Looking back or looking all around: comparing two spell checking strategies for documents edition in an electronic patient record system

Patrick Ruch, Robert H. Baud, Antoine Geissbühler,
Christian Lovis, Anne-Marie Rassinoux, Alain Rivière

Medical Informatics Division, University Hospital of Geneva

{ruch, baud, geissbuhler, lovis, rassinoux, riviere}@dim.hcuge.ch

We report on the comparison of two systems for correcting spelling errors resulting in non-existent words (i.e. not listed in any lexicon). Both systems aim at improving edition of medical reports. Unlike traditional systems, based on word language models, both semantic and syntactic contexts are considered here. Both systems share the same string-to-string edit distance module, and the same contextual disambiguation principles. The differences between the two systems are located at the user interaction level: while the first system is using exclusively the left context, simulating the underlining of every misspellings at the end of every word typing, the second system uses the left as well as the right context and simulate a post-edition correction, when asked by the author. Our conclusion shows the improvements brought by the second approach.

INTRODUCTION

In clinical or general practice, physicians have to document the patient file using free text. They are looking for quality description, but are parsimonious on time spent for document edition.. The systems we design aims at correcting errors resulting in non-existent words. If the majority of words are recognized by search in a dictionary, the misspelled words are difficult to cope with, because they are generally not known in advance. Misspellings are organized following three steps. The first module is based on a context independent string-to-string edit distance calculus (cf. [1] for a survey of the probabilistic models of pronunciation and spelling). The second module, based on the morpho-syntactic context, attempts to rank more relevantly the data set provided by the first module, finally a third contextual module to process words with the same POS (part-of-speech, also called morpho-syntactic (MS) category, i.e. *verb*, *noun*, *adjective*...) by applying contextual WS (word-sense, as for example *body part*, *temporal concept*...) disambiguation. If the string-edit distance calculus is the same in both systems, the nature of the context is different. In the first system (called LCS: left context system), only the two words before the misspelled word are considered, while the second system (called LRS: left and right system) relies also on the two words following the misspellings. A final evaluation shows the improvement brought by the system LRS compared to the RCS system.

From an applicative point-of-view, spelling-correction in medical patient records (as reported in [2], rates of misspelling in medical texts -up to 10%- are incomparable to misspellings rates in other corpora, such as newspaper samples) constitutes a critical issue, likely to result in dramatic side effects. It has been extensively studied and reported in medical literature (see for example [3], [4]). These studies conclude that automated measures of similarities between medication names –and any other types of medical appearing in medical reports- can diminish significantly care-providers errors.

Spelling correction problems can be divided into two categories. The first category addresses the problem of correcting spelling that result in valid, though unintended words (as for example[1] in *a peace of cake*, where *piece* is misspelled) and also the problem of correcting particular word usage errors (such as *among* and *between*). The second category is concerned only with errors that result in words that cannot be found in a lexicon. While the first problem is sometimes referred as *context sensitive spelling correction*, with numerous studies (see [5][6][7]), as opposed to the second, referred implicitly *as context free spelling correction*, often perceived as a problem where progress can not be made[2] [8], some works showed the importance of the context for improving accuracy of the second category too [9][10]. At this level, we suggest a new terminology for qualifying

[1] The experiment was conducted on French corpora, however when possible, examples are provided in English for the sake of clarity.

[2] However, the problem is still very crucial for agglutinative languages [11], where the vocabulary can be hardly listed in an exhaustive manner. Although some authors [12] underlined the high compositionally of the medical language even for morphologically poor languages such as French and English, the French medical language will be considered exhaustively listed in a manageable size list of words (i.e. about 10^5 entries).

1067—5027/01/$5.00 © 2001 AMIA, Inc.

each category: we will call *word correction* the first category, and *character correction* the second, leaving the *context sensitivity/insensitivity* question open for both correction types.

BACKGROUND

While recent experiments on word correction use more linguistic modules (mainly POS disambiguation tools as in [5]) for handling the context, we observe that character correction tools -even when they use the context- do not use comparable approaches, and rely on word language models ([9], [10]) instead. The first specificity of our system consists in applying morpho-syntactic disambiguation to the character correction problem.

Working on a word correction problem, Golding and Shabes [5] introduce a method using POS trigrams to encode the context. Although this method greatly reduces the number of parameters compared to methods based on word trigrams[3], it empirically appeared to discriminate poorly when words in the confusion set have the same POS. In this last case, the method is coupled with a more traditional word model. We also started filtering with a POS tagger, but then, instead of using an expensive word language module, we use a WS tagger for discriminating among candidates, which have the same POS.

Syntactic correction

Another related promising way of research concerns syntactic correction. Syntactic correction addresses a) word order/presence, and b) agreement problems:

a. We decided operate the patient *to*. *
   We decided to operate the patient.
b. They starts the treatment. *
   They start the treatment.

Of course, character correction and word correction may be necessary for processing a correct syntactic correction, therefore in such systems, the usual processing is: first, a string edit module solves the character errors, and second a syntactic module looks for syntactic errors [13]. We decided to apply syntactic constraints at the character correction level!

String-to-string edit distance

In parallel with improving the context-based disambiguation, some other experiments on character correction deal with improving the string edit distance operation ([9] [14] [15]). This central question shall not be treated here: first, because it would go far beyond the scope of the paper, second because all these investigations require large amount of training data (as for example [15] worked with a 3 millions word corpus for speech recognition).

Balancing act

The morpho-syntactic and semantic filtering (combining hand-crafted rules and Hidden Markov Model) can be seen as a winner-takes-all process, where only the most reliable part-of-speech candidates are given more weight, similarly to what occurs in a decision-list system. Like numerous Bayesian approaches, decision lists [16] have been successfully applied to a wide range of problems, including lexical ambiguity resolution. Unlike standard Bayesian approaches, however, a decision list does not combine the log-likelihood of each classifier, but bases its classification solely on the most reliable piece of evidence identified in the target context. Its major advantage is perhaps to gather multiple classifiers, operating on non-independent sources of evidence, in a unified and traceable framework. Thus, in the context of developing a spelling checker tailored for medical texts, using both hand crafted and data-driven sources of evidence, together with facing sparse data issues, such architecture seems particularly well adapted.

This balanced architecture for disambiguation: rules and transition probabilities rather than log-likelihood combination constitutes the last originality of our character correction system. The rule-based part-of-speech tagger we used, as well as its HMM (Hidden Markov Model) component have been extensively described elsewhere ([18] and [19]), and will not be presented here, instead we will present the application of the tool to the character correction task. The semantic filtering behaves along the same lines as the morpho-syntactic one, and has also been described in detail together with the 40 UMLS-based semantic types it uses ([20] and [21]).

Left context vs. right and left context

As in MS-Word, spell checking can be provided via two modes: the first one is fully interactive, it underlines in red color every misspelled word; the second one (tools >> spelling and grammar) starts from the beginning to the end of the document, when

[3] POS n-grams represent the morpho-syntactic level, word n-grams represent the token level, and WS n-grams the semantic level, thus the phrase *we discover* can be represented by 3 different models: *we discover* (word level), *prop v[12]* (morpho-syntactic level), and *pers diap* (semantic level). The meaning of prop, v[12], pers, and diap is respectively: personal pronoun, verb 1st and 2nd person, human being (UMLS T016), and diagnostic procedure (UMLS T060). POS tags attempt to follow the MULTEXT morpho-syntactic description.

asked by the author. While the first mode is certainly the most common it appeared to be less attractive when contextual correction becomes available.

METHOD

We first collected a set of misspelled words together with the left and right adjacent context (+/- 2 words), and got a total of 424 records. This set is split into two equivalent subsets, set A is used for tuning the system, while set B is kept as final test set. This collection step was carried out manually and semi-automatically, and will be the subject of a future report. Some examples of the misspelled words are given here:

LC2	LC1	MW	RC1	RC2	WFW
find	an	uncer	in	the	ulcer
.	Your	patint	has	been	patient

Example 1: Records of the misspelled word database

We used the following lexical resources[4]: a 90000 items list of well written tokens for the string-to-string edit distance module, a lexicon with 30000 lexemes for the POS filtering [12], among these lexemes about 20% are provided with a semantic type (the semantic classes follow and sometimes extend the UMLS semantic network).

String-to-string edit distance calculus

Modern spelling checkers[5] are usually based on a variant of the Levenshtein-Damerau distance. Most misspellings can be generated from correct spellings by a few simple rules. Damerau [22] indicates that 80 percent of all spelling errors are the result of:

a. transposition of two adjacent letters: he*ap*titis (err1)
b. insertion of a letter: hep*p*atitis (err2)
c. deletion of a letter: hepattis (err3)
d. replacement of a letter by another one: hepat*o*tis (err4)

In the standard model, each of these operations cost 1 unit, i.e. the distance between err1, err2, err3, err4 and the word *hepatitis* is 1, while the distance between *hepatitis* and *heppatotis* is 2 (one replacement + one insertion). However, more accurate models, where each operation might have a more specific cost, depending on the letter have been developed [14]. The error model, we developed, includes such refinements. Thus, if the default replacement operation has a one unit cost, more probable replacements (a frequent confusion is for example the letter set {ï, i}) will be weighted less expensively. The cost matrix was trained manually by using regression tests on set A.

Contextual filtering

After processing by the edit distance module, each candidate word comes out with a score. This score expresses the distance between the candidate and the misspelled word. The two next modules are applied sequentially in order to get a more optimal ranking of the candidates.

It is important to notice that if one word within the candidate set is not provided with a POS tag, then the following filters (POS and WS) are not applied. Similarly, the WS filter is not applied if one of the candidates is provided without WS tag. This caution is important in order not to select a priori the words listed in our lexicon vs. words appearing in the 90000 items list.

Part-of-speech filtering

The goal of this module is to modify the edit-distance scoring by combining the morpho-syntactic information brought by the context. Let us consider a misspelled word in context together with a short list of likely candidates. List 1 provides the list as returned by the MS-Word 2000 spell checker, while List 2 shows what would be expected if MS-Word would use the left adjacent context (Fig. 1):

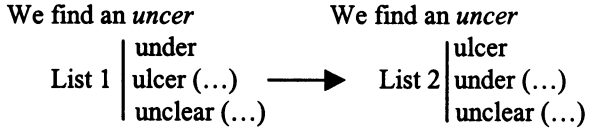

Fig. 1: Example of part-of-speech filtering

In this example, it is clear that list 2 provides a more accurate ranking than list 1 for the misspelled string *uncer*, if we consider the adjacent left context: a determiner like *an* cannot be followed by a preposition like *under*.

The POS tagger attributes one part-of-speech (expressed by a tag) to every token. Thus, in the above example, and after a lexical access (Fig. 3) the tool provides the following top candidate list (Fig. 2):

We find an *uncer* → ulcer
prop v[12] d[s] nc[s]

Fig. 2: POS disambiguation (after POS tagging)

We find an *uncer*

[4] All these resources allow a correction time of less than 200 ms. See [8] for optimization strategies.

[5] Alternative approaches include n-gram distances and similarity keys, cf. [23].

```
prop v[12]  d[s]  sp      (under)
                  nc[s]   (ulcer)
                  a       (unclear)
```
Fig. 3: POS lexical ambiguity (before POS tagging)

In the above figures, the candidate(s) with the tag *nc[s]* are factorized to get ranked closer to the misspelled word. Here is the meaning of the POS tags:

- prop: personal pronoun
- v[12]: verb first or second person
- d[s]: d[s] determiner singular
- nc[s]: common noun singular
- sp: preposition
- a: adjective

Word-sense filtering

In the following example, the part-of-speech does not provide any discrimination rule between the candidates, as both have the same part-of-speech (*nc[s]*). However, the semantic left adjacent context can operate as a discriminator, indeed *until the* is to be followed by some temporal concept (*temp*), like for example *summer*, rather than by some human person (*pers*), which is the tag of *swimmer* (Fig. 4):

```
Until the | swmmer          Until the | swmmer
          | swimmer    ⟶             | summer
List 3    | summer       List 4     | swmmer
```
Fig. 4: Example of word-sense filtering

When processing the above sentence, and after lexical access (Fig. 6), the word-sense tagger returns the following top candidate list (Fig. 5):

```
Until the  swmmer      summer
rtemp      temp   ⟶
```
Fig. 5: WS disambiguation (after WS tagging)

```
Until the  swmmer
rtemp      temp    (summer)
           pers    (swimmer)
```
Fig. 6: WS lexical ambiguity (before WS tagging)

Meaning of WS tags in the above figures:

- rtemp: temporal relations (UMLS T136)
- temp: temporal concept (UMLS T079)
- pers: human, person (UMLS T016)

Left and right filtering

In spite the relevance of the above examples, they are many cases where the left context does not provide enough disambiguation evidence for improving the ranking of candidates, therefore the second system uses both left and right window. We will give only a couple of examples for such cases. The following example capitalized on the right syntactic context in order to rank the candidates more relevantly:

```
The patientt are eating...     The patientt are eating...
     patient          ⟶            patients
     patients                      patient
```

In some other cases, both left and right contexts are necessary, while sometimes nothing in the context (whether syntactic or semantic) can help, as in: *The patientt showed that...*

RESULTS AND CONCLUSION

	SSE	LCS	LRS
top-1	87.0	90.5	94.5
top-3	95	97.5	98.1
top-5	97.5	97.5	98.8
Top-7	98.1	98.1	98.8
Top-20[6]	98.8	98.8	98.8

Tab. 1: Results of the evaluation on the set B (%)

Table 1 provides the results of the evaluation on set B. The string-to-string edit (SSE) distance is taken as a baseline for assessing the improvement brought respectively by the LCS and the BCS.

Five types of measures are provided: top-1, when the well-formed token is provided at the top of the returned list; top-3, when the well-formed token appears within the three first items of the returned list, etc...

In comparison to the string-to-string edit (SSE) output, results are quite encouraging for any kind of contextual filter. But the extended context (left and right) clearly outperforms the other strategies.

In conclusion, we reported on the construction of an original approach for spelling correction using the morpho-syntactic and semantic context, and showed a significant improvement of the correction performances when using both left and right contexts. We plan to combine the contextual filters reported here with more traditional word language models ran on large collection of text in order to improve these very promising results. Another way to investigate will be the spelling correction without user-interaction as it may be necessary in information retrieval systems.

[6] The best score is 98.8% as 2 items out of the 212 records in the set B were not listed in the list of 90000 words.

References

1. Jurafsky D. and Martin J.H.: Speech and Language Processing, Prentice Hall. London.
2. Hersh W.R., Campbell E.M., Malveau S.E.: Assessing the feasibility of large-scale natural language processing in a corpus of ordinary medical records: a lexical analysis.. Proc AMIA Annu Fall Symp (United States), 1997, p580-4
3. Lilley L.L., Guancy R.: Sound-alike cephalosporins. How drugs with similar spellings and sounds can lead to serious errors. Am J Nurs (United States), Jun 1995, 95(6) p14
4. Lambert B.L.: Predicting look-alike and sound-alike medication errors. Am J Health Syst Pharm (United States), May 15 1997, 54(10) p1161-71
5. Golding A.R., Shabes Y.: Combining Trigram-based and Feature-based Methods for Context-Sensitive Spelling Correction. In Proc. of the 34th Annual Meeting of the ACL, Santa Cruz, (1996) p. 71-78.
6. Golding A.R., Roth D.: Applying Winnow to Context-Sensitive Spelling Correction. In Proc of ICML (1996): p 182-190.
7 Mangu L., and Brill E.: Automatic Rule Acquisition for Spelling Correction. In Proc. of ICML, (1997).
8 Peterson, JL.: Computer Programs for Detecting and Correcting Spelling Errors. Computer Practices, Communications of the ACM (1980), vol. 23, number 12.
9 Brill E. and Moore R.C.: An Improved Error Model for Noisy Channel Spelling Correction. Proc. of the 38th Annual Meeting of the ACL, Hong-Kong (2000).
10 Mays E., Damereau F., Mercer R.L.: Context based spelling correction. Information Processing and Management, 27(5), (1991), p. 517-522.
11. Oflazer, K.: Error-tolerant Finite State Recognition with Applications to Morphological Analysis and Spelling Correction. Computational Linguistics (1996), 1-18. Association for Computational Linguistics Eds.
12. Baud R., Lovis C., Ruch P., Rassinoux A.-M.: A Toolset for Medicl Text Processing, in Medical Infobahn for Europe, Proc. of MIE'2000. A. Hasman, B. Blobel, J. Dudeck, R. Engelbrecht, G. Gell, H.-U. Prokosh (eds). IOS Press. (2000).
13. Courtin J., Dujardin D., Kowarski I., Genthial D., De Lima V.L.: Towards a complete detection/correction system. Proc. of the ICCICL, Penang, Malaysia. (1991), p. 158-173.
14. Church K.W., Gale W.A.: Probability scoring for spelling correction. In Stat. Comp. 1., (1991) p. 93-3.
15. Ristad E., and Yanilos P.: Learning String Edit Distance. Int. Conf. on Machine Learning, Morgan Kaufmann. (1997).
16. Rivest R.L.: Learning Decision Lists, in Machine Learning, 2, (1987) 229-246.
17. Yarowsky D.: Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French. In Proc. of ACL (1994), p. 88-95.
18. Ruch P., Baud R., Bouillon P., Rassinoux A.-M., Robert G.: Tagging medical text: a rule-based experiment, in Medical Infobahn for Europe, Proc. of MIE'2000. A. Hasman, B. Blobel, J. Dudeck, R. Engelbrecht, G. Gell, H.-U. Prokosh (eds). IOS Press. (2000).
19. Ruch P., Baud R., Bouillon P., Robert G.: Minimal Commitment and Full Lexical Disambiguation: Balancing Rules and Hidden Markov Models. In Proc. of CoNLL-2000 (ACL-SIGNLL). Lisbon. ACL (ed). (2000), p.111-115.
20. Ruch P., Baud R., Bouillon P., Rassinoux A.-M., Scherrer J.-R., MEDTAG: Tag-like Semantics for Medical Document Indexing. In Proc. of the AMIA'99 Annual Symposium. Washington. (1999).
21. Bouillon, P., Baud R., Robert G., Ruch P., Indexing by statistical tagging. In Proc. of the JADT'2000. Lausanne. (2000).
22. Damereau, F.J.: A technique for computer detection and correction of spelling errors. Commun. ACM, vol. 7, number 3. (1964)
23. Pollock J.J., Zamora A.: Automatic spelling correction in scientific and scholarly text. Computer Practices, Communications of the ACM (1984), vol. 27, number 4.