Spelling correction in Clinical Notes with Emphasis on First Suggestion Accuracy

Jon Patrick, Mojtaba Sabbagh, Suvir Jain, Haifeng Zheng

Health Information Technology Research Laboratory
School of IT, University of Sydney
jonpat@it.usyd.edu.au, ssab8677@uni.sydney.edu.au, suvir@cs.usyd.edu.au, hzhe6571@uni.sydney.edu.au

Abstract

Spelling correction in a large collection of clinical notes is a relatively lesser explored research area and serious problem as typically 30% of tokens are unknown. In this paper we explore techniques to devise a spelling suggestion generator for clinical data with an emphasis on the accuracy of the first suggestion. We use a combination of a rule-based suggestion generation system and a context-sensitive ranking algorithm to obtain the best results. The ranking is achieved using a language model created by bootstrapping from the corpus. The described method is embedded in a practical work-flow that we use for Knowledge Discovery and Reuse in clinical records.

1. Introduction

Our work specializes in processing corpora of medical texts ((Wang and Patrick, 2008), (Ryan et al., 2008), (Zhang and Patrick, 2007) and (Asgari and Patrick, 2008)). These corpora usually contain many years of patient records from hospital clinical departments. Discovered knowledge about the texts needs to be reused immediately to automate certain language extraction tasks that support the workflow of staff in the hospital.

However, to be able to extract the required knowledge, first various lexical verification steps need to be undertaken, such as expansion of abbreviations and acronyms, recognition of named entities, drug dosage details etc. An initial analysis of a corpus is to identify unknown words and non-words (tokens containing any non-alphabetic characters) which typically constitute 30% of a corpus. A large majority of unknown words are misspellings but a significant number (10%-15%) are meaningful and need to be correctly interpreted.

The nature of the requisite processing techniques is more complex than simple spelling correction because the medical texts also contain various abbreviations, acronyms, names of people, places and even a number of neologisms. In such a text, every word that does not appear in a dictionary need not be an invalid word. Therefore, during the process of knowledge discovery, extreme care has to be taken that valid tokens are not misclassified as spelling errors. For abbreviations, acronyms and named entities, we apply separate workflows for their detection and resolution.

Our study is based on those words in the corpus that have been designated by medical experts as genuine spelling mistakes. These words include both medical and non-medical words. The diverse nature of these words renders traditional spell checkers ineffective and requires a novel spell checking system for their resolution.

Currently the nature of spelling correction of this system is 'offline', i.e., these suggestions are not being generated during entry of data in the hospitals. Instead, these methods are applied to the corpus as a whole. We look to implement this system as an 'online' process in the near future.

Our knowledge base to begin the knowledge discovery includes the following resources:

1. Systematised Nomenclature of Medicine-Clinical Terms, i.e., SNOMED-CT: 99860 words
2. The Moby Lexicon: 354992 words
3. The Unified Medical Language System, i.e., UMLS: 427578 words
4. Lists of abbreviations and acronyms discovered in previously processed clinical corpora: 1483 words
5. Gazetteers built of named entities discovered in previously processed clinical corpora: 19264 words
6. List of the gold standard corrections of misspellings in previously processed texts: 78889 words

Combined, this knowledge base provides us with 758,481 unique words at our disposal. The fact that the test corpus of 20 million tokens which contained 57,523 words and initially had 24,735 unknown tokens, i.e., tokens which did not exist in our knowledge base, only shows the infinite diversity in clinical notes and the magnitude and complexity of the problem.

2. Related Work

Various works have explored the problem of spelling correction in the past but literature on spelling correction in clinical notes, which is a distinct problem, is quite sparse. A previous work (Crowell et al., 2004) used word frequency based sorting to improve the ranking of suggestions generated by programs like GSpell and Aspell. Their method does not actually detect any misspellings nor generate suggestions. Work (Ruch, 2002) has also been done to study contextual spelling correction to improve the effectiveness of an IR system.

Another group (Tolentino et al., 2007) created a prototype spell checker using UMLS and Wordnet as their sources of knowledge. See (Mykowiecka and Marciniak, 2006) for a

program for automatic spelling correction in mammography reports. They utilized edit distances and bigram probabilities and their method is applied to a very specific domain. None of these methods scale up satisfactorily to the size and diversity of our problem.
Due to the lack of previous work on spelling correction in our very large medical corpus (60 million tokens), we were motivated to devise our own method.
The remainder of this paper is organized in the following manner. Section 3 contains details about the suggestion generation, context sensitive ranking algorithm and the heuristics employed by our method. Section 4 gives details about the data sets that we used in our experiments. Section 5 presents the experiments with the results. Sections 6, 7 and 8 include the interpretations, possible improvements in future work and conclusion.

3. Method

Our method for spelling correction uses a combination of rule-based suggestion generation and a context-sensitive ranking algorithm.

3.1. Suggestion Generation

The suggestion generator that we used follows a series of rules to choose the suggestions that would be ranked in subsequent processing. The following are various rules that are applied to the unknown word to choose valid suggestions. It should be noted that if the algorithm finds that a rule matches atleast one valid suggestion, it does not process the subsequent rules. It simply passes on the computed set of suggestions to the ranking algorithm. For example, if the 'Edit Distace One' algorithm finds atleast one suggestion that matches a word in our knowledge base, the subsequent rules are not processed. Otherwise, the next rule would have been processed and so on.
Table 1 provides the average individual contribution of each of the following rules if they are used in isolation from the rest.

1. *Edit Distance 1:* Edit distance (also known as Damerau-Levenshtein distance (Damerau, 1964), (Levenshtein, 1966)) is a "distance" (string metric) between two strings, i.e., finite sequence of symbols, given by counting the minimum number of operations needed to transform one string into the other, where an operation is defined as an insertion, deletion, or substitution of a single character, or a transposition of two characters. The suggestions are generated by simulating **deletion**, **transposition**, **alteration** and **insertion** in one character position of the unknown word and, all those strings that are verified as legal from our knowledge base are put into the bag of suggestions.

   Misspellings like *blokage*: (blockage), *patholigical*: (pathological) and *maliase*: (malaise) are corrected.

   This method is based on Damerau's(Damerau, 1964) findings, according to which a majority of all spelling mistakes are represented by these actions. This list covers common typographic errors such as those caused due to pressing adjacent keys. It also covers those misspellings where the typist forgot to press a key or exchanged adjacent letters.

   The suggestions are sorted according to their frequency in the corpus or the knowledge base, the differences of which are evaluated later.

   It must be noted that these suggestions are not passed to the subsequent context-sensitive ranking algorithm. This is a heuristic that we have employed. The sorting according to the word frequency greatly increased the accuracy of the suggestions generated by this process(see Table 2). In the results we have shown the accuracy achieved by sorting both with corpus frequencies (CFSORT) and knowledge base frequencies (KBSORT) and their combination (CFKBSORT).

2. *Missing white space between 2 words:* In this step, most of the tokens which consist of two valid words concatenated due to missing white-space, are identified. This is achieved by inserting a space between every adjacent letter pair and then checking if the two substrings exist in the knowledge base. The set of suggestions generated by this function is passed on to the ranking algorithm.

   Misspellings like *patientrefused*: (patient refused) are corrected.

3. *Edit Distance 2:* In this step we compute all the strings within edit distance 2 of the misspelling. This is achieved by finding all strings within edit distance 1 of the unknown word and repeating the process for all generated words. This method too has a good coverage and covers misspellings which might have a single error at 2 distinct positions in the word. It also covers those words which have multiple errors at a single site. Of these, all those strings that exist in our database are passed to the ranking algorithm.

   Misspellings like *statarating*: (saturating) are corrected.

4. *Phonetic*: This process generates all the valid strings which are phonetically similar to the misspelling. The set of strings generated is verified against the knowledge base and all those strings which are valid are passed on to the ranking algorithm.

   The algorithm used here is the "Double Metaphone" algorithm.The Double Metaphone search algorithm is a phonetic algorithm and is the second generation of the Metaphone algorithm (Phillips, 2000). It is called "Double" because it can return both a primary and a secondary code for a string; this accounts for some ambiguous cases as well as for multiple variants of surnames with common ancestry. For example, encoding the name "Smith" yields a primary code of SM0 and a secondary code of XMT, while the name "Schmidt" yields a primary code of XMT and a secondary code of SMT–both have XMT in common.

   Misspellings like *simptamotolgi*: (symptomatology) are corrected.

5. *Phonetic edit distance 1*: In our medical lexicon, many words are long and complex. Typists often tend to miss one syllable or do a contraction. With this function, such words will be coded by phonetics and then we generate all words with edit distance one for this phonetic code. The suggestion list is populated by all words which have the same phonetic code as we generated here. It will largely extend our suggestion list.

   Misspellings like *amns*: (amounts) and *ascultion*: (auscultation) are corrected.

6. *Edit distance 1 and concatenated word:* The combination of rules 1 and 2 is used. There are often words in the text where a valid word is joined with an invalid one, which is usually within 1-edit distance of its correction. Such cases are dealt with at this stage. As before, the set of suggestions is passed on to the ranking algorithm.

   Misspellings like *affectinback*: (affecting back) are corrected.

7. *Multiple concatenated words:* Sometimes, 3 or more words have been concatenated due to missing white space. Such words are filtered here and the suggestions are passed on to the ranking algorithm.

   Misspelling like *explorationforbleedingand*: (exploration for bleeding and) is corrected.

3.2. Context-Sensitive Ranking Algorithm

Previous studies (see (Elmi and Evens, 1998), (Golding and Schabes, 1996) and (Kukich, 1992)) have suggested that spelling correction is much more effective when the method takes into account the context in which the word occurs. We take into account the context considering the misspelling, one word before it and one word after it. Unlike some previous studies, we did not experiment with context defined as adjacent letters. Since our corpus is large, we could get reliable estimates if we considered the surrounding words as the context.

For this purpose, we used the CMU-Cambridge Statistical Language Modelling Toolkit (Clarkson and Rosenfeld, 1997). It is a suite of UNIX software tools to facilitate the construction and testing of statistical language models. It includes tools to process general textual data to word frequency list and vocabularies, n-gram counts and backoff models.

The toolkit was used to build a vocabulary using all the words in the corpus which occurred more than once. Using these words, a language model was constructed. This tool was used to calculate all unigram, bigram and trigram probabilities along with their corresponding backoff values. The probability values were smoothed to increase the reliability. One of the salient features of our method is that the corpus was bootstrapped to correct its own mistakes. This was done by ranking the suggestions using the language model and the frequency of the misspellings in the corpus itself. This method helped promote the more relevant suggestions in the suggestion set to the top positions. This method greatly increased the accuracy of the first suggestion.

3.3. Complex Frequency Ranking Algorithms

The basic ranking algorithm for candidate corrections is a frequency-based technique. (Crowell et al., 2004) posed their frequency-based model to improve the spelling suggestion rank in medical queries. Our trained dictionaries have two kinds of frequency values. The first is a knowledge based word frequency which computes the number of times the same word is repeated when we are training the dictionary set. The second is a corpus word frequency which uses the word frequencies in the corpus. As shown in previous works, frequency-based methods can improve the first suggestion accuracy of spelling correction, however, the dictionary configuration is a real problem, and is fairly hard to optimize. At this stage, we train our frequency values based on the following three methods: the first is based on corpus word frequency; the second is based on knowledge base word frequency; the third is based on the combined frequency in the knowledge base and corpus. Considering there are many misspellings from missing white space(s), it is not fair to compute the frequency of the misspelling as a whole or just compute the frequency of the first separated word. This paper introduces two new frequency ranking algorithms:

1. First two words ranking: This algorithm computes the frequency of the first two words when a suggestion contains more than one word.

2. First and last word ranking: This algorithm computes the combined frequency of the first and last word of a suggestion.

3.4. Heuristics Employed

A set of the interesting heuristics that was evaluated to improve the first suggestion's accuracy is also provided.

1. Suggestions generated by Edit Distance 1 were not ranked by the context-sensitive ranking algorithm. They were instead sorted by their word frequencies in the knowledge base. The rationale behind this strategy was that suggestions that occur more frequently in the knowledge base are more likely to occur in the corpus. The accuracy was found to drop sharply when we tried to rank the suggestions by this method.

2. The misspellings of adverbs ending with 'ly' were not being corrected in a satisfactory manner. For this purpose, we devised a simple heuristic. If a misspelling ended with 'ly', then the first suggestion in the suggestion-set ending with 'ly' was arbitrarily promoted to the first position in the suggestion set. The application of this heuristic enabled us to correct almost all such misspelt adverbs.

   For example, if the misspelling was *accidentaly*, the first suggestion might be *accidental* whereas the correct suggestion *accidentally* might be third or fourth in the list.

3. The orthographic case for the misspelling was not considered. All words were converted to lower case. It was noted that although words beginning with upper

case do give an idea if the word is a named entity, many such named entities did not begin with upper case. Therefore, to simplify processing the case was ignored for all words.

4. The Edit Distance methods used Inverse Edit Distance instead of direct edit distance due to the large size of our dictionaries. The use of Direct Edit Distance would have led to unnecessarily long processing times. Instead Inverse Edit Distance allowed us to calculate all the possible strings and then check the dictionaries for them.

5. In some cases, one or more suggestions have the same probability when we try to rank them by language model. This was due to sparseness in corpus. In such cases, we summed frequency of suggestions in the corpus and frequency of the suggestion in knowledge base and included them in computation of the language model. This helped us get more reliable ranking for suggestion.

4. Data Set

1. The **training data** consisted of misspellings in a corpus of clinical records of the Emergency Department at the Concord Hospital, Sydney. The total numbers of unique words in the corpus was 57,523. This corpus contained 7442 misspellings for which we already had gold standard corrections verified by expert analysis. We found that this set served our purpose quite well because it contained a good mixture of medical as well as non-medical words. It also contained many rare words and some very convoluted misspellings which could not be corrected by usual spelling correction techniques.

   Interestingly, due to this reason the methods show a higher accuracy on the test data than the training data.

2. The **test data** contained 2 separate data sets as described below.

   - The first data set consisted of 12,438 words in the Concord Emergency Department Corpus which has been described above. These words were hitherto unknown words which did not exist in our knowledge base. These words were manually corrected and along with their gold standard they serve as one of the test data sets of our method.
   - The second data set consists of 65,429 misspellings from the Intensive Care Unit patient records at the Royal Prince Alfred Hospital, Sydney, the RPAH-ICU corpus. This corpus contained a total of 164,302 unique words.

5. Experiments and Results

5.1. Experiments

Initial experiments started with a simple spell checker to establish a baseline. At this stage the spellchecker only checked for a known word, word with 1-edit distance or 2-edit distance, as described in section 3.1. The first suggestion was matched with the gold standard and the accuracy was calculated. This method gave an accuracy of 60.52% on our test data and 63.78% on the training data. We then incrementally added the different heuristics to further improve the accuracy. The percentage increase with the addition of each step is presented in Table 2. Finally, the accuracy reached 62.31% on the training data and 75.11% on the test data.

At this point sorting of the suggestions based on their frequency in the knowledge base was incorporated. Words which are more frequent in the knowledge base should logically be ranked higher in the suggestion list.

This option was only available because of the large knowledge base. Therefore, the suggestions based on their word frequencies in the corpus was also used. In both cases, the accuracy improved by a significant margin. But the sorting based on corpus frequencies provided slightly higher accuracy than that of knowledge base frequencies.

Looking at an example to illustrate why **sorting based on knowledge base frequencies** is useful. The word *haemodynamically* has been misspelt in 381 different ways in previously processed texts. In the ICU corpus alone, it occurs 30,162 times in its correct form. This makes it highly probable that if ‘haemodynamically’ is present anywhere in the suggestion list, it deserves to be among the top suggestions.

Similarly, **sorting based on corpus word frequencies** is also helpful and is a method which can be be used for any large corpus. Also, the texts belong to a single department and a great deal of medical terms that are used in one department would not have been used at all in another. Therefore this method helps improve the ranking especially for words that are typical of the specific corpus. Phone1 algorithm works better in the Concord-ED collection because it contains more complex errors than ‘Train’ and the RPAH-ICU data set.

Sorting based on combined frequencies in corpus and knowledge base was slightly more effective than these methods individually(See Table 2).

After trying a number of different methods to generate suggestions, the next improvement was generated from experimenting with ranking suggestions. When the number of suggestions is large ranking is difficult because there are a lot of high-frequency small words which are within ‘Edit Distance 1’ or ‘Edit Distance 2’ of many misspellings. So, some dictionaries were progressively removed from our knowledge base. As shown in Table 3, removing each dictionary improves the result slightly. While having rich resources may increase the coverage (recall) however it injects noise into the suggestions and leads to a decrease in accuracy. We need to find a tradeoff between the coverage and the number of suggestions for each misspelling. By reducing the number of dictionaries, number of suggestions is reduced. The last row of this table shows the effect of a language model on the overall accuracy. For this, a language model was created from all the lexically verified words that occurred more than once in the specific medical text that is being processed. The language model stores all the word unigrams, bigrams and trigrams with their corresponding backoff values and probability values. The language model probabilities were added to the corpus and knowledge base frequencies as explained in Heuristic 5. The reason RPAH-

Rule	Percentage	Description
Edit distance 1	69.80	Edit distance one
Two words	6.73	Missing white space
Edit distance 2	85.52	Contains Edit distance one and two
Phonetic	42.52	Pure Phonetic
Two word edit distance 1	7.09	Contains Two words and Two edit distance one
Multi word	6.86	Contains two or more concatenated words
Phonetic edit distance 1	84.78	Pure Phonetic and Phonetic with Edit distance one

Table 1: Mean Individual Contribution of each Spelling Correction Algorithm to the Concord-ED Corpus

Rule	Training data	Concord-ED	RPAH-ICU
(Baseline)Edit Distance 1 and 2	60.62	63.78	62.28
Adding ‘Two words’	62.31	74.99	66.42
Adding ‘Phonetic’	62.31	75.11	66.57
Adding ‘Two word edits 1’	62.31	75.11	66.58
Adding ‘Multi word’	62.31	75.11	66.61
Adding ‘Sorting based on corpus word frequency’	87.19	93.17	81.24
Adding ‘Sorting based on knowledge base word frequency’	83.41	93.07	79.99
Adding ‘Sorting based on combined frequency in knowledge base and corpus’	87.25	**93.54**	81.45
Replace ‘Phonetic’ with ‘Phonetic edit distance 1’	87.24	93.43	81.69
Adding ‘First two words ranking’	87.25	92.89	81.75
Adding ‘First and last word ranking’	**87.26**	93.05	**81.83**

Table 2: Percentage Accuracy After Addition of each Process Described in Section 3.1

ICU data collection was used here is, it contains a complete set of all errors and so is a good test set for this method. The language model was applied to the suggestions computed after application of each of the sorting strategies. The result drops a little bit because the frequency ranking is very good and adding each ranking method may worsen the accuracy.

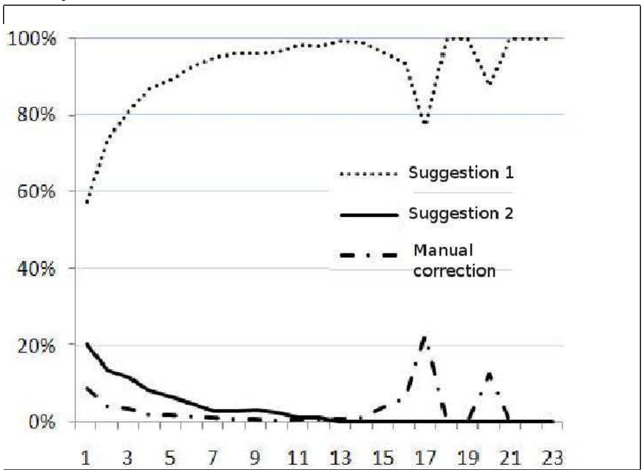

Fig 1. Variation of Accuracy of Suggestions with Word length

5.2. Observations

1. Each of the three data sets that has been used in this experiment is unique in itself. This is the reason that the accuracies are very different for each data set. But analysis of an individual data set's results shows that the increase with the addition of sorting and/or language model was equally pronounced. The training set has some from very complex misspellings which could not be corrected by a general heuristic. The Concord Test Corpus had more non-medical words and fewer medical terms. The RPAH-ICU corpus had more of a mix of medical terms and non-medical words.

2. As shown in Table 2, both ‘First two words ranking’ and ‘First and last word ranking algorithms’ improved the accuracy, but obviously the latter is better. According to the analysis of over 200 multiple concatenated word cases, the small word coverage of ‘First two words ranking’ is 89 against 81 of ‘First and last word ranking algorithms’. The small words are defined as prepositions, conjunctions, pronouns and some common adjectives, adverbs, verbs and nouns, which have fairly short word length and fairly high frequency value. The short length makes them easier to be considered as suggestions, and the high frequency value can adversely affect the suggestion ranking. Thus, FLW ranking is more suitable as it skips the short words and only considers the first and last word.

3. An analysis of the results showed that the accuracy of our method was found to increase as the word length increased. This is because the bag of suggestions for

Rule	Correct Num (out of 65429)	RPAH-ICU corpus
Remove UMLS	54038	82.59
Remove SNOMED-CT	53769	82.18
Remove Both	54857	83.84
Remove Gazetteers	53614	81.94
Remove All	55197	84.36
Reomve All + Adding Language Model	54985	84.04

Table 3: Percentage Accuracy After Removing Dictionaries and Adding Language Model

longer words contains less number of suggestions and we can easily rank them.

Figure 1 shows the gradual increase of the first suggestion's accuracy with increasing word length. It also shows the decrease in accuracy of second suggestion and manual corrections with increasing word length. Spikes at word length 17 and 20 are anomalies due to low frequency of very long tokens in the corpus.

4. The results show that an exhaustive knowledge base is a boon for a spell checker.

5. With a large corpus, both corpus word frequencies and context based probabilities are quite effective. A combination of these two techniques yields improved results.

6. Possible Improvements

The authors believe that the method described in this paper can be improved by one or more of the following:

1. Both language model based sorting and simple word frequency sorting to improve the ranking of suggestions, have given us comparable results. This does not mean that the use of language model is futile. We believe that there is room for improvement in the language model. We have used only trigram probabilities but context is often beyond trigrams. In the future, we plan to implement a more sophisticated language model to augment the spelling correction system.

2. Some words could have been properly corrected with a good US-UK spelling disambiguation system. It was seen that sometimes the first suggestions was *anesthetize* but the corresponding gold standard was *anaesthetize*. In our case, the obvious work-around to this problem was adding both to the gold standard. In some cases, we looked at the Wordnet synonym list of the first suggestion. For non-medical terms, this method often worked fine. But far better than all such methods, would have been the use of a robust US-UK spelling disambiguation system with good coverage.

3. Sometimes when the misspelling was that of a verb, the spelling suggestions preserved the root but not the morphology. A system which incorporates such functionality would be another possible improvement.

4. A small percentage of words in the gold standard are erroneous and this slightly lowered the accuracy.

5. A limitation of the context sensitive ranking algorithm was that the suggestion generated did not always exist in the corpus itself. We tried countering this error by adding the Knowledge Base word frequencies to the trigram probabilities. A semantic approach to such ambiguous cases would be more effective and, at the same time, elegant.

7. Conclusions

We have presented a method for spelling correction based on combining heuristic-based suggestion generation and ranking algorithms based on word frequencies and trigram probabilities. We have achieved high accuracies on both the test data sets with both the methods. We believe the results are satisfactory and an improvement upon previous work. We look to implement more sophisticated methods in the future to utilize the context in a more effective manner. This spelling suggester has been incorporated in the group's workflow and is used in the manual process of spelling correction to great effect.

8. References

P. Asgari and J. Patrick. [2008]. The new frontier - analysing clinical notes for translation research - back to the future. In Violaine Prince and Mathieu Roche, editors, *Information Retrieval in Biomedicine: Natural Language Processing for Knowledge Integration.*, pp 357–378. IGI Global.

P.R. Clarkson and R. Rosenfeld. [1997]. Statistical language modeling using the cmu-cambridge toolkit. In *Proceedings ESCA Eurospeech*, pp 2701–2710, September.

Jonathan Crowell, Qing Zeng, Long Ngo, and Eve-Marie Lacroix. [2004]. A frequency-based technique to improve the spelling suggestion rank in medical queries. *J Am Med Inform Assoc.*, 11:179–185.

F.J. Damerau. [1964]. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 3(7):171–176, March.

Mohammed Ali Elmi and Marthe Evens. [1998]. Spelling correction using context. In *Proceedings of the 17th international conference on Computational linguistics*, volume 98, pp 360–364.

A. Golding and Y. Schabes. [1996]. Combining trigram-based and feature-based methods for context-sensitive spelling correction. In *Proc. 34th ACL*, pp 71–78.

K. Kukich. [1992]. Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24(4):377–349.

V.I. Levenshtein. [1966]. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707.

Agnieszka Mykowiecka and Magorzata Marciniak. [2006]. Domaindriven automatic spelling correction for mammography reports. *Intelligent Information Processing and Web Mining*, 5:521–530.

Lawrence Phillips. [2000]. The double metaphone search algorithm. Technical report, C/C++ Users Journal.

Patrick Ruch. [2002]. Using contextual spelling correction to improve retrieval effectiveness in degraded text collections. In *Proceedings of the 19th International conference on Computational linguistics.*, volume 1, pp 1–7. Association for Computational Linguistics.

A. Ryan, J. Patrick, and R. Herkes. [2008]. Introduction of enhancement technologies into the intensive care service, royal prince alfred hospital, sydney. *Health Information Management Journal*, 37(1):39–44.

Herman D Tolentino, Michael D Matters, Wikke Walop, Barbara Law, Wesley Tong, Fang Liu, Paul Fontelo, Katrin Kohl, and Daniel C Payne. [2007]. A umls-based spell checker for natural language processing in vaccine safety. *BMC Med Inform Decis Mak.*, 7(3), February.

F. Wang and J. Patrick. [2008]. Mapping clinical notes to medical terminologies at point of care. In D. Demner-Fushman, S. Ananiadou, K. Bretonnel Cohen, J. Pestian, J. Tsujii, and B. Webber, editors, *In Proc of BioNLP2008*, pp 102–103. Association for Computational Linguistics (ACL).

Y. Zhang and J. Patrick. [2007]. Extracting semantics in a clinical scenario. In *Health Knowledge and Data Mining Workshop, Ballarat, Research and Practice in Information Technology*, volume 249, pp 217–224. Health Knowledge and Data Mining Workshop, Ballarat, Research and Practice in Information Technology.