

Automated Detection of Usage Errors in non-native English Writing using One-Class Support Vector Machines

SATORU FUJISHIMA[†] and SHUN ISHIZAKI[†]

[†]Graduate School of Media and Governance, Keio University 5322 Endo, Fujisawa, Kanagawa, 252-0882 Japan

E-mail: † {satoru, ishizaki} @sfc.keio.ac.jp

Abstract

In an investigation of the use of a novelty detection algorithm for identifying inappropriate word combinations in a raw English corpus, we employ an unsupervised detection algorithm based on the one-class support vector machines (OC-SVMs) and extract sentences containing word sequences whose frequency of appearance is significantly low in native English writing. Combined with n-gram language models and document categorization techniques, the OC-SVM classifier assigns given sentences into two different groups; the sentences containing errors and those without errors. Accuracies are 79.30 % with bigram model, 86.63 % with trigram model, and 34.34 % with four-gram model.

1. Introduction

Researchers in automated grammatical error detection and correction have long been facing problems due to a lack of available corpus data. Whilst many other fields of natural language processing (NLP) research have experienced a breakthrough by applying corpus-based quantitative methods during the last two decades, they could hardly utilize the latest advancements in statistical NLP for their studies.[4] Text corpora which contain grammatical and/or semantic errors are relatively easy to be found, however, annotation tasks of those error types require semantic understanding of expressions and thus are fairly difficult for computer programs and non-native speakers to carry out.

Even today, corpus creation and error annotation tasks involve, to a certain extent, a human judge and manual labor and are seen as a major constraint in the research of automatic identification and correction of errors in non-native writing. To reduce manual handling costs of error annotation and learner corpus creation tasks, we employ a detection algorithm based on one-class support vector machines (OC-SVMs) and extract sentences containing unusual word sequences from untagged text data. The idea behind this method is that word sequences whose frequency of appearance is significantly low in native English writing are likely to be inappropriate or erroneous. We regard those

word sequences that appear in non-native English writing as outliers and attempt to find them in accordance with their frequency of appearance in a large untagged corpus of American English.

Errors in English usage are not necessarily grammatically incorrect and usually take place in a sequence of two or more words. These attributes make them remain out of the scope of most existing style and grammar checkers which apply string matching algorithms and rule-based models. Yet these usage errors are no less important than the grammatical and spelling errors that are being actively researched. More often than not, the learners have to face the situation of trying to come up with sentences using words that they are unsure about, in terms of meaning and suitability for the context. There, usage errors can equally cause miscommunication. Linguists report that even advanced learners of English are not free from errors associated with word choices. [6] [8] [11] In this paper, we focus our attention on error detection in a single sentence. The combination of words and the frequency of appearance of those word sequences are used in the detection procedure. Paragraphs are broken into a single sentence and sentences after being singled out are processed individually.

2. Related Work

Hermet et al. (2008) use web search hit counts for preposition error detection and correction tasks. Given a sentence with a selected preposition, they create a pruned and generalized phrase using the target word and then generate a minimal list of alternative prepositions that are easily confused with the target expression. Both of these expressions are evaluated by the number of websites found on the World Wide Web when they are used as search queries. If the number of appearances of the alternative expression(s) exceed(s) that of the original one, the input expression is replaced with the alternative expression whose frequency of appearance is the highest on the web. By testing on 133 French sentences, they achieve 69.9 % in accuracy. They also report that the accuracy rate drops to 30.8%, when the size of French websites is reduced to 1/1000th. [7]

Tetreault and Chodorow (2009) use geographic “region-specific” web search counts in search engine APIs to

detect typical (prepositional) errors in the writing of non-native speakers of English. [13]

Apart from the web, Izumi et al. (2003) classify grammatical and lexical errors found in their Japanese learner corpus of spoken English into “omission-type errors” and “replacement-type errors” and propose to use different error detection methods for the different error types. Attempting to improve the accuracy of grammatical and lexical error detection with a limited amount of training data, they conduct an experiment using their 45 error tag sets and the Maximum Entropy model. [10]

Brockett et al. (2006) applied a noisy channel model of statistical machine translation methods, to capture errors of ESL learners, which are not covered by popular proofing tools designed for native English speakers. Concerning the impact of a first language or mother tongue increases the detection and correction accuracy. [2]

Oyama and Matsumoto (2008) combine n-gram features and supervised document categorization techniques based on the (hard margin) SVM to find learner errors in written Japanese. They prepare 10978 Japanese sentences, of which 5489 sentences contain grammatical mistakes and the other 5489 do not, and make bigram, trigram, four-gram and 5-gram out of them. They omit the word sequences that appear in the both groups before training and build a classification model using the n-grams that appear only in one of the either groups. They achieve the accuracy of 61.4% with trigram, 70.1% with four-gram and 82.6% with 5-gram. [12]

Alam et al. (2006) calculate the probability of co-occurrence of targeted PoS tags using their frequencies in their training corpus. The overall performance is 63% in English (detected 545 out of 866 sentences) and in Bangla 53.7% (203 out of 378 sentences) They mention that the accuracy of grammar checker depends partly on the precision of grammar checkers. [1]

Lee (2009) selects entropy of a trigram language model, parse score, parse tree deviations, head word of a base noun phrase and its determiner, and word dependency types for classification in a subtask of his Ph.D. thesis on Automatic Collection of Grammatical Errors. Employing the ranking mode of the SVM, the accuracy reaches 76.2% with all five features. [5]

Reviewing the previous related researches imply that accuracies of sentence level grammatical error detection are, at best, approximately 80% or around.

3. Experimental Setup

3.1 Test Data

The dataset used in this paper is a raw corpus, created exclusively for the purpose with the support of donators and

volunteers. The sentences in the dataset were originally written as a part of an email, online diary or an assignment of English class by Chinese, German, Japanese and Latin-American learners of English during the year 2006 to 2011. They are composed of 20020 English sentences with at least one error associated with word choice and the manual correction of them that count for 25059. We call the former ones the original sentences and the latter the corrected sentences. The statistical information of the error corpus is provided below.

Table 1: Statistical information of the Error Corpus: the numbers of sentences, the total number of words and the average number and variance of the number of words per sentence.

	Sentences	Words	Average	Variance
Original sentences	20020	257630	12.87	6.80
Corrected sentences	25059	376919	15.04	10.80

In order to conduct a comparative experiment using the SVM, we randomly extract 3333 incorrect sentences (16.6%) from the Error Corpus, omit 178 selected sentences that are composed of less than 4 words, and use the resting 3155 selected sentences as a part of the test set. The remaining part of the Error Corpus, which is composed of 16687 incorrect sentences (83.4%) and 25059 corrected sentences, is going to be used as the training set in the comparative experiment.

Since the one-class SVM is a novelty detection algorithm, the 3155 extracted incorrect sentences are combined with a significant number of clearly formed English sentences from the Open Portion of the *American National Corpus*. From *icic* (letters), *oup* (non-fiction) and *verbatim* (journal) files, 30682 sentences are chosen for the purpose and used to create the test set.

Table 2: Statistical information of the error sentences and test data : the numbers of sentences, the total number of words and the average number and variance of the number of words per sentence.

	Sentences	Words	Average	Variance
Error sentence	3333	42500	12.75	6.66
Test set	34015	777267	22.85	13.77

In preparation, we manually replace expressions of numbers with the same special word that represents all numbers, remove errors with punctuations, spelling and capitalization and then lemmatize verbs and nouns, when possible. British English spellings are converted into American ones. After the noise reduction process, we make the lists of bigram, trigram and four-gram language models using the test set.

3.2 Error Detection Procedure

When being input, the test set is split into a single string at the point, where a string ends and another string or a new line starts. In the following error detection process, all word sequences in the singled-out sentence are compared with the language models made in the preparatory part. If an expression in the input string matches one of the word sequence models in the list, we give the sentence the attribute number and its frequency of appearance. Once all the sentences are attached attribute numbers and 0 frequencies, they are assigned to two different groups in accordance with the classification algorithms. The next subsection provides an explanation of basic functions of support vector machines and one-class SVMs.

3.3 Support vector classifiers

Support vector classifiers approach classification problems with the concept of margin. For binary classification problems, the classifier first maps the original data vectors from the input space to a high-dimensional feature space using a kernel function and then defines a decision boundary in the feature space.

Given a training set $\{x_i, y_i\}_{i=1}^n$ where $x_i \in R^d$ is an input vector, $y_i \in \{-1, 1\}$ is the class labels and $\Phi: R^n \mapsto H$ is a linear mapping function, the decision boundary for a given feature vector is defined as $\omega^T \Phi(x) = 0$ and the discriminant function $f(x) = \omega^T \Phi(x) + b = 0$. Adding $b \in R$ to the discriminant function, the classifier assign a data vector x_i to the first class if

$$f(x_i) = \text{sgn}(\omega^T \Phi(x_i) + b) = 1$$

and to the second if

$$f(x_i) = \text{sgn}(\omega^T \Phi(x_i) + b) = -1$$

As illustrated in Figure 1, there may exist more than one separating hyperplanes, since the decision boundary can be placed anywhere in the feature space.

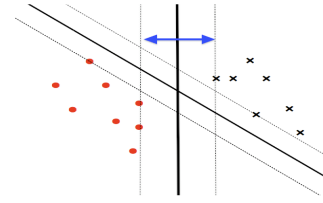


Figure 1: Optimal hyperplane and support vectors on its margin

To avoid potential pitfall of over fitting, the classifier applies the optimal hyperplane that maximizes the smallest perpendicular distance between the nearest data point(s) from two different classes. Derived from the following notations,

$$\omega^T \Phi(x_i) + b \leq -1, y_i = -1$$

$$\omega^T \Phi(x_i) + b \geq 1, y_i = 1$$

the separating hyperplane is written as

$$f(x) = \text{sgn}(\omega^T \Phi(x_i) + b)$$

3.4 Loss functions and ν -trick

In order to minimize the misclassification rate, the general support vector classifiers apply $r_{\text{hinge}}(yf(x)) = \max\{0, 1 - yf(x)\}$ as the loss function. For support vector regression, however, the loss function is often optimized for given data. When $\rho > 0$, the loss function $r_\rho(yf(x)) = \max\{0, \rho - yf(x)\}$ is the parallel transformation of r_{hinge} and has $\frac{\rho}{\|\omega\|}$ as the size of the margin. In that case, the region whose error rate is equal to 0 becomes larger as the value of ρ becomes smaller. To prevent ρ from converging to 0, a penalty term for r_ρ is proposed. This can also be written as

$$\min_{\xi, \alpha, \rho} \sum_{i=1}^n \xi_i + \frac{1}{2} \alpha^T K \alpha - \nu \rho$$

$$\text{s.t. } \rho \geq 0, \xi \geq 0, \xi_i \geq \rho - y_i \sum_{j=1}^n \alpha_j K_{ij}$$

One-class support vector machines employ this technique to find outliers in a given data set.

3.5 One-class support vector machines

A One-class support vector machine is an SVM extension devised to density estimation. When mapped to feature space using a radial basis function (RBF) as a kernel, a dataset $\{x_1, \dots, x_n\} \in R^d$ is expressed as

$$(k(x_i, x_1), \dots, k(x_i, x_n)) \in R^n, \quad i = 1, \dots, n.$$

Here, each data point x_i is represented by a single vector x_i in R^n . Because of the attributes of the RBF, $k(x, x') = \exp(-\sigma \|x - x'\|^2)$, the values of other than x_i can get very close to the origin, when the data point x_i does not have many neighboring data points. Thus, the norm of x_i is likely to be small when it is located in an area with few data points, and large in a densely marked region. The region of high density is expressed as :

$$f(x) = \sum \alpha_i k(x, x_i), \alpha, \rho \in R, \quad i = 1, \dots, n$$

One-class SVM make a density estimation in the following order

$$\min \frac{1}{2} K \alpha^T + C \sum_{i=1}^n \varepsilon_i \geq 0$$

$$\text{s.t. } \varepsilon \geq 0, \left(\sum_{j=1}^n \alpha_j K_{ij} - \rho \right) \geq 1, \quad i = 1, \dots, n$$

Solving this, the estimated result $\sum k(x, x_i) - \rho$ takes positive values at most data points and negative when the norm of $(k(x_i, x_1), \dots, k(x_i, x_n))$ is small. In other words, the one-class SVM builds a separating hyperplane between the densely populated region and the origin, around where outlying data points are mapped. We apply this function to identify unusual word sequences. The LIBSVM library is used in actual implementation. [3]

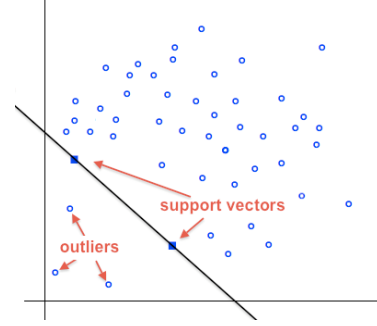


Figure 2: One-Class SVM and Novelty Detection

3.6 Comparative experiment

Following the research of Oyama and Matsumoto, we construct a supervised classification model based on LIBSVM. [3] Using the 14577 incorrect sentences and the 18115 error-free sentences that are not used as the test data from the Error Corpus (both nearly 5 times as large as the number of the sentences used in the test data) as the training data, the SVM classifier create a binary classification model. The test data created in the exceeding section are, then, assigned into two groups in accordance with the classification model. As the type of kernel is not specified in the previous research, a linear kernel is applied in this comparative experiment.

4. Results

The results of the experiments using the one-class SVM classification model are shown in the tables below. In these experiments, each sentence of the test data is analyzed using the frequency of appearance of n-gram language models. The table 3 provides the accuracy and the detailed information of each classification model. Together with the value of the parameters γ and ρ , the total number of support vectors is given.

Table 3: Results using the one-class SVM classifier :

Accuracy, the value of γ , the value of ρ and the number of support vectors.

	Accuracy	γ	ρ	SV
Bigram	79.30 %	.5	.50	22207
Trigram	86.63 %	.5	.88	29042
Four-gram	34.34 %	.5	1.44	22091

Table 4: The result of the experiment using Bigram : The number of sentences assigned to the sentences with errors group, and the number of sentences assigned to the sentences without error group.

	Assigned to Incorrect	Assigned to Correct
Incorrect sentences	3127 (99.11%)	28 (.88%)
Correct sentences	7011 (33.72%)	23849 (77.28%)

Table 5: The result of the experiment using Trigram

	Assigned to Incorrect	Assigned to Correct
Incorrect sentences	404 (12.80%)	2751 (87.19%)
Correct sentences	1794 (5.81%)	29066 (94.18%)

Table 6: The result of the experiment using Four-gram

	Assigned to Incorrect	Assigned to Correct
Incorrect sentences	1652 (52.36%)	1503 (47.63%)
Correct sentences	20862 (67.60%)	9998 (32.39%)

Table 7: Results of Supervised Error Detection based on SVM : Accuracy rate, the number of sentences contained in the training set, the number of feature values, and the number of support vectors

	Accuracy	Training set	Features	SV
Bigram	62.26%	39182	203951	36638
Trigram	75.93%	39182	379655	38217
Four-gram	45.79%	39182	453558	34969

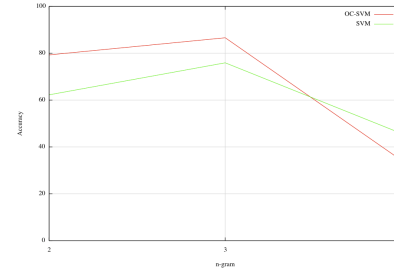


Figure 3: Comparison of the accuracy rates

5. Discussion

These results are assumed to reflect the attributes of n-gram models and their frequency of appearance. As the number of words contained in the n-gram gets larger, the frequency of appearance of each n-gram gets smaller. When frequency values of most word sequences become as small as those of peculiar expressions, it is virtually impossible for the algorithm to identify the outliers. As a matter of fact, the one-class SVM classifier produced better result than the SVM classifier in combination with bigram and trigram models.

On the other hand, the classification accuracy using four-gram model is equally low in the comparative experiment. That contradicts the result of the previous research explained in 2. The accuracy improves as the number of words included in the language model increases in [12], but this difference may stem from the difference of the language structure of English and Japanese, the target language of the previous research.

Although the method investigated in the research is not effective at finding the collocation and syntax errors which occur in longer word sequences, the application possibilities of single-class SVMs for the purpose is supported by the results of the experiments, especially when large corpora are hard to obtain.

6. Conclusion and Future Work

Combined with n-gram features, the unsupervised novelty detection algorithm achieves almost the same prediction accuracy as the supervised learning algorithms that require more computational costs. However, problems related to the distance between the elements of expressions, the attributes of the error corpus (including the first language of the writers) and contexts cannot be covered by the n-gram attributes employed in this research. Alongside with the improvement of the classification accuracies, problems related with the automation of error annotation remain unsolved. Furthermore, the application of the algorithm in researches on underrepresented languages and other error types are also expected.

Acknowledgements

We wish to thank Dr. Chih-Chung Chang and Dr. Chih-Jen Lin kindly for allowing us to use their libraries in this research and Dr. Randi Reppen, Dr. Nancy Ide and many other anonymous donators and helpers for sharing their corpus.

References

- Japanese learners English spoken data. *Companion Volume to the Proceedings of ACL '03*. pp.145-148
- [11] Nesselhauf, N. (2003). The Use of Collocations by Advanced Learners of English and Some Implications for Teaching. *Applied Linguistics*, 24(2). pp. 223-242
- [12] Oyama, H., Matsumoto, Y. (2008). A Machine Learning Approach for Error Identification for Learners of Japanese [in Japanese]. *The Society for Teaching Japanese as a Foreign Language Spring Meeting 2008*. pp.31-38
- [13] Tetreault, J. and Chodorow, M. (2009). Examining the use of region web counts for ESL error detection. *Web as Corpus Workshop (WAC-5)*.
- [1] Alam, M.J., UzZaman, N., Khan, M. (2006). N-gram based statistical checker for Bangla and English. *Proceedings of International Conference on Computer and Information Technology (ICCIIT)*.
- [2] Brockett, C., William D. B., and Gamon, M. (2006). Correcting ESL errors using phrasal SMT techniques. *ACL-44 Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, 2006.
- [3] Chang, C.C. and Lin, C. J. (2001). LIBSVM : a library for support vector machines Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
- [4] Leacock, C., Chodorow, M., Gamon, M. and Tetreault, J. (2010). *Automated Grammatical Error Detection for Language Learners (Synthesis Lectures on Human Language Technologies)*. Morgan and Claypool Publishers.
- [5] Lee, J.S.Y. (2009). Automatic Correction of Grammatical Errors in Nonnative English Text (Doctoral Dissertation at the Massachusetts Institute of Technology). Massachusetts Institute of Technology. Cambridge, MA
- [6] Han, N.,R., Chodorow, M. and Leacock, C. (2006). Detecting errors in English article usage by non-native speakers. *Natural Language Engineering*, 12 (2). Cambridge University Press. pp.115-129
- [7] Hermet, M., Désilets, A. and Szpakowicz, S. (2008). Using the web as a linguistic resource to automatically correct lexico-syntactic errors. *Proceedings of the 6th Conference on Language Resources and Evaluation (LREC)*. pp.874-878
- [8] Howarth, P. (1998). Phraseology and Second Language Proficiency. *Applied Linguistics*, 19/1. Oxford University Press.
- [9] Ide, N., Reppen, R., and Suderman, K. (2002). The American National Corpus: More than the Web Can Provide. *Proceedings of the Third Language Resources and Evaluation Conference (LREC)*. pp.839-844
- [10] Izumi,E., Uchimoto, K., Saiga, T., Supnithi, T., Isahara, H. (2003). Automatic error detection in the