RC24912 (W0912-017) December 2, 2009
Computer Science

IBM Research Report

An Empirical Study on Building a State-of-the-art English Spelling Error Correction System

Ming Sun
Johns Hopkins University

Bing Zhao
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

An Empirical Study on Building a State-of-the-art English Spelling Error Correction System

Ming Sun
John Hopkins University
msun8@jhu.edu

Bing Zhao
IBM T.J. Watson Research
zhaob@us.ibm.com

Abstract

In this paper, we present our empirical studies on learning a state-of-the-art English spelling error correction system using Oxford corpora: from detecting spelling errors, building candidate sets, to finally selecting the most likely candidate corrections. With a log-linear model framework, we integrated features based on spelling correction ngrams from supervised data, probabilistic edit-distance, and various distributional similarities. In particular, we did an empirical comparison over these measures, showing the effectiveness for especially the probabilistic edit-distances. We obtained significantly better F-measures, achieving a relative improvement of 49% over Microsoft Word.

1 Introduction

With the boom of text on the web, more and more text in English are coming from nonnative speakers and native speakers typing with errors. Data collected from web, containing many spelling errors, is a big challenge for natural language processes down the stream, as those spelling errors lead to high entropy ngrams. Such errors could lead to significant downgrades of natural language processing systems' quality. A need for spelling error correction is becoming stronger, and stressing more on both *accuracy* and *speed*, for the high volume of noisy web text.

People have been studying English spelling error correction since early 90s. Kukich (1992) summarized methods of detecting English spelling errors, from simple pattern matching to complex methods using keyboard adjacency. Brill and Moore (2000) used an enriched noisy-channel model. Li et al. (2006) studied several distributional similarities for correcting queries in a web search engine. There are many other works, on exploring context-independent word-pair similarities for spelling error corrections such as using pronunciations as in (Boyd, 2009) recently. Most of previous work used *some* data set, which is not easily available for others. In this paper, we used Oxford corpora, which is publicly available[1], on building a *context-dependent* spelling error correction system for both typing mistakes and cognitive errors. Using Oxford corpora, several empirically effective types of feature functions are explored. First is the context-dependent spelling-correction ngram table, the second is the probabilistic edit-distance, and the third is based on distributional similarities. A ngram language model is also integrated to further disambiguate the corrections. A log-linear model, as commonly applied in statistical machine translations, is used to combine all the features in a monotone decoder.

The rest of the paper is structured as follows: in section 2, we briefly introduce spelling error detections; in section 3, we present, in details, on building a spelling error correction models; experiments are reported in section 4, and discussions are in section 5.

2 Spelling Errors Detection

To detect if a word contains potential spelling errors or not, we use a sequential combination of the three methods in our approach. There are: a dictionary lookup, a word-level ngram analysis, and a letter-level ngram analysis. Practically, the combination of the three methods is to narrow down the searching scope, and locate the spelling error efficiently.

Dictionary Lookup

Assuming a *clean* English vocabulary, and one could simply look up the vocabulary for each word. An OOV is usually a potential spelling error.

Word level Ngram Analysis

It is usually hard to define a clean English vocabulary in practice. We employ a ngram language model trained from vast amount of English data for error detection. We calculate every ngram's probability. If a ngram doesn't exist in the statistics, or the ngram score (frequency) is way too small, there is potentially a spelling error inside.

Letter-level Ngram Analysis

Given an English word, we can simply check the letter-level ngrams (n-letters) inside the word. This is because an English word's spelling is highly structured, such as "*consonant-vowel-consonant*" (CVC). A low frequency of a letter-level ngram signals a potential spelling error within the word. Empirically, this model gives more robust estimations than the other two. However, it is commonly ignored in the previous approaches.

[1]URL: http://ota.oucs.ox.ac.uk/scripts/download.php?approval=4a70de5c01855e280939

3 Modeling Spelling Errors Corrections

A query sequence $\vec{q}$ is a sentence with I tokens. Given $\vec{q}$, our goal is to find the spelling errors in $\vec{q}$, and propose the correction – a sentence in the same length – denoted as $\vec{c}$. We pick the best candidate $\vec{c^*}$ which has the lowest cost from a log-linear model combining feature functions.

3.1 Notations

We denote the query sequence $\vec{q}$ as in Eqn. 1, a sentence containing I tokens, and the candidate sequence $\vec{c}$ in Eqn. 2, a sentence containing the same number of tokens[2].

$$\vec{q} = q_1, q_2, \ldots, q_I, \quad (1)$$

$$\vec{c} = c_1, c_2, \ldots, c_I. \quad (2)$$

At the sequence-level, we only consider the one-to-one mapping for the tokens: q_i is aligned to c_i at the i'th position in the query. If q_i is a correct-spelling word, c_i will be identical to q_i. Otherwise, c_i will be the correction for q_i.

A candidate sequences set, $\Omega = \{\vec{c}_k, k = 1 \ldots K\}$, is a set of all alternative spellings for a given query sequence $\vec{q}$. What we want is to we pick up the best candidate $\vec{c^*}$ from Ω, to maximize the posterior probability $Pr(\vec{c} \mid \vec{q})$:

$$\vec{c^*} = \arg\max_{\Omega} P(\vec{c} \mid \vec{q}). \quad (3)$$

Based on Bayesian theorem, it is equivalent to a noisy channel model as below:

$$\vec{c^*} = \arg\max_{\Omega} P(\vec{q} \mid \vec{c}) P(\vec{c}), \quad (4)$$

where $P(\vec{q} \mid \vec{c})$ is a translation model, and $P(\vec{c})$ is a language model (such as a 5-gram). The translation model can be decomposed further into word-level probabilities:$P(\vec{q} \mid \vec{c}) = \prod_{i=1}^{I} P(q_i \mid c_i)$. Given supervised data, in which a spelling error is paired with its correction, we can learn the word-level probabilities for $P(q_i \mid c_i)$. The details are in section 3.3. With a language model to further disambiguate the corrections, we can infer the best candidate $\vec{c^*}$ from the pool Ω, by using a *monotone* phrase-based machine translation decoder.

3.2 Building Candidate Set Ω

The set of candidates Ω is a critical component for building an efficient spelling error correction system. If Ω is too large, the system will be very slow, and if it is too noisy, the system would suffer significantly on its accuracy. To choose a small yet highly-accurate candidate set, we applied the following two steps:

[2]We use i to index a word's position in a sentence, and j to index a letter's position in a word.

1. Unigram frequency: We only consider those words with no smaller unigram frequency than the spelling error: $P_{\text{unigram}}(c) \geq P_{\text{unigram}}(q)$. This means all the candidates should be at least as frequent as the spelling error.

2. Edit-distance limit: Edit distance is used to measure the amount of difference between two sequences. For example, the edit distance between spelling error "feamail" and correction "female" is 3: 2 deletion (a $\rightarrow$ 0, i $\rightarrow$ 0) and 1 insertion (0 $\rightarrow$ e). We consider words within the edit-distance up to $\delta = 2$ to be the candidates: $C = \{c : EditDist(q, c) \leq \delta\}$.

3.3 Features for $P(q_i \mid c_i)$

We explore three types of feature functions, within a log-linear framework for $p(q_i \mid c_i)$. The first one is a table of spelling-correction phrase-pairs learned directly from supervised data. The second is a probabilistic distribution based on edit-distance between candidate and query words. The third one computes distributional similarities using sentence-level context.

3.3.1 Direct Modeling using Supervised Data

We can directly collect the frequencies for pairs (q_i, c_i) from given supervised corpora. For instance, the word 'grated' is corrected as 'rated' for 81 times, and 'grateful' for 110 times. We compute the relative frequencies to have $P(q_i \mid c_i)$ and $P(c_i \mid q_i)$. Besides the unigram-pairs, we can collect ngram-pairs, in which phrasal context are paired with their spelling corrections, to handle cognitive errors effectively.

3.3.2 Edit-Distance Based Probabilities

The second type of feature functions are edit-distance probabilities, which are decomposed into letter-level alignment probabilities in both directions: $P_{\text{edit}}(q \mid c) = \prod_{j=1}^{|q|} p(q^j \mid c^j)$, and $P_{\text{edit}}(c \mid q) = \prod_{j=1}^{|c|} p(c^j \mid q^j)$, where j is the letter index inthe words of c and q.

In order to calculate the letter-based probability, we download the Oxford Corpora from internet. The data includes the spelling errors and their corrections. We compute the confusion matrix for all the letters $a - zA - Z$, and the top part of it is shown in Table 1: where 0 repre-

Table 1: Pasrt of the Confusion Matrix for letters from a–f; errors are at the columns, and corrections are at the rows.

	0	a	b	c	d	e	f
0	0	3750	269	2145	1668	7884	486
a	1832	0	20	157	63	1997	94
b	145	58	0	42	80	28	19
c	942	141	12	0	36	186	50
d	1380	151	71	102	0	133	56

sents the null position, and an entry is a counter for the pair of letters mapped. For the entry $(0, a)$, the count is 3750, meaning that, the letter 'a' is deleted 3750 times. There are several observations on the most frequent errors collected from the Oxford corpora: the most frequent error is deletion (45379 times); the most errorful letter is e (7884 times deleted and 5431 times inserted); the most frequent substitution is: i $\rightarrow$ e (2669 times). This is because letter e is the most frequent letter used in English text. Also, deletions are more frequent than others, because people tend to omit letters during typing.

The confusion matrix is normalized to the row and to the column, to get edit-distance probabilities in both directions. As some letter-pairs in our test data are never seen in the Oxford corpora, we applied an *add-one* smoothing to overcome data sparseness.

3.3.3 Distributional Similarities

Distributional similarities are important measures for spelling error corrections. Let q be a query word, c a candidate word, and v the word co-occurring with both q and c. We compute the following six measures:

1. Geometric Measures: Cosine distances as in Eqn. 5 is selected.

$$sim_{\text{cos}}(q, c) = \frac{\sum_v P(v \mid q)P(v \mid c)}{\sqrt{\sum_v P(v \mid q)^2 \sum_v P(v \mid c)^2}} \quad (5)$$

2. Correlation Measures: the Pearson's Product-Moment Correlation Coefficient in Eqn. 6 is selected.

$$sim_{\text{pm}} = \frac{\sum_v (P - \overline{P})(Q - \overline{Q})}{\sqrt{\sum_v (P - \overline{P})^2 - \sum_v (Q - \overline{Q})^2}}, \quad (6)$$

where $P = P(v \mid q)$ and $Q = P(v \mid c)$.

3. Combinatorial Measures: Jaccard's Coefficient is selected.

$$sim_{\text{jacc}}(q, c) = \frac{\mid V_q \cap V_c \mid}{\mid V_q \cup V_c \mid} \quad (7)$$

and the Dice Coefficient

$$sim_{\text{dice}}(q, c) = \frac{2 \mid V_q \cap V_c \mid}{\mid V_q \mid + \mid V_c \mid}, \quad (8)$$

where V_q is the size of words co-occurring with q and V_c is the size of words co-occurring with c.

4. Substitutability Measures: we use the confusion probability

$$sim_{\text{cp}}(q \mid c) = \sum_v \frac{P(v \mid c)}{P(v)} P(v \mid q)P(q) \quad (9)$$

and the reversed confusion probability

$$sim_{\text{cp}'}(c \mid q) = \sum_v \frac{P(v \mid q)}{P(v)} P(v \mid c)P(c) \quad (10)$$

4 Experiments

Oxford corpora are a collection of many small files, containing misspellings of English words from both native and nonnative speakers. We unified the files' format, and collected the sentences and phrases containing spelling errors with their corrections. We picked the first 10-chapter of '*The Young Visitors*' by Daisy Ashford, a nine-year-old in Victorian England, as the major test set.

4.1 A Monotone Decoder

As in Eqn. 4, we used a monotone phrase-based statistical machine translation engine. We learn three kinds of features for $p(q_i \mid c_i)$: a supervised spelling correction table, edit-distance based probabilities, and distributional similarities. If a potential spelling error was not covered by the supervised spelling-correction table, we dynamically construct a candidate pool § 3.2, and use edit-distance based probabilities and distributional similarities to rank the candidates. A 5-gram language model $p(c_i)$ was used to further disambiguate the correction choices.

4.2 Building Supervised Spelling-Correction Table

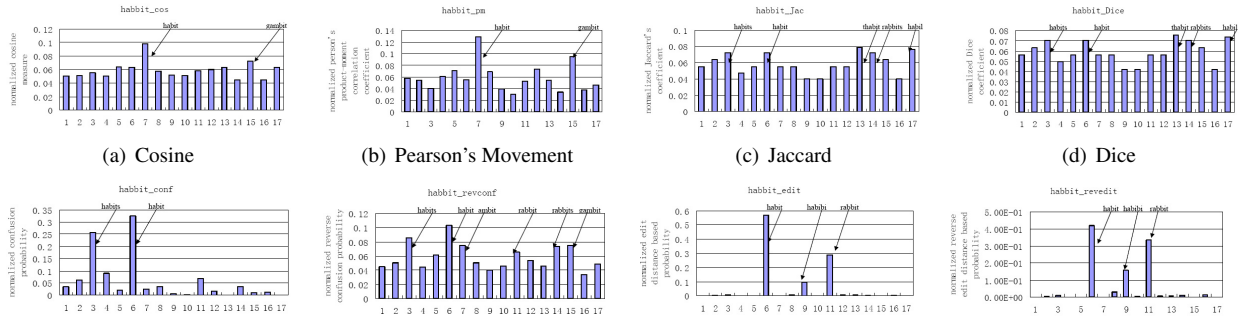
In Oxford corpora, there are many types of spelling errors including cognitive errors and spelling errors with edit distance of larger than 4. We collected $53,206$ context-dependent spelling-correction ngram/phrase pairs, similar to the phrase-table as used in statistical machine translations. Here, our source side is a phrase containing spelling errors, and the target side is its correction. This table can be used to handle cognitive errors. For instance, the source ngram "it is *hire*" is corrected into "it is *here*", while in "*hire* values", it is corrected into "*higher* values".

4.3 Learning Word Co-occurrence Table

w_1 and w_2 co-occur, if they are in the same sentence. This word co-occurrence table is needed to compute all the distributional similarities in § 3.3.3. We collect the statistics from English gigaword corpus, with 4-billion running tokens. We prune the table by removing all punctuation and keeping only the top 10%, with a minimum of 10 and up to top-50 co-occurring words for each word.

4.4 Computing Eight Similarity Measures

Here, we use the query word 'habbit' to illustrate the eight similarity measures selected in our final system. The candidates chosen for 'habbit' are listed in Table 2, with the corresponding measures. The measures with more candidates are shown in Figure 1. First, we observed the probabilistic edits in § 3.3.2 are much sharper than others; secondly, the confusion probability is also strong in choosing the correct candidate 'habit', which was also positively reported in Li et al. (2006), but the reverse direction (setup "revconf") is not as good. The rest measures shares similar strength in rankings.



(a) Cosine (b) Pearson's Movement (c) Jaccard (d) Dice

(e) Confusion Prob (f) Reverse Confusion Prob (g) Edits (h) Reverse Edits

Figure 1: Comparing measures for spelling error correction.

Table 2: Correction Candidates and Similarity Measures

	Cosine	PM	Jacc	Dice	Conf	Conf'(reverse)	Edit	Edit'(reverse)
habit	0.063	0.055	0.072	0.070	0.325	0.104	0.568	0.420
rabbit	0.058	0.053	0.056	0.056	0.068	0.065	0.286	0.336
harbin	0.051	0.061	0.048	0.049	0.090	0.044	1e-4	2e-4

Table 3: F-Measures for Oxford Unseen Data

	MS Word	Supervised Correction Table	Prob. Edit-distances	All Feat(edit dist:1)	All Feat(edit dist:2)
Precision	0.296	0.863	0.758	0.721	0.379
Recall	0.922	0.302	0.373	0.620	0.423
F-Measure	0.447	0.448	0.564	0.667	0.473

4.5 Results on Spelling Error Corrections

We first applied our system on the 'Ashford' test set, containing 6,353 tokens with 208 corrections. The edit-distance limit for candidates was chosen to be up to 2 in our experiments. The results are shown in Table 3:

Microsoft Word is applied to select the top-1 suggested candidate those words labeled with errors, and we got a F-measure of 44.7%. The supervised correction table § 4.2 alone achieved the same performance at 44.8%, and two-direction probabilistic edit-distances alone as described in § 3.3.2 give 56.4%. Combining all features, with a edit-distance limit of 1 for building candidate pool, we achieved a F-measure of 66.7%, a 49% relative improvement over MS-word. Arguably, most of spelling errors are within edit-distance of one. If we relax the edit-limit to 2, the F-measure drops significantly, as too much noises will be introduced in the candidates, which can also slow down the decoding significantly.

On a second test set from web genre, we have 9,962 tokens, and 30 corrections. We got a F-measure of 0.586 (precision 0.786, recall 0.467), while MS Word only achieved a F-measure of 0.31, missed 4 cognitive errors, and a few named entities. Our supervised spelling-correction table from Oxford corpora handles all the 4 cognitive errors correctly. MS-Word, however, has higher recall than our system on most cases.

As we are using monotone decoding, the speed of our system is on average 203 tokens per second.

5 Conclusions and Discussions

Our empirical results highlighted a few components with significant practical values for building a high-accuracy English spelling error correction system. They are: a letter-ngram analysis for spelling error detection, a spelling-correction table directly extracted from Oxford corpora to handle cognitive errors, and eight measures using edit-distance based probabilities and distributional similarities. The edit-distance based probabilities are shown to be much sharper than other measures. Our system produces significantly better F-measures than MS-Word on two difficult test sets, using simply the statistics from the Oxford corpora.

References

Adriane Boyd. 2009. Pronunciation modeling in spelling correction for writers of English as a Foreign Language. In *Proceedings of HLT*, pages 31–36, Boulder, Colorado, June.

Eric Brill and Robert C. Moore. 2000. An improved error model for noisy channel spelling correction. In *Proceedings of ACL*, pages 286–293, Morristown, NJ, USA.

K. Kukich. 1992. Techniques for automatically correcting words in text. In *ACM Computing Surveys.*, volume 24.

Mu Li, Muhua Zhu, Yang Zhang, and Ming Zhou. 2006. Exploring distributional similarity based models for query spelling correction. In *Proceedings of Coling-ACL*, pages 1025–1032, Sydney, Australia, July.