Automatic Classification of Article Errors in L2 Written English

Aliva M. Pradhan[1], Aparna S. Varde[1],
Jing Peng[1], Eileen M. Fitzpatrick[2]

1. Department of Computer Science
2. Department of Linguistics
Montclair State University
One Normal Avenue
Montclair, NJ 07043, USA
pradhana1@mail.montcliar.edu vardea@mail.montclair.edu
pengj@mail.montclair.edu fitzpatricke@mail.montclair.edu

Abstract

This paper presents an approach to the automatic classification of article errors in non native (L2) English writing, using data cho sen from the MELD corpus that was purposely selected to contain only cases with article errors. We report on two experiments on the data: one to assess the performance of different machine learning algorithms in predicting correct article usage, and the other to determine the feasibility of using the MELD data to iden tify which linguistic properties of the noun phrase containing the article are the most salient with respect to the classification of er rors in article usage.

1. Introduction

Because English is so widely spoken, it has often been referred to as a "world language," the lingua franca of the modern era. Along with preposition choice, article usage remains the most common difficulty for non-native speakers of English, particularly written English. The ultimate goal of our work is to develop tools that provide feedback and suggestion to these writers to choose an appropriate article when required (or not to use an article when not required). Our intermediate goal is to identify the cases of article usage that pose the most difficulty for writers based on the linguistic reason for the presence or absence of an article. At this stage, we do not distinguish between the appropriateness of *a(n)* vs. *the*; we considered only whether the context required an article or not.

In this paper, we first describe the performance of machine learning algorithms as classifiers. Classification is the process of predicting the target class of a given attribute based on a study of existing data. In this work, we use classification in order to predict article usage and related properties. The classifiers we use are pre-defined in the WEKA software suite [9] with respect to their ability to identify article errors in the MELD data set [6]. MELD is a corpus of the academic writing of adult, non-native speakers of English, with 50,000 words of the corpus manually annotated for error. The annotation includes error correction. The time and cost of this annotation accounts for the current small data set. The native language backgrounds of the writers include languages with no articles (Bengali, Hindi, Gujarati, Malayalam, Mandarin, Polish, Taiwanese, Vietnamese) and languages with different types and usage of articles (Arabic, Haitian Creole, Spanish).

We trained the classifiers on 100 noun phrases (NPs) extracted from the corpus, all of which contain article errors. The system uses local contextual features in the form of part of speech tags and a +/-count feature on the head noun to compute the confusion matrix for the purpose of classifying the data set.

The two main objectives of the paper are to predict article distribution in cases where L2 English writers make errors, and to determine the feasibility of using the MELD data to identify which linguistic properties of the noun phrase are the most salient with respect to the classification of errors in article usage. In order to predict article distribution, we use several machine learning classifiers that serve the purpose of prediction or classification. After conducting evaluation with multiple classifiers, we also aim to test the feasibility of classifying errors in article usage to determine if it would be possible to rank article errors by cause of error. Single tree classifiers are the most suitable for this task because they allow us to relate tree paths to the respective classes depicted by leaves of the tree. Among the single tree classifiers, the J4.8 algorithm was found to yield the best results and therefore we use J4.8 decision trees to achieve the second objective of providing an estimate for determining the causes of article errors. The identification of cause of error is intended to serve as an input to language teachers and developers of intelligent computer based tutors who seek to prioritize the causes of article errors.

The rest of this paper is organized as follows. Section 2 presents an overview of related work in the area. Section 3 explains our proposed approach to article error classification. Section 4 summarizes our experimental evaluation on the given corpus. Section 5 describes the analysis of our experimental results from a linguistic angle. Section 6 gives the conclusions.

2. Related Work

Izumi et al. [4] apply a maximum entropy classifier trained on transcribed speech data, annotated for error, to detect errors in non-native English. The features used for classification are the two words on either side of the targeted word, part-of-speech class of these words, and the root form of the targeted word. Transcribed speech data was the basis for this analysis. The recall and precision on omission-type article errors were 30% and 52% respectively, indicating the great difficulty of this particular task.

Han, et al. [3] make use of a maximum entropy classifier trained on published (error-free) text to test error detection of article usage on written TOEFL (Test of English as a Foreign Language) data. The features used are words in the noun phrase (NP) containing the targeted word, part-of-speech class of the words in the NP, positions relative to NP (Noun Phrase) boundaries, and +/- count judgments of the head noun derived from corpus-based frequency measures, giving precision and recall results of 52% and 80% respectively.

Gamon et al. [2] train decision tree classifiers to recognize incorrect use of determiners and prepositions in ESL (English as a Second Language) texts. However, the evaluation over non-native text is not automated.

Lee et al. [5] work on ranking and classification of non-native sentences using Support Vector Machines, considering the use of machine-translated data as a substitute for original non-native writing samples. Classification with machine-translated data is not found to be as good as that with original non-native data, although ranking is found to work well with both.

The work described here is distinct from prior work in its use of error annotated data from written English and in its testing of multiple classifiers for their ability to identify error in article usage.

3. Proposed Approach

We have chosen a somewhat harder task: predicting the presence or absence of an article using only cases in which English learner writers have made errors. Our ultimate goal is not only to predict presence or absence of an article, but also to gain an understanding, using machine learning techniques, of which rules governing article usage are the most difficult for learners. We selected from MELD fifty essays on a variety of topics. The error-annotated essays were then tagged using a part of speech tagger [1] for the purpose of selecting the features that feed the classifier. Testing of the tagger on 3072 words (1521 uncorrected for error and 1551 corrected) found the same 22 tagging errors in both sets, an additional 4 tagging errors in the uncorrected set and 2 tagging errors in the corrected set, giving the tagging a high reliability with respect to data containing non-native English speaker errors.

3.1 Description of Classifiers

We describe the use of J4.8, Bagging, AdaBoost and Random Forest classifiers to predict article usage. We chose these because they fall into the category of decision trees and ensemble learning which intuitively seemed appropriate to use in our evaluation given the task of predicting article errors. Accordingly we describe these classifiers because they were in fact found to produce good results with our datasets. These classifiers are briefly described below.

J4.8: The J4.8 is a classifier that builds a single decision tree over a given data set. A decision tree consists of a root, several branches, nodes and leaves, such that the root depicts the starting point, the branches indicate the various paths, the nodes depict intermediate steps and the leaves indicate the final actions or decisions for each complete path. J4.8 is a Java-based decision tree learning algorithm proposed by Quinlan [8], with C4.5 as its C-based equivalent, such that it gets trained with a data set using induction and then returns predictions for single data rows. The type of the evaluation is important.

Bagging: The concept of bagging is used in ensemble learning to run a classification algorithm several times, by sampling with replacement. Thus, a bagged tree classifier is created from sampling the training data multiple times with replacement [8]. That is, a bagged tree is obtained by combining many independently generated trees, by replicated bootstrap sampling of the given data.

RandomForest: Random Forest is an ensemble classifier that consists of many decision trees and outputs the class that is the statistical mode [10]. In other words, it outputs the most frequently occurring value of the class's output by individual trees.

AdaBoost: AdaBoost is a meta-algorithm, and can be used in conjunction with many other learning algorithms to improve their performance [10]. AdaBoost is adaptive in the sense that subsequent classifiers built are tweaked in favor of those instances misclassified by previous classifiers, thereby serving to boost performance.

3.2 Non-native speaker text

For the article usage task, we extracted 100 instances known to be of incorrect article usage and asked the classifier to predict them as correct or incorrect. The following examples from MELD show typical errors in article usage:

1. She longs for the real friendship.
2. They do not have same closeness with their family.

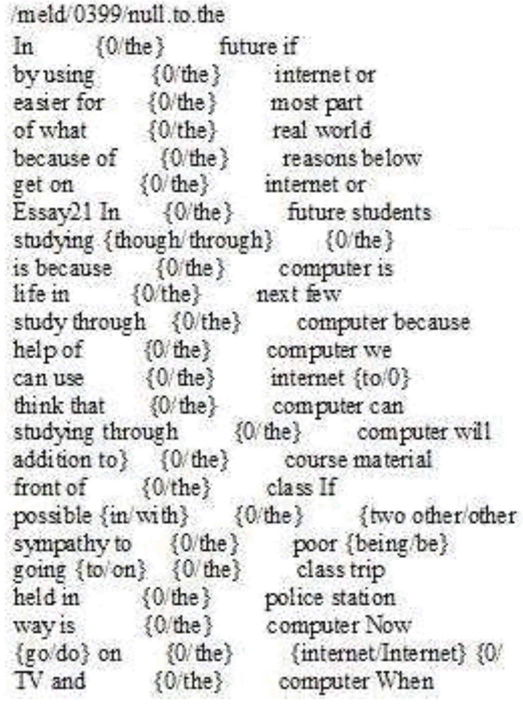
The input consists of cases with *the* and *a(n)* incorrectly present, as in (1) above, or incorrectly absent, as in (2). Our goal is to predict the correct usage in each case.

3.3 Feature Selection

In the classifier experiments, we used the following features to predict the presence or absence of the article, with the position of the article as w: the part of speech of w+1, w+2, w-1, w-2, the count/non-count status of the head noun, the feature w+1 (the lexical word following the article) and the classification of the article usage as correct or incorrect.

4. Experimental Evaluation

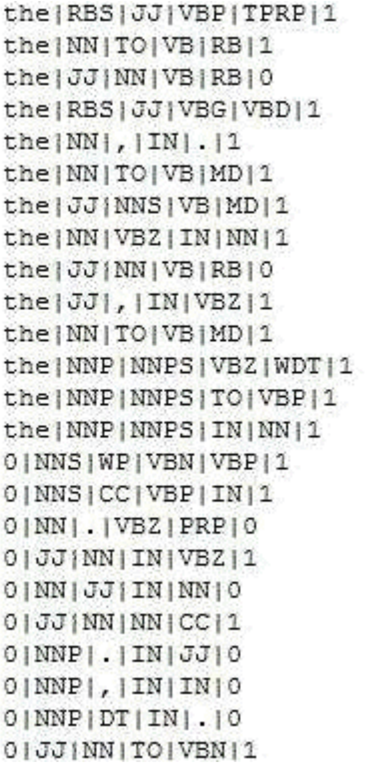
For conducting evaluation, we used the well-known WEKA suite (Waikato Environment for Knowledge Analysis) [9], which provides an implementation of various classifiers. We summarize the results of our experimentation in two parts below. The first part focuses on the use of machine learning algorithms as classifiers for various targets such as the class of the article usage error itself and its related attributes. The second one focuses on the linguistic properties to set the stage for discovering the causes of article errors.



```
/meld/0399/null.to.the
In          {0/the}        future if
by using         {0/the}        internet or
easier for       {0/the}        most part
of what          {0/the}        real world
because of       {0/the}        reasons below
get on           {0/the}        internet or
Essay21 In       {0/the}        future students
studying {though/through}        {0/the}
is because       {0/the}        computer is
life in          {0/the}        next few
study through    {0/the}        computer because
help of          {0/the}        computer we
can use          {0/the}        internet {to/0}
think that       {0/the}        computer can
studying through        {0/the}        computer will
addition to}     {0/the}        course material
front of         {0/the}        class If
possible {in/with}      {0/the}        {two other/other
sympathy to      {0/the}        poor {being/be}
going {to/on}    {0/the}        class trip
held in          {0/the}        police station
way is           {0/the}        computer Now
{go/do} on       {0/the}        {internet/Internet} {0/
TV and           {0/the}        computer When
```

Figure1: MELD Corpus Snapshot

Figure 1 shows a partial snapshot of the MELD corpus. Figure 2 shows a sample of tagged data from the corpus after preprocessing. The first line in Figure 2 shows the incorrect presence of 'the' (w) in the student's phrase, the position of w+1, the position of w+2, the position of w-1, the position of w-2, and the count status of the head noun (such that 1 = count noun).



```
the|RBS|JJ|VBP|TPRP|1
the|NN|TO|VB|RB|1
the|JJ|NN|VB|RB|0
the|RBS|JJ|VBG|VBD|1
the|NN|,|IN|.|1
the|NN|TO|VB|MD|1
the|JJ|NNS|VB|MD|1
the|NN|VBZ|IN|NN|1
the|JJ|NN|VB|RB|0
the|JJ|,|IN|VBZ|1
the|NN|TO|VB|MD|1
the|NNP|NNPS|VBZ|WDT|1
the|NNP|NNPS|TO|VBP|1
the|NNP|NNPS|IN|NN|1
0|NNS|WP|VBN|VBP|1
0|NNS|CC|VBP|IN|1
0|NN|.|VBZ|PRP|0
0|JJ|NN|IN|VBZ|1
0|NN|JJ|IN|NN|0
0|JJ|NN|NN|CC|1
0|NNP|.|IN|JJ|0
0|NNP|,|IN|IN|0
0|NNP|DT|IN|.|0
0|JJ|NN|TO|VBN|1
```

Figure 2: Sample of Tagged Data

The following experimental results are with *error* as the classification target, i.e., the goal is to predict the occurrence of error in article usage. Note that the "*error*" column here relates to the accuracy of classification in predicting article usage error.

In Figure 3, we summarize the results of our evaluation using regular decision trees with J4.8 (the Single Tree Approach). The type of error being considered is the presence or absence of *the* as {0/the} or {the/0}. Figure 4 gives a summary of the evaluation using bagging of decision trees (the Bagged Tree Approach). Predicted and actual values for articles are shown with respect to the test set, along with the error in prediction and probability distribution.

We found that bagged trees perform better, which could be due to the fact that repeated sampling is performed with replacement thus increasing randomization and giving more robustness in classification. We also conducted experiments with other ensemble classifiers using article usage error as the classification target and they gave us accuracy approximately in the range similar to bagged trees.

```
Instance actual  predicted  error   probability distribution
    1       2:1      2:1                 0.294 *0.706
    2       2:1      2:1                 0.294 *0.706
    3       2:1     1:-1          +     *0.714  0.286
    4       2:1      2:1                 0.294 *0.706
    5       2:1      2:1                 0.294 *0.706
    6       2:1      2:1                 0.294 *0.706
    7       2:1      2:1                 0.294 *0.706
    8       2:1      2:1                 0.294 *0.706
    9       2:1     1:-1          +     *0.714  0.286
   10       2:1      2.1                 0.294 *0.706
   11       2:1      2:1                 0.294 *0.706
   12       2:1      2:1                 0.294 *0.706
   13       2:1      2:1                 0.294 *0.706
   14       2:1      2:1                 0.294 *0.706
   15      1:-1      2:1          +      0.294 *0.706
```

Figure 3: Single Tree Approach for Classification

```
Instance actual   predicted error  probability distribution
    1       2:1     2:1                 0.338 *0.662
    2       2:1     2:1                 0.288 *0.712
    3       2:1     2:1                 0.488 *0.512
    4       2:1     2:1                 0.391 *0.609
    5       2:1     2:1                 0.455 *0.545
    6       2:1     2:1                 0.288 *0.712
    7       2:1     2:1                 0.341 *0.659
    8       2:1     2:1                 0.347 *0.653
    9       2:1     2:1                 0.488 *0.512
   10       2:1     1:-1          +    *0.538  0.462
   11       2:1     2:1                 0.288 *0.712
   12       2:1     2:1                 0.288 *0.712
   13       2:1     2:1                 0.388 *0.612
   14       2:1     2:1                 0.347 *0.653
   15      1:-1     1:-1               *0.588  0.412
```

Figure 4: Bagged Tree Approach for Classification

We used J4.8 to test one of the conditions that determines article usage: the +/-count distinction. This is because J4.8 is a single tree classifier and therefore it is possible to track back the original conditions through the paths of the tree, in order to find the causes of error in article usage. Although bagged trees give better results, they cannot be used to backtrack and trace the condition, because there is a combination of multiple trees produced by bagging. The goal in these experiments is to predict whether the erroneous cases were due to incorrect classification of the head noun with respect to the +/- count distinction or not. These experiments serve the dual purpose of comparing the classifiers and setting the stage for further experimentation and analysis to find the causes of article errors.

To test the ability of a classifier to recognize the conditions for prediction, we tested on a single condition, the high frequency +/-count. We ran the test using each of the ensemble classifiers and the single tree classifier. Here we show the results of the RandomForest ensemble classifier and the J4.8 single tree classifier in classifying the target *count noun*. We have used cross validation for training and testing, which allows us to have multiple iterations of testing, keeping the training set and test sets distinct in each iteration or fold. If the dataset has 100 instances, then in each fold, 90 instances are used as the training set and 10 as the test set, and likewise the testing process is performed 10 times. This is to provide greater robustness and generality to the learned hypothesis. Note that while we have chosen to show the results from only two classifiers on classification of error and count noun class, we have conducted experiments with all classifiers on both categories. We only present a summary of our evaluation here.

Figure 5 is a snapshot of the results obtained with the RandomForest classifier with *count noun* as the target.

```
weka.classifiers.trees.RandomForest  I 10  K 0  S 1
Relation:    meld data
Instances:   37
Attributes:  6
w+1
w+2
w 1
w 2
correct
count
Test mode:10 fold cross validation
Time taken to build model: 0 seconds
Correctly Classified Instances        34   91.8919 %
Incorrectly Classified Instances       3   8.1081 %
Total Number of Instances             37
      Confusion Matrix
a  b   <   classified as
1  3 |  a    0
0 33 |  b    1
```

Figure 5: RandomForest Classifier for Target *Count Noun*

Figure 6 represents the result snapshot for the J4.8 classifier with the target being *count noun.*

```
Scheme:      weka.classifiiers.J4.8
Relation:    meld data
Instances:   103
Attributes:  6
 w+1
 w+2
 w 1
 w 2
 correct
 count
Test mode:   split 66.0% train, remainder test
Time taken to build model: 0 seconds
      Evaluation on test split
      Summary
Correctly Classified Instances        30              85.7143 %
Incorrectly Classified Instances       5              14.2857 %
Total Number of Instances             35

      Confusion Matrix
  a  b   <   classified as
  0  5 |  a    0
  0 30 |  b    1
```

Figure 6: J4.8 Classifier for Target *Count Noun*

While the result of the ensemble classifier test on count noun prediction, i.e., 91% of cases of +/- count correctly predicted, is better than the result of the single tree classifier at 85%, ensemble classifiers as noted above do not give us the ability to linguistically analyze individual cases where students had trouble with article use because of the count properties of the head noun. We therefore used the single tree approach of J4.8 to be able to determine why prediction of the +/- count distinction might go awry.

5. Can Noun Phrase (NP) Features be used to Determine Cause of Error?

There is a bifurcation of article usage in English: nouns can be modified by *a(n)* or *the*, depending on definiteness of the noun, and nouns can be modified by an article or not depending on a variety of conditions, including whether the head noun is a count noun (*eat the apple, eat meat*), a collective noun (*send the letter, send mail*), a proper noun (*the bill, Bill*), a noun modified by an adjective (*the United Kingdom, Britain*), a non-specific institution (*go to the store, go to school*), a meal (*eat a sandwich, eat lunch*), or a variety of smaller categories. As far as we know, there is no knowledge of which of these conditions is more problematic for English language learners. Knowing which conditions cause the most problems for learners would help teachers and textbooks prioritize instruction. It would also provide input to computer based tutors for English language learners.

The classification target being *count noun,* Figure 6 above depicts the classification results using J4.8 to predict whether the erroneous cases contained a count noun corresponding to the article.

The J4.8 classifier had a 14.3% error rate in classifying the head noun as +/-count, showing that the classifier can be used to prioritize the conditions (count, collective, etc.) that cause a head noun to require an article. The cases of +/- count errors that failed to be correctly classified as such involved primarily polysemous nouns where an understanding of the proper sense of the noun is needed to determine whether an article is required. For example, *education* requires an article when it refers to a kind or stage of the process (*a liberal education*), but not when it refers to the process itself (*public education*). It is not possible to account for this type of count classification error without recourse to a larger native speaker corpus of English that provides co-occurrence information to disambiguate the polysemous head noun. Nevertheless, the 85% success in identifying an error as due to the +/- count status of the head noun leads us to believe that we can identify the linguistic conditions that dictate article usage and rank them with respect to their correlation with error in article usage.

6. Conclusions

In this paper, we have addressed the problem of automatically classifying errors in non-native or L2 written English using real data from an online corpus. We have evaluated several machine learning algorithms for their ability to predict the occurrence of error and we have used one classifier, J4.8, to predict the presence of a count noun in cases where non-native writers of English demonstrated erroneous article usage.

The primary task we set ourselves predicting article distribution using only the cases in which writers make errors is a more difficult task than previous work, which includes low hanging fruit. We anticipate that building a classifier trained on this difficult dataset will maximize our performance on a larger dataset containing easier cases. However, considering both the difficulty of the task and the small size of the data set, the results of the bagged tree approach are promising. The overall accuracy rate of 70% using the bagged tree classifier is substantially higher than in previous work based on non-native speaker data and approaches the results of [3] despite the small size of the data set.

The secondary task of identifying one of the conditions that determines article usage with the goal of prioritizing the learner difficulty with these conditions shows a solid result, with 85% of the cases identified, and the primary reason for difficulty (polysemous nouns) apparent in the remaining cases.

Future work includes expanding the MELD corpus to allow for more robust testing, correlation of the conditions that determine article usage with erroneous usage and ranking of the conditions in terms of degree of correlation, and detailed comparison with state-of-the-art approaches such as maximum entropy for various error detection tasks [2,3,4,5,7]. We hope that our current and ongoing work makes further contributions to linguistics and machine learning and also motivates the development of artificial intelligence tools such as computer based tutors.

References

[1] Brill, E. A simple rule based part of speech tagger. Proceedings of the Workshop on Speech and Natural Language (February 1992), Harriman, N.Y.

[2] Gamon, M., Gao, J., Brockett, C., Klementiev, A., Dolan, B., Belenko, D. and Vanderwende L.:Using Contextual Speller Techniques and Language Modeling for ESL Error Correction. Proceedings of the International Joint Conference on Natural Language Processing, Hyderabad, India (January 2008), 449 456.

[3] Han, N., Chodorow, M. and Leacock, C.: Detecting Errors in English Article Usage by Non native Speakers. Natural Language Engineering, Cambridge University Press, United Kingdom (2006), Vol. 12(2), 115 129.

[4] Izumi, E., Uchimoto, K., Saiga, T., Thepchai, S. and Isahara, H.: Automatic Error Detection in the Japanese Learners' English Spoken Data. Proceedings of the 41st Annual Meeting on Asso

ciation for Computational Linguistics, Sapporo, Japan (July 2003), Vol. 2, 145 148.

[5] Lee, J., Zhou, M. and Xiaohua L.: Detection of Non native Sentences using Machine translated Training Data. Proceedings of the North American Chapter of the Association for Computational Linguistics Human Language Technologies, Rochester, NY (April 2007), 93 96.

[6] MELD: Montclair Electronic Language Database, www.chss.montclair.edu/linguistics/MELD

[7] Park, T., Lank., E., Poupart, P. and Terry, M.: ”Is the Sky Pure Today?” AwkChecker: An Assistive tool for Detecting and Correcting Collocation Errors. ACM Symposium on User Interface Software and Technology, Monterey, California (October 2008), 121 130.

[8] Quinlan, J.R.: Bagging, Boosting and C4.5. Proceedings of the 13th National Conference of the American Association of Artificial Intelligence, Portland, Oregon (1996), 725 730.

[9] WEKA3: Data Mining Software in Java, Waikato Environment for Knowledge Analysis, University of Waikato, New Zealand (2005).

[10] Witten I. and Frank E.: Data Mining Practical Machine Learning Tools and Techniques (2nd Edition), Morgan Kaufmann (June 2005).