An Unsupervised Approach to Preposition Error Correction

Aminul ISLAM
Department of Computer Science, SITE
University of Ottawa
Ottawa, ON, Canada
mdislam@site.uottawa.ca

Diana INKPEN
Department of Computer Science, SITE
University of Ottawa
Ottawa, ON, Canada
diana@site.uottawa.ca

Abstract:

In this work, an unsupervised statistical method for automatic correction of preposition errors using the Google n-gram data set is presented and compared to the state-of-the-art. We use the Google n-gram data set in a back-off fashion that increases the performance of the method. The method works automatically, does not require any human-annotated knowledge resources (e.g., ontologies) and can be applied to English language texts, including non-native (L2) ones in which preposition errors are known to be numerous. The method can be applied to other languages for which Google n-grams are available.

Keywords:

Preposition errors; Google web 1T; n-grams

1. Introduction

Prepositions are known to be one of the most frequent sources of errors for L2 English speakers. Bitchener et al. [1] found that preposition errors accounted for 29% of all the errors made by intermediate to advanced English as a Second Language (ESL) students. As a result, it seems desirable to focus on this problematic part of speech, in developing a system for automatic error correction in English writing.

Prepositions are challenging for learners because they can appear to have an idiosyncratic behavior which does not follow any predictable pattern even across nearly identical contexts. That is, the choice of a preposition for a given context also depends upon the intention of the writer. For example, "they sat near the beach", "at the beach", "on the beach", "by the beach" are all grammatically correct. Prepositions are so difficult to master because they perform so many complex roles. In English, prepositions appear in adjuncts, they mark the arguments of predicates, and they combine with other parts of speech to express new meanings.

The preposition error correction method that we propose here uses the Google Web 1T n-gram data set [2] that contains English word n-grams (from unigrams to 5-grams) and their observed frequency counts. Our preposition correction method can be applied to other languages for which Google n-grams are available, namely the ten European languages for which n-grams were recently released[1].

[1]http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2009T25

978-1-4244-6899-7/10/$26.00

This paper is organized as follow: Section 2 presents a brief overview of the related work. Our proposed method is described in Section 3. Evaluation and experimental results are discussed in Section 4. We conclude in Section 5.

2. Related Work

To the best of our knowledge, most of the methods for correcting preposition error are based on supervised approaches. An unsupervised method for correcting preposition errors for French as a second language is presented in [3] and it uses counts collected from the Web in a simple way, in order to rank the candidates. Eeg-Olofsson [4] used 31 handcrafted matching rules to detect extraneous, omitted, and incorrect prepositions in Swedish text written by native speakers of English, Arabic, and Japanese. In a test of the system, 11 of 40 preposition errors were correctly detected.

Izumi et al. [5] train a maximum entropy classifier to recognize various errors using contextual features. Their results for different error types are: for omission - precision 75.7%, recall 45.67%; for replacement - precision 31.17%, recall 8%; but there is no break-down of results by individual parts of speech. Therefore we do not know what were the results for prepositions only. Chodorow et al. [6] present an approach to preposition error detection which also uses a model based on a maximum entropy classifier trained on a set of contextual features, together with a rule-based filter. They report 80% precision and 30% recall. Gamon et al. [7] use a complex system including a decision tree and a language model for preposition errors.

Bergsma et al. [8] present a unified view of web-scale approaches to lexical disambiguation where they use the counts of n-grams ($n \in \{5, 4, 3, 2\}$) in a supervised method on the task of preposition selection. They also use an unsupervised version of their supervised method where they produce a score for each candidate by summing the (unweighted) log-counts of all context patterns (i.e., n-grams) using that candidate, and the candidate with the highest score is taken as the solution. While they use the counts of all the n-grams ($n \in \{5, 4, 3, 2\}$), we use the counts of any subsequent (n-1)-grams only if we do not get any suggestion using the counts of n-grams. As a result, our method is computationally more efficient. Their supervised method obtained 75.4% accuracy and unsupervised method obtained 73.7% accuracy. We do not directly compare our results to their results because of the unavailability of their test data set.

Felice and Pulman [9] propose a classifier-based supervised approach to correct preposition error in native (L1) English

and L2 English that uses a corpus of grammatically correct English to train a maximum entropy classifier on examples of correct usage. The L1 source they use is the British National Corpus (BNC). Their feature vectors include 13 feature categories for prepositions. We will compare our results to their results, on the same size of test data.

3. Proposed Method

Our task is to find the best preposition from a set of candidates that could fill in the gap in an input text, using the Google n-gram data set. The gap is where a preposition was in the original text. In this case, we know what is the expected solution, the original preposition in this L1 text. First, we use the Google 5-gram data set to find the best choice from a set of candidate prepositions. If the 5-gram data set fails to generate a choice, then we move to the 4-gram data set, the 3-gram data set, or the 2-gram data set, if the preceding data set fails to generate at least one choice. Let us consider an input text W which after tokenization has p ($2 \leq p \leq 9$) words[2], i.e., $W = \{\ldots\ w_{i-4}\ w_{i-3}\ w_{i-2}\ w_{i-1}\ \boxed{w_i}\ w_{i+1}\ w_{i+2}\ w_{i+3}\ w_{i+4}\ \ldots\}$, where $\boxed{w_i}$ (in position i) indicates the gap and w_i in $\boxed{w_i}$ denotes a set of m prepositions (i.e., $w_i = \{s_1, s_2, \cdots, s_j, \cdots, s_m\}$). Our task is to choose the $s_j \in w_i$ that best matches with the context. In other words, the position of i is the gap that needs to be filled with the best suited member from the set, w_i. First, we discuss how we categorize different n-grams based on the gap position and how we find the *normalized frequency value*. Then, we discuss the procedure to find the best choice preposition using the n-gram data set.

3.1 Categorizing n-gram Type Based on the Gap Position

We categorize an n-gram type (we call it, k) based on the position of the gap in the n-gram. When we consider the 5-gram data set, there might be five types of 5-grams (i.e., $k \in \{1, 2, 3, 4, 5\}$) based on the position of the gap in the 5-gram. For example, if the gap position is the last position in the 5-gram then we call it type 1 (i.e., $k = 1$). Thus, a 5-gram, $w_{i-4}\ w_{i-3}\ w_{i-2}\ w_{i-1}\ \boxed{s_j}$, could be represented incorporating k as $w_{(i-4)k}\ w_{(i-3)k}\ w_{(i-2)k}\ w_{(i-1)k}\ \boxed{s_{jk}}$. All the five types of 5-grams are as follows:

$$w_{(i-4)k}\ w_{(i-3)k}\ w_{(i-2)k}\ w_{(i-1)k}\ \boxed{s_{jk}} \quad (k = 1)$$
$$w_{(i-3)k}\ w_{(i-2)k}\ w_{(i-1)k}\ \boxed{s_{jk}}\ w_{(i+1)k} \quad (k = 2)$$
$$w_{(i-2)k}\ w_{(i-1)k}\ \boxed{s_{jk}}\ w_{(i+1)k}\ w_{(i+2)k} \quad (k = 3)$$
$$w_{(i-1)k}\ \boxed{s_{jk}}\ w_{(i+1)k}\ w_{(i+2)k}\ w_{(i+3)k} \quad (k = 4)$$
$$\boxed{s_{jk}}\ w_{(i+1)k}\ w_{(i+2)k}\ w_{(i+3)k}\ w_{(i+4)k} \quad (k = 5)$$

Similarly, all the four types of 4-grams are:

$$w_{(i-3)k}\ w_{(i-2)k}\ w_{(i-1)k}\ \boxed{s_{jk}} \quad (k = 1)$$
$$w_{(i-2)k}\ w_{(i-1)k}\ \boxed{s_{jk}}\ w_{(i+1)k} \quad (k = 2)$$
$$w_{(i-1)k}\ \boxed{s_{jk}}\ w_{(i+1)k}\ w_{(i+2)k} \quad (k = 3)$$
$$\boxed{s_{jk}}\ w_{(i+1)k}\ w_{(i+2)k}\ w_{(i+3)k} \quad (k = 4)$$

[2]If the input text has more than 9 words then we keep at most four words before the gap and four words after the gap to make the length of the text 9. We choose these numbers so that we could maximize the number of n-grams to use, given that we have up to 5-grams in the n-gram data set.

The three types of 3-grams are as follows:

$$w_{(i-2)k}\ w_{(i-1)k}\ \boxed{s_{jk}} \quad (k = 1)$$
$$w_{(i-1)k}\ \boxed{s_{jk}}\ w_{(i+1)k} \quad (k = 2)$$
$$\boxed{s_{jk}}\ w_{(i+1)k}\ w_{(i+2)k} \quad (k = 3)$$

The two types of 2-grams are:

$$w_{(i-1)k}\ \boxed{s_{jk}} \quad (k = 1)$$
$$\boxed{s_{jk}}\ w_{(i+1)k} \quad (k = 2)$$

3.2 Normalized Frequency Value

We determine the *normalized frequency value* of each candidate preposition for the gap position with respect to all other candidates. If we have m candidate choices for the gap position, i, which are $\{s_1, s_2, \cdots, s_j, \cdots, s_m\}$, and their frequencies $\{f_1, f_2, \cdots, f_j, \cdots, f_m\}$, where f_j is the frequency of a n-gram (where $n \in \{5, 4, 3, 2\}$ and any candidate preposition s_j is a member of the n-gram), then we determine the normalized frequency value of any candidate preposition s_j as the frequency of the n-gram containing s_j, over the maximum frequency among all the candidate prepositions for that position.

$$F(s_j) = \frac{f_j}{\max(f_1, f_2, \cdots, f_j, \cdots, f_m)} \tag{1}$$

Now, based on the types of n-gram, k, equation 1 can be written as:

$$F(s_{jk}) = \frac{f_{jk}}{\max(f_{1k}, f_{2k}, \cdots, f_{jk}, \cdots, f_{mk})} \tag{2}$$

3.3 Determining the Best Choice Preposition (Phase 1)

In Phase 1, we first use the Google 5-gram data set to find the best choice preposition. If the 5-gram data set fails to generate a choice then we back off to the 4-gram data set, the 3-gram data set, or the 2-gram data set, if needed. We apply Phase 2 only if the n-gram (where $n \in \{5, 4, 3, 2\}$ data set in Phase 1 fails to generate at least one choice.

3.3.1 Determining the Best Choice Preposition using the 5-gram Data Set

First, we determine f_{11}, which is the frequency of a n-gram with a specific type $k = 1$, where the last word of the n-gram is the first preposition choice among the m candidates. Similarly, we determine f_{j1}, for all $j \in \{2 \cdots m\}$. Now, we determine $F(s_{jk})$ using equation 2, where $k \in \{1 \cdots n\}$. Now, the index of the preposition is:

$$j = \begin{cases} j & \text{if } |\underset{j \in \{1 \cdots m\}}{\text{argmax}} \sum_{k=1}^{n} F(s_{jk})| = 1 \\ 0 & \text{if } |\underset{j \in \{1 \cdots m\}}{\text{argmax}} \sum_{k=1}^{n} F(s_{jk})| > 1 \end{cases} \tag{3}$$

For simplicity, we assume that $\underset{j \in \{1 \cdots m\}}{\text{argmax}} \sum_{k=1}^{n} F(s_{jk})$ returns the set of values[3] of $j \in \{1 \cdots m\}$ for which $\sum_{k=1}^{n} F(s_{jk})$ for all $j \in \{1 \cdots m\}$ attains its maximum value. If the expression $\sum_{k=1}^{n} F(s_{jk})$ returns 0 for all $j \in \{1 \cdots m\}$, then $\underset{j \in \{1 \cdots m\}}{\text{argmax}} \sum_{k=1}^{n} F(s_{jk})$ will return the set $\{1 \cdots m\}$. Again, if $\sum_{k=1}^{n} F(s_{jk})$ has unique value for more than one candidates, then $\underset{j \in \{1 \cdots m\}}{\text{argmax}} \sum_{k=1}^{n} F(s_{jk})$ will return a set containing the indices of those candidates. Thus, in general, if

[3]Even if it returns a single value, we assume it returns a set containing a single value, though the standard argmax returns a value not a set for single value.

$\underset{j\in\{1\cdots m\}}{\text{argmax}} \sum_{k=1}^{n} F(s_{jk})$ returns a single index for n=5 means that the preposition choice is the word $s_j \in w_i$, we return s_j and exit. If $\underset{j\in\{1\cdots m\}}{\text{argmax}} \sum_{k=1}^{n} F(s_{jk})$ returns a set of indices containing at least two indices for n=5 means, we try equation 3 with all possible decreasing n until we get $j \neq 0$, and then we return s_j and exit. Otherwise, we go to Phase 2.

3.4 Determining the Best Choice Preposition (Phase 2)

The question of why we use Phase 2 is best understood by the example "$\cdots$ and Parchment on Bridgefoot □ Stratford-upon-Avon, where the barn $\cdots$" where the gap, □, needs to be filled up by any of {*of*, *to*, *in*, *for*, *on*, *with*, *at*, *by*, *from*}. But, there is no such 5-gram (e.g., *and Parchment on Bridgefoot* □, *Parchment on Bridgefoot* □ *Stratford-upon-Avon* and so on), 4-gram (e.g., *Parchment on Bridgefoot* □, *on Bridgefoot* □ *Stratford-upon-Avon* and so on), 3-gram (e.g., *on Bridgefoot* □, *Bridgefoot* □ *Stratford-upon-Avon* and so on), 2-gram (e.g., *Bridgefoot* □). The reason of the unavailability of such n-grams is that "Bridgefoot" is not a very common word in the Google Web 1T data set.

To solve this issue is straightforward. We follow Phase 1 with some small changes: instead of trying to find all the n-grams ($n \in \{5, 4, 3, 2\}$) where only $s_{jk} \in w_i$ is changed while keeping all of $\{\cdots, w_{(i-2)k}, w_{(i-1)k}, \cdots\}$ unchanged, we try to find all the n-grams ($n \in \{5, 4, 3, 2\}$) where $s_{jk} \in w_i$, as well as any but the first member of $\{\cdots, w_{(i-2)k}, w_{(i-1)k}, \cdots\}$ are changed while keeping the rest of $\{\cdots, w_{(i-2)k}, w_{(i-1)k}, \cdots\}$ unchanged.

4. Evaluation and Experimental Results

We restrict our candidate preposition set to the nine most frequent prepositions in the British National Corpus (BNC): ***of***, ***to***, ***in***, ***for***, ***on***, ***with***, ***at***, ***by***, and ***from***, same as [9] to ensure the conformity, for direct comparison. Felice and Pulman [9] tested their model on a section of the BNC with test set size 536,193. Felice and Pulman [9] mentioned that their model's performance compared favorably to the best results in the literature, although direct comparisons were hard to draw since different groups trained and tested on different preposition sets and on different types of data. To directly compare with Felice and Pulman [9], we also use the same test set size (536,193 cases) from the BNC. Our best result to date is 75.64% accuracy. Figure 1 relates our results

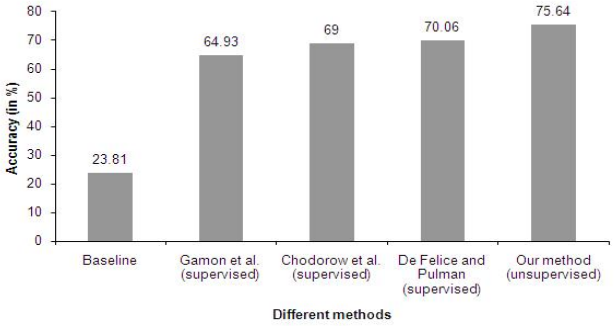

Figure 1: Performance of different methods on L1 prepositions.

to others reported in the literature on comparable task. The baseline refers to always choosing the most frequent option, namely ***of***. Gamon et al. [7] report more than one figure in their results. The figure reported here refers to the task that is most similar to the one we are evaluating. Chodorow et al. [6] also discuss some modifications to their model which can increase accuracy; the result noted here is the one more directly comparable to our own approach.

4.1 Further discussion and analysis

To assess the method's performance on the L1 data, it is important to consider factors such as performance on individual prepositions, the relationship between test set size and accuracy (shown in Table 1), and the kinds of errors made by the model (shown in Table 2). Table 3 shows some

Table 1: L1 results - individual prepositions.

Prepositions	Test set size	Accuracy
of	135,161	94.45%
to	111,834	86.12%
in	97,558	75.73%
for	42,428	56.81%
with	34,953	57.37%
on	32,628	58.54%
by	31,278	54.44%
at	25,652	63.45%
from	24,701	45.24%

Table 2: Confusion matrix for L1 data - prepositions.

Target Prep	Confused with (in %)								
	of	to	in	for	with	on	by	at	from
of		17.00	43.67	14.67	7.33	6.33	5.00	3.67	2.33
to	35.10		30.92	11.92	7.09	4.19	5.15	2.09	3.54
in	51.32	14.04		12.25	5.49	5.81	5.60	2.85	2.64
for	32.88	22.10	25.78		7.23	2.73	4.09	2.86	2.32
with	32.55	17.79	31.71	8.56		3.69	2.52	1.01	2.18
on	30.87	15.71	29.02	5.91	6.84		4.25	3.33	4.07
by	33.86	13.86	28.60	8.25	7.54	3.16		2.63	2.11
at	18.67	18.40	32.00	11.47	6.40	8.00	2.40		2.67
from	29.39	14.97	27.73	8.50	6.28	3.51	5.36	4.25	

Table 3: Examples of method's errors on preposition L1 task

Method's choice	**Correct phrase**
The connections and friendships **with** Surrealism can also be	friendships **of** Surrealism
He wants to escape **from** the world	escape **to** the world
hand were essential ingredients **to** the success of The	ingredients **in** the success
may not be enough **to** a theoretician.	enough **for** a theoretician
saw the twentieth century **in** their eyes but they	century **with** their eyes
figure begins to work **with** us further, now less	work **on** us
as a sympathetic appraisal **of** a critic who is	appraisal **by** a critic
Indeed, an article **of** this length will frequently	an article **at** this length
they seem to proceed **on** his own mind entirely,	proceed **from** his own

examples of instances where the method's chosen preposition differs from that found in the original text. In most cases, the method's suggestion is also grammatically correct, but the overall meaning of the phrases changes somewhat.

Figure 2 shows the number of cases where either a choice (correct or incorrect) or *no suggestion* is generated for different combinations of n-grams[4].

[4]Apostrophe ($'$) is used to denote the n-grams used in Phase 2. x-y-$\cdots$ z-gram means that we use x-grams, y-grams, $\cdots$ and z-grams.

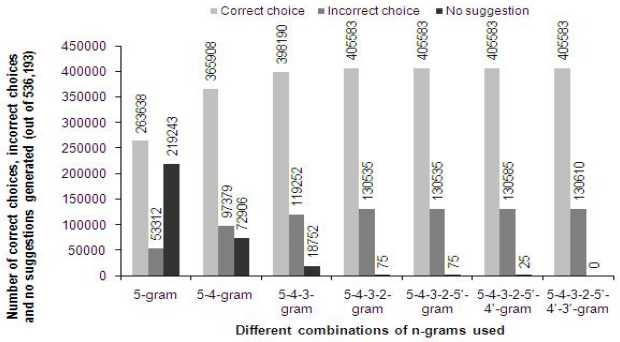

Figure 2: Number of correct choices, incorrect choices and *no suggestion* returned for different combinations of n-grams used.

The performance among different combinations of n-grams is measured using *Precision* and *Recall*. The fraction of suggestions that are correct is the correction *precision* and the fraction of cases corrected is the correction *recall*. Figure 3 shows *precision* and *recall* for different combinations of n-

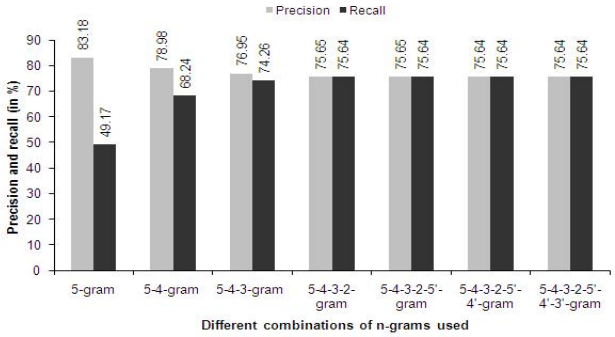

Figure 3: Precision and recall for different combinations of n-grams used.

grams used. We get the highest *precision* (83.18%) when using only 5-grams, which is obvious because 5-grams use the maximum possible context (4 words) and as a result the chance of getting the highest ratio between the number of correct suggestions returned and the number of suggestions returned increases. But the *recall* at this level is very poor (only 49.17%). Figure 3 demonstrates how *recall* gets better using different combinations of n-grams while keeping *precision* as high as possible. Using a combination of 5-4-3-2-5′-4′-3′-grams, we achieve equal precision and recall (which is also the accuracy of the method). Thus, the equal precision, recall and accuracy ensure that for each preposition, we get one and only one suggestion. This is not the case for other approaches.

Using a combination of 5-4-3-2-grams, we get a significant improvement of *recall*, but after that (i.e., a combination of 5-4-3-2-grams to a combination of 5-4-3-2-5′-4′-3′-grams), we do not get any improvement. Figure 4 shows that we need to process[5] only 175 more cases when we move from a combination of 5-4-3-2-grams to a combination of 5-4-3-2-5′-4′-3′-grams. Though for this data set we do not get any new correct suggestions, there is always some chance to provide some correct suggestions. Thus, it is worth taking

[5]When we use only 5-grams, we process all 536193 cases (*no suggestion* for 219243 cases) and then when we use 4-grams along with 5-grams for 5-4-grams combination, we process these unsolved 219243 cases again, thus totalling the number of cases processed to 755436 for 5-4-grams combination.

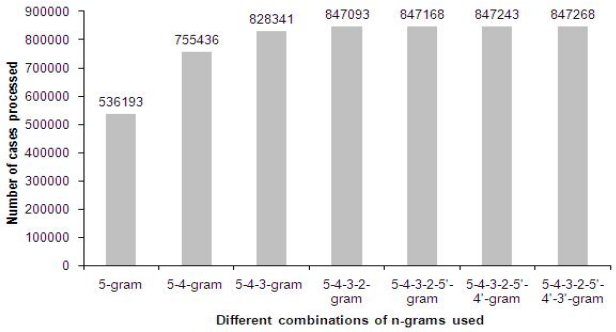

Figure 4: Number of cases processed for different combinations of n-grams used.

into account these later combinations.

5. Conclusion and Future Work

We presented an unsupervised statistical method of correcting preposition errors. We compared this method with three previous supervised methods and show that the performance is comparable or even better. For proprietary reason, we cannot test our method to the L2 data set that Felice and Pulman [9] and Chodorow et al. [6] use. Because the Google n-gram data set is a representation of both native and non-native English, we can say that our proposed unsupervised method is also equally applicable to L2 English texts. In future, we plan to test our method on a L2 data set.

References

[1] J. Bitchener, S. Young, and D. Cameron, "The effect of different types of corrective feedback on ESL student writing," *Journal of Second Language Writing*, vol. 14, pp. 191–205, 2005.

[2] T. Brants and A. Franz, "Web 1T 5-gram corpus version 1.1," tech. rep., Google Research, 2006.

[3] A. D. Matthieu Hermet and S. Szpakowicz, "Using the web as a linguistic resource to automatically correct lexico-syntactic errors," in *LREC'08*, (Marrakech, Morocco), May 2008.

[4] J. Eeg-olofsson and O. Knutsson, "Automatic grammar checking for second language learners - the use of prepositions," in *NoDaLiDa*, (Reykjavik, Iceland), 2003.

[5] E. Izumia, K. Uchimotoa, and H. Isaharaa, "SST speech corpus of Japanese learners' English and automatic detection of learners' errors," *ICAME Journal*, vol. 28, pp. 31–48, 2004.

[6] M. Chodorow, J. R. Tetreault, and N.-R. Han, "Detection of grammatical errors involving prepositions," in *SigSem'07*, (Morristown, NJ, USA), pp. 25–30, 2007.

[7] M. Gamon, J. Gao, C. Brockett, A. Klementiev, W. B. Dolan, D. Belenko, and L. Vanderwende, "Using contextual speller techniques and language modeling for ESL error correction," in *IJCNLP'08*, pp. 449–456, 2008.

[8] S. Bergsma, D. Lin, and R. Goebel, "Web-scale n-gram models for lexical disambiguation," in *IJCAI'09*, (Pasadena, California), pp. 1507–1512, July 2009.

[9] R. D. Felice and S. G. Pulman, "A classifier-based approach to preposition and determiner error correction in L2 English," in *Coling'08*, (Manchester, UK), pp. 169–176, August 2008.