

Jin Hu Huang
School of Informatics and Engineering
The Flinders University of South Australia
huang@ist.flinders.edu.au

David Powers
School of Informatics and Engineering
The Flinders University of South Australia
powers@ist.flinders.edu.au

Abstract

This paper describes a new approach to automatically learn contextual knowledge for spelling and grammar correction – we aim particularly to deal with cases where the words are all in the dictionary and so it is not obvious that there is an error. Traditional approaches are dictionary based, or use elementary tagging or partial parsing of the sentence to obtain context knowledge. Our approach uses affix information and only the most frequent words to reduce the complexity in terms of training time and running time for context-sensitive spelling correction. We build large scale confused word sets based on keyboard adjacency and apply our new approach to learn the contextual knowledge to detect and correct them. We explore the performance of auto-correction under conditions where significance and probabilty are set by the user.

1. Introduction

In many applications it is necessary to correct errors that have been introduced by human typists and operators, including non-native speakers, or by Artificial Intelligence systems such as Speech Recognition or (Optical or Handwritten) Character Recognition, or even by Machine Translation.

Errors that simply involve non-words being generated can very easily be discovered by looking up a dictionary, but such simple Spell-Checkers are inadequate to the extent that they cannot pick up errors which involve substitution of another valid word, or which involve grammatical errors. We [1] distinguish six different types of reasons for substituted word errors: typographic error ('form' versus 'from'), homophone error ('peace' and 'piece'), grammatical error ('among' and 'between'), frequency disparity errors, learners' errors and idiosyncratic error. These errors account for anywhere from 25% to over 50% of observed spelling errors [2]. Fixing these kinds of errors requires analyzing the contextual information and is not handled by conventional spell-check programs. The task of fixing these spelling errors that happen to result in valid words is called context-sensitive spelling correction. Note however, that all spelling correction is context sensitive – the difference with confused words is that the identification of spelling errors is also context sensitive.

2. Confused Words

Rather than attempting to detect and correct all possible errors, our context-sensitive correction algorithm attempts to choose between known pairs or sets of ambiguous words for which statistics are present at significant levels. The ambiguity among words is modelled by confused sets. A confused set means that each word in the set could mistakenly be typed when another word in the set was intended.

These confused sets can be discovered based on a number of models and sources of errors, including keyboard proximity (typos), phonological similarity (phonos), grammatical confusion (grammos), frequency disparity, foreign idioms, and idiolectic idiosyncrasies. These are often present in combination - in particular frequent words like ' are' are often substituted for less frequent but similar sounding words like ' our' : it seems that our fingers automatically complete the more common confusions of words that are nearby either on the keyboard or phonetically – and they can even complete common endings like ' ing' .

For keyboard proximity, we model which keys are adjacent and thus often substituted, we model omissions of letters, shifting of a pattern left or right on the keyboard, clipping an adjacent key causing an insertion. These models can be used to autocorrect words that aren' t in the dictionary, or can be used with the methods explained below to pick up and correct problems where they happen to produce a valid word.

For phonological similarity, we use a dictionary to map to a phonological representation and then look in a similar way for exact homophones as well as near homophones resulting from substitution, deletion or insertion of a phoneme.

Frequency information also needs to be taken into account as a bias, and we can potentially tune our models at run time to the kinds of idiosyncratic errors that are

frequently made by an individual - taking note of the corrections that they make themselves as they type or on subsequent proof-reading.

There are also databases/corpora of common errors made by second-language learners, e.g. foreign speakers of English. This information can be treated in the same way as the sets of words discovered using the above models, and indeed there are also models explaining the type of errors made by language learners of a specific linguistic/cultural background.

3. Context-sensitive spelling correction

The general problem considered in context-sensitive spelling correction is the resolution of lexical ambiguity, both syntactic and semantic, based on the features of the surrounding context. Two kinds of features have been shown useful for this: context words and collocations. Context words test for the presence of a particular context word within $\pm n$ words of the ambiguous target word. The context words capture the semantic atmosphere (discourse topic, tense, etc.). Collocations test for a pattern of up to m contiguous words and/or part of speech tags around the target word. Collocations capture local syntax.

Previous work has been done based on the combinations of these two types of features. Word-trigram methods [3], Bayesian classifiers [4], Decision Lists [5], Bayesian hybrids [6], Winnow-based methods [7] and transformation-based learning [8] have gradually improved the accuracy of the context-sensitive spelling correction. But in obtaining the collocations most need to use a dictionary to tag each word in the sentence with its set of possible part-of-speech tags, which increases the complexity of the system in terms of both training time and the running time, whilst those that use words directly are limited to trigram statistics due to the exponential explosion of possibilities.

Entwisle's [9] parser which uses crude affix information to parse English inspires us to obtain syntactic information only based on sentence form. We use two kinds of word forms to capture the syntax around the target word: the most frequent words and affixes. Kilgarriff [10] shows the most frequent words tend to be syntactic in nature and to be are almost all function words. A vowel or a consonant prefix shows us the ‘a/an’ distinction. Suffixes capture the useful syntactic features. Both the most frequent words and affixes give us the syntactic cues to discriminate the confused words. We define them as eigenunits. Tagging each word around the target word using a dictionary is simply replaced by matching the eigenunit. This significantly reduces the complexity from the order of a million possible tokens per position, to a few hundred.

With the availability of large text corpora, it has become possible to automatically learn the grammatical rules directly from the text, instead of manually generated rules, which can be time consuming. Furthermore it is difficult to generate all syntactic and semantic rules, as the rules of language are vast and idiosyncratic. Learning rules from corpora is more realistic and applicable. Traditional ‘spell-checking’ and ‘grammar-checking’ tend to use fixed rules of thumb which lead them to flag all occurrences of particular words like 'which' or particular constructs like passives or prepositions at the end of sentences. These are deprecated by style manuals, but are very commonly used and not really wrong.

4. Experiment and result

The Wall Street Journal (1987-1992 - WSJ) and the Lewis Carroll's novel Alice's Adventures in Wonderland (Alice) were used in this experiment. Around 71.6M words (WSJ87-89, 91-92) were used for training and 1990 WSJ was used for testing.

The first phase of the project involved developing the initial sets of confused words - primarily for the modeled typographical errors. Peterson [11] shows that up to 15% of typographical errors yield another valid word in the language. We extracted 7,407 pairs of confusable words based on 6,131 words from the 25,143-word Unix dictionary by systematically performing character insertion, deletion or transposition. These include the following four situations: a) where adjacent keys are substituted such as 'sun' and 'sin'; b) where one character is deleted or inserted such as 'its' and 'it’s'; c) two characters are transposed such as 'form' and 'from'; d) where two characters are adjacent on the key board and are substituted with the wrong pair of adjacent characters such as 'trap' and 'reap'. About 44% of the words in the training corpus belong to these confused words.

The second phase concerns selection of the eigenunits. We use the 145 frequent words and function words plus 65 common suffixes, a dummy null-inflection suffix and the 34 individual non-alphanumeric punctuation characters as our eigenunits. In order to distinguish between ‘a’ and ‘an’, we use vowel and consonant prefix versions of the 66 suffixes. We also classify week, month, ordinal number and cardinal number as separate classes. Irregular word forms can also be usefully added to the eigenunits to reduce to the noise in these but we did not choose to use these as the existing eigenunits cover at least 85% of the training and validation texts (85.6% in the 18.7M 1991 WSJ and 88.4% in the 26.5K Alice corpus).

Once we had the sets of confused words and the eigenunits, the third phase was to develop statistics from a large corpus (5 years of WSJ from 1987-89,91-92). For

each ambiguous word we learn the rules simply by substituting the surrounding words with eigenunits and counting the occurrences of the rules. We gradually extend the window size from 1 to 5 on both sides until a desired degree of significance is reached. We do this to avoid learning rules that are useless because the context is so large that insufficient examples are present to learn from. Based on each context for each confused set, we ignore the context occurring less than a minimum occurrence threshold, currently set to 10, as these occurrences are not sufficient to discriminate confused words reliably. Only where there are more than 10 contexts available do we perform the relatively expensive significance calculation according to Fisher's exact test [12].

Table 1 Diameter with occurrences, significance and probability – number of contexts

	1	2	3	4	5	Total	%
Occurs<10	5601175	3928959	114481	414	2171	9647200	87
Occurs>=10	1020095	398380	13798	151	653	1433077	13
S>=70,P>=70	283959	23002	0	0	0	306961	2.8

Table 2 Relationship between probability and significance – number of contexts

Occurence>=10	P>=0	P>=70	P>=80	P>=90	P>=95
S>=0	1433077	1413030	1399495	1379896	1350122
S>=70	326420	306961	294005	272192	247368
S>=80	291713	272843	261050	242090	220657
S>=90	247499	231416	222023	206704	189346
S>=95	218889	204697	196581	183469	168604

Table 3 False errors and recall testing on test and validation corpora

Corpus	Words	S & P	Errors Introduced	% (E/W)	Confused Words	Confused Sets(CS)	Significant Sets(SS)	Recall(%) (SS/CS)
WSJ0801 (90)	64,718	S,P>=95	199 (unchecked)	0.30	34805	216963	52791	24.3
		S,P>=80	532	0.80			65355	30.1
		S,P>=70	666	1.02			70794	32.6
WSJ1231 (90)	56,163	S,P>=95	169 (2 true errors)	0.30	29594	186812	44587	23.9
		S,P>=80	467	0.83			55543	29.7
		S,P>=70	572	1.02			60435	32.4
Alice	26,457	S,P>=95	156 (11 true errors)	0.60	20082	116640	17467	15.0
		S,P>=80	772	2.92			22464	19.3
		S,P>=70	938	3.55			24472	21.0

Table 4 True errors detected and corrected when errors seeded randomly

Corpus	Errors seeded	S & P	Errors Detected	Detect Rate(%)	Errors Corrected	Correct Rate(%)
WSJ0801(90)	3253	S,P>=95	806	24.8	664	*20.4*
		S,P>=80	*980*	*30.1*	*804*	24.7
		S,P>=70	*1058*	*32.5*	*853*	26.2
WSJ1231(90)	2792	S,P>=95	663	23.7	541	*19.4*
		S,P>=80	*856*	*30.7*	*686*	24.6
		S,P>=70	*919*	*32.9*	*734*	26.3
Alice	1588	S,P>=95	279	17.6	226	*14.2*
		S,P>=80	*364*	*22.9*	*293*	18.5
		S,P>=70	*390*	*24.6*	*308*	19.4

Table 5 Seeded errors of the confusion set of 'from' and 'form' (S,P>=95)

Corpus	Errors seeded	Errors Detected	Detect Rate(%)	Errors Corrected	Correct Rate(%)
WSJ0801(90)	331	267	80.7	227	68.6
WSJ1231(90)	288	245	85.1	214	74.3
Alice	36	18	50.0	16	44.4

From the Tables 1 and 2 we can work out which contexts allow reliable correction and what window size of the contexts best represents the syntactic information which is more significant and useful. From table 1 the diameter 2 is enough to catch the syntax around the targeted word. Golding [6] obtained a similar result indicating that the window size 2 for collocations generally did best to discriminate among words in the confusion set. Table 2 shows that most highly significant contexts are high probability but not vice versa, as expected. High probabilities without high significance are probably not trustworthy. It remains to be seen how best to tradeoff between probability and significance in user trials – some users want to be sure to catch all errors even if that means lots of false corrections are proposed. Others would rather see only errors with a high degree of certainty, viz. high significance and probability.

We record the contexts which are reasonably significant and likely to suggest a correction (S>=70 and P>70). Both significance and probability can be used in defining a function for correction. These statistics based on the surrounding words will be sufficient to give us a context in which one choice is clearly preferred.

We tested our text corrector on two issues from the withheld 1990 WSJ test corpus as well as on a validation corpus of an entirely different genre, namely Alice's Adventure on Wonderland. Initially we did not seed any error into these corpora. Table 3 tells us that our system will introduce around 0.3% false errors on the same genre (WSJ) but introduce 0.6% false errors on the different genre (Alice). With less significance and probability, more false errors will introduce. This shows that our system is a genre oriented as expected, and that our use of significance and probability even at these moderate levels keeps the number of false corrections under control – this is the major problem with conventional systems.

In order to evaluate our system, we collect the statistics for all the confused words occurring on the test and validation corpora. For each of these confused words we count the significant confused word sets of all its possible confused word sets according to the levels of significance and probability. We can only detect and correct the errors occurring on these significant confused word sets. The rate of these significant word sets and all the possible confused word sets is equal to recall. Table 3 shows us that our system can obtain about 24% recall on the same genre but only 15% on the different genre (Alice) at the levels of 95% significance and probability. One reason why we obtain such a low recall is that our confusion sets are rather big and the training corpus is not large enough to learn significant contexts. Our confusion sets include semantic errors such as 'he' and 'she' which are difficult to distinguish using local context alone. Table 5 shows that we get better result of about 80% recall for syntactic errors 'form' versus 'from'.

Another reason for low recall is that irregular forms of the eigenunits particularly for the suffixes distort the contexts around the target word. This distortion also causes many of the false errors introduced by the system. We can decrease the level of significance and probability to increase the recall but it will then introduce more false errors (Table 3).

As seen in Table 4 we actually obtain about 24% detect rate at 95% significance/probability level overall. This coincides with the testing results on Table 3. But we only obtain about 20% correct rate at the same significance/probability level. From Table 4 we know that the system can detect 806 errors at levels of 95% both significance and probability when seeded with 3253 errors on the testing corpus (WSJ0801). Of these 806 errors the system can automatically correct 664(%82) errors. The other 142 errors have two or more proposals to correct them. Further experiment need to be done to find out how many errors of these 142 errors detected can be automatically corrected based on the value of significance and probability of each proposal. According to these statistics it is possible to perform reliable auto-correction.

We now turn to look at accuracy in terms of the false errors from the original corpus. No matter how many true errors are seeded in the corpus, we cannot change these false errors. The more seeded errors, and the higher accuracy we require, the more false errors introduced.

A final issue relating to accuracy is the lack of a psychologically or empirically motivated user-model. At this stage we are using an elementary model that assumes that all errors relate directly to low keyboard or phonological distance, but in fact as discussed above, word frequency, language and ideolectic background play a role, and certain types of errors compassed in our confused words sets are much rarer than our model predicts. We propose to tune this model by obtaining corpora of language learner errors, typographic corrections, and by making use of the statistics for errors which do lead to non-dictionary words to inform our model.

5. Interface

In order to compare our text corrector to Microsoft Spelling and Grammar-checking, we integrated our text corrector into Microsoft Word using Macro, Visual Basic and Access. This is useful for the user in evaluating the performance of the system as well. Microsoft Word can only correct 90 pairs of confused words but our corrector can check and correct 7,407 pairs of confused words. Our text corrector outperforms Microsoft Word in picking up errors but still introduces some new errors. Initially we proposed to use the significance and probability to colour the words so that the words that are more likely to be

wrong are highlighted more strongly but experience with the colour coding in the latest versions of Word indicate that this may confuse or annoy the user and detract from appropriate attention to the significant corrections in the text. At this stage we only display the significance and probability of the alternative to the user in a dialog box when a highlighted word satisfying the significance and probability thresholds is selected.

Note that, as discussed above, there are two types of errors that a spelling corrector always can make: false negatives (complaining about a correct word) and false positives (failing to notice an error), so in order to give the user the opportunity to trade off these two kinds of errors, we allow the user to change the significance and probability at which notification of potential errors occurs. Thus users can decide the balance between being bothered for some false errors and missing some true errors. Normally this is set at a 95% significance level and a precision setting of 95% in the confused set. It is possible to set levels of significance and likelihood for auto-correction to occur in the interface.

6. Conclusion and future work

The technique we developed here can be used to resolve lexical ambiguity in the syntactic sense. It captures the local syntactic patterns but not semantic information as the eigenunits can not represent the semantic association with the target word. For example the word 'cake' maybe is useful to disambiguate the confusion set dessert and desert but 'cake' does not exist in the eigenunits so this association cannot be learned. Furthermore the window size 2 is too small to capture this association. If we extend the window size, keeping the whole environment is not suitable for this semantic purpose. Further work need to be done to exploit this distant word association to generate more efficient algorithm for resolving this problem and minimising the features we learned.

In order to improve the performance of the system, we must handle the noise caused by the irregular words in the eigenunits. As mentioned above this noise did not make the statistic collection worse but it will distort the context around the target word when the correction happens. This is the main cause for the false errors. As the vast confusion sets we have, we can optimise the confusion sets to build a better model through evaluating each confusion set as mentioned in the testing stage.

We expect to be able to reduce the number of false corrections by modelling the kind of errors people actually make in more detail as at present we only use keyboard adjacency.

7. References

[1] David M. W. Powers, "Learning and Application of Differential Grammars", CoNLL97: Computational Natural Language Learning, ACL, Association for Computational Linguistics, Madrid, 1997, pp88-96.

[2] Karen Kukich, "Techniques for automatically correcting words in text", ACM computing survey, 24(4), December 1992, pp377-439.

[3] E. Mays, F.J Damerau and R.L. Merser, " Context based spelling correction", Information Processing and Management, 27(5), 1991, pp.517-522.

[4] William A, Gale, Kenneth W. Chruch and David Yarowsky, "Discrimination decisions for 100,000 dimensional spaces", In Current Issues in Computational Linquistics: In Honour of Don Walker, Kluwer Academic Publishers,1994, pp429-450.

[5] D. Yarowsky, "Decision lists for lexical ambiguity resolution: application to accent restoration in Spanish and French". In Proceedings of the 32 nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, 1994, pp 88-95.

[6] A. R. Golding, "A Bayesian Hybrid Method for Context-Sensitive Spelling Correction", In Proceedings of the third Workshop on Very Large Corpora, Boston, 1995, pp. 39-53.

[7] A. R. Golding and D. Roth, "Applying Winnow to Context-Sensitive Spelling Correction", In Machine Learning: Proceedings of 13 th International Conference, San Francisco, 1996, pp 182-190.

[8] Lidia Mangu and Eric Brill, "Automatic Rule Acquisition for Spelling Correction", In proceedings of International Conference on Machine Learning, Morgan Kaufmann,1997.

[9] Jim Entwisle, PhD thesis, An Investigative Parser for English Using Constraints on Surface Sentence-Form, The Flinders University of South Australia, 1997

[10] Adam Kilgaroff, "Which words are particularly characteristic of a text?" A survery of statictical approaches, ITRI Technical Report, University of Brighton, 1996.

[11] James L Peterson, "A note on undetected typing errors", Communications of the ACM, 29(7), 1986, pp633-637.

[12] Patrick Winston, Artificial Intelligence, 3rd Edition, Addison Wesley, 1993.