Detecting Erroneous Sentences using Automatically Mined Sequential Patterns

Guihua Sun *
Chongqing University
sunguihua5018@163.com

Xiaohua Liu **Gao Cong** **Ming Zhou**
Microsoft Research Asia
{xiaoliu, gaocong, mingzhou}@microsoft.com

Zhongyang Xiong
Chongqing University
zyxiong@cqu.edu.cn

John Lee †
MIT
jsylee@mit.edu

Chin-Yew Lin
Microsoft Research Asia
cyl@microsoft.com

Abstract

This paper studies the problem of identifying erroneous/correct sentences. The problem has important applications, e.g., providing feedback for writers of English as a Second Language, controlling the quality of parallel bilingual sentences mined from the Web, and evaluating machine translation results. In this paper, we propose a new approach to detecting erroneous sentences by integrating pattern discovery with supervised learning models. Experimental results show that our techniques are promising.

1 Introduction

Detecting erroneous/correct sentences has the following applications. First, it can provide feedback for writers of English as a Second Language (ESL) as to whether a sentence contains errors. Second, it can be applied to control the quality of parallel bilingual sentences mined from the Web, which are critical sources for a wide range of applications, such as statistical machine translation (Brown et al., 1993) and cross-lingual information retrieval (Nie et al., 1999). Third, it can be used to evaluate machine translation results. As demonstrated in (Corston-Oliver et al., 2001; Gamon et al., 2005), the better human reference translations can be distinguished from machine translations by a classification model, the worse the machine translation system is.

The previous work on identifying erroneous sentences mainly aims to find errors from the writing of ESL learners. The common mistakes (Yukio et al., 2001; Gui and Yang, 2003) made by ESL learners include spelling, lexical collocation, sentence structure, tense, agreement, verb formation, wrong Part-Of-Speech (POS), article usage, etc. The previous work focuses on grammar errors, including tense, agreement, verb formation, article usage, etc. However, little work has been done to detect sentence structure and lexical collocation errors.

Some methods of detecting erroneous sentences are based on manual rules. These methods (Heidorn, 2000; Michaud et al., 2000; Bender et al., 2004) have been shown to be effective in detecting certain kinds of grammatical errors in the writing of English learners. However, it could be expensive to write rules manually. Linguistic experts are needed to write rules of high quality; Also, it is difficult to produce and maintain a large number of non-conflicting rules to cover a wide range of grammatical errors. Moreover, ESL writers of different first-language backgrounds and skill levels may make different errors, and thus different sets of rules may be required. Worse still, it is hard to write rules for some grammatical errors, for example, detecting errors concerning the articles and singular plural usage (Nagata et al., 2006).

Instead of asking experts to write hand-crafted rules, statistical approaches (Chodorow and Leacock, 2000; Izumi et al., 2003; Brockett et al., 2006; Nagata et al., 2006) build statistical models to identify sentences containing errors. However, existing

*Work done while the author was a visiting student at MSRA
†Work done while the author was a visiting student at MSRA

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 81–88,
Prague, Czech Republic, June 2007.

statistical approaches focus on some pre-defined errors and the reported results are not attractive. Moreover, these approaches, e.g., (Izumi et al., 2003; Brockett et al., 2006) usually need errors to be specified and tagged in the training sentences, which requires expert help to be recruited and is time consuming and labor intensive.

Considering the limitations of the previous work, in this paper we propose a novel approach that is based on pattern discovery and supervised learning to successfully identify erroneous/correct sentences. The basic idea of our approach is to build a machine learning model to automatically classify each sentence into one of the two classes, "erroneous" and "correct." To build the learning model, we automatically extract *labeled sequential patterns* (LSPs) from both erroneous sentences and correct sentences, and use them as input features for classification models. Our main contributions are:

- We mine *labeled sequential patterns*(LSPs) from the preprocessed training data to build leaning models. Note that LSPs are also very different from N-gram language models that only consider continuous sequences.
- We also enrich the LSP features with other automatically computed linguistic features, including lexical collocation, language model, syntactic score, and function word density. In contrast with previous work focusing on (a specific type of) grammatical errors, our model can handle a wide range of errors, including grammar, sentence structure, and lexical choice.
- We empirically evaluate our methods on two datasets consisting of sentences written by Japanese and Chinese, respectively. Experimental results show that *labeled sequential patterns* are highly useful for the classification results, and greatly outperform other features. Our method outperforms Microsoft Word03 and ALEK (Chodorow and Leacock, 2000) from Educational Testing Service (ETS) in some cases. We also apply our learning model to machine translation (MT) data as a complementary measure to evaluate MT results.

The rest of this paper is organized as follows. The next section discusses related work. Section 3 presents the proposed technique. We evaluate our proposed technique in Section 4. Section 5 concludes this paper and discusses future work.

2 Related Work

Research on detecting erroneous sentences can be classified into two categories. The first category makes use of hand-crafted rules, e.g., template rules (Heidorn, 2000) and mal-rules in context-free grammars (Michaud et al., 2000; Bender et al., 2004). As discussed in Section 1, manual rule based methods have some shortcomings.

The second category uses statistical techniques to detect erroneous sentences. An unsupervised method (Chodorow and Leacock, 2000) is employed to detect grammatical errors by inferring negative evidence from TOEFL administrated by ETS. The method (Izumi et al., 2003) aims to detect omission-type and replacement-type errors and transformation-based leaning is employed in (Shi and Zhou, 2005) to learn rules to detect errors for speech recognition outputs. They also require specifying error tags that can tell the specific errors and their corrections in the training corpus. The phrasal Statistical Machine Translation (SMT) technique is employed to identify and correct writing errors (Brockett et al., 2006). This method must collect a large number of parallel corpora (pairs of erroneous sentences and their corrections) and performance depends on SMT techniques that are not yet mature. The work in (Nagata et al., 2006) focuses on a type of error, namely mass vs. count nouns. In contrast to existing statistical methods, our technique needs neither errors tagged nor parallel corpora, and is not limited to a specific type of grammatical error.

There are also studies on automatic essay scoring at document-level. For example, E-rater (Burstein et al., 1998), developed by the ETS, and Intelligent Essay Assessor (Foltz et al., 1999). The evaluation criteria for documents are different from those for sentences. A document is evaluated mainly by its organization, topic, diversity of vocabulary, and grammar while a sentence is done by grammar, sentence structure, and lexical choice.

Another related work is Machine Translation (MT) evaluation. Classification models are employed in (Corston-Oliver et al., 2001; Gamon et al., 2005) to evaluate the well-formedness of machine translation outputs. The writers of ESL and MT normally make different mistakes: in general, ESL writers can write overall grammatically correct sentences with some local mistakes while MT outputs normally produce locally well-formed phrases with overall grammatically wrong sentences. Hence, the manual features designed for MT evaluation are not applicable to detect erroneous sentences from ESL learners.

LSPs differ from the traditional sequential patterns, e.g., (Agrawal and Srikant, 1995; Pei et al., 2001) in that LSPs are attached with class labels and we prefer those with discriminating ability to build classification model. In our other work (Sun et al., 2007), labeled sequential patterns, together with labeled tree patterns, are used to build pattern-based classifier to detect erroneous sentences. The classification method in (Sun et al., 2007) is different from those used in this paper. Moreover, instead of labeled sequential patterns, in (Sun et al., 2007) the most significant k labeled sequential patterns with constraints for each training sentence are mined to build classifiers. Another related work is (Jindal and Liu, 2006), where sequential patterns with labels are used to identify comparative sentences.

3 Proposed Technique

This section first gives our problem statement and then presents our proposed technique to build learning models.

3.1 Problem Statement

In this paper we study the problem of identifying erroneous/correct sentences. A set of training data containing correct and erroneous sentences is given. Unlike some previous work, our technique requires neither that the erroneous sentences are tagged with detailed errors, nor that the training data consist of parallel pairs of sentences (an error sentence and its correction). The erroneous sentence contains a wide range of errors on grammar, sentence structure, and lexical choice. We do not consider spelling errors in this paper.

We address the problem by building classification models. The main challenge is to automatically extract representative features for both correct and erroneous sentences to build effective classification models. We illustrate the challenge with an example. Consider an erroneous sentence, "*If* Maggie *will* go to supermarket, she *will* buy a bag for you." It is difficult for previous methods using statistical techniques to capture such an error. For example, N-gram language model is considered to be effective in writing evaluation (Burstein et al., 1998; Corston-Oliver et al., 2001). However, it becomes very expensive if $N > 3$ and N-grams only consider continuous sequence of words, which is unable to detect the above error *"if...will...will"*.

We propose *labeled sequential patterns* to effectively characterize the features of correct and erroneous sentences (Section 3.2), and design some complementary features (Section 3.3).

3.2 Mining *Labeled Sequential Patterns* (LSP)

Labeled Sequential Patterns (LSP). A labeled sequential pattern, p, is in the form of $\texttt{LHS} \rightarrow \texttt{c}$, where $\texttt{LHS}$ is a sequence and $\texttt{c}$ is a class label. Let I be a set of items and L be a set of class labels. Let D be a sequence database in which each tuple is composed of a list of items in I and a class label in L. We say that a sequence $s_1 =< a_1, ..., a_m >$ is *contained* in a sequence $s_2 =< b_1, ..., b_n >$ if there exist integers $i_1, ...i_m$ such that $1 \leq i_1 < i_2 < ... < i_m \leq n$ and $a_j = b_{i_j}$ for all $j \in 1, ..., m$. Similarly, we say that a LSP p_1 is contained by p_2 if the sequence $p_1.LHS$ is contained by $p_2.LHS$ and $p_1.c = p_2.c$. Note that it is not required that s_1 appears continuously in s_2. We will further refine the definition of "contain" by imposing some constraints (to be explained soon). A LSP p is attached with two measures, *support* and *confidence*. The support of p, denoted by $\texttt{sup}(p)$, is the percentage of tuples in database D that contain the LSP p. The probability of the LSP p being true is referred to as "the confidence of p ", denoted by $\texttt{conf}(p)$, and is computed as $\frac{\texttt{sup}(p)}{\texttt{sup}(p.LHS)}$. The support is to measure the generality of the pattern p and minimum confidence is a statement of predictive ability of p.

Example 1: Consider a sequence database containing three tuples $t_1 = (< a, d, e, f >, E)$, $t_2 = (< a, f, e, f >, E)$ and $t_3 = (< d, a, f >, C)$. One example LSP $p_1 = < a, e, f > \rightarrow E$, which is contained in tuples t_1 and t_2. Its support is 66.7% and its confidence is 100%. As another example, LSP p_2

= $< a, f > \rightarrow E$ with support 66.7% and confidence 66.7%. p_1 is a better indication of class E than p_2. □

Generating Sequence Database. We generate the database by applying Part-Of-Speech (POS) tagger to tag each training sentence while keeping function words[1] and time words[2]. After the processing, each sentence together with its label becomes a database tuple. The function words and POS tags play important roles in both grammars and sentence structures. In addition, the time words are key clues in detecting errors of tense usage. The combination of them allows us to capture representative features for correct/erroneous sentences by mining LSPs. Some example LSPs include "*<a, NNS>* $\rightarrow$ $Error$"(singular determiner preceding plural noun), and "*<yesterday, is>* $\rightarrow Error$". Note that the confidences of these LSPs are not necessary 100%.

First, we use MXPOST-Maximum Entropy Part of Speech Tagger Toolkit[3] for POS tags. The MXPOST tagger can provide fine-grained tag information. For example, noun can be tagged with "NN"(singular noun) and "NNS"(plural noun); verb can be tagged with "VB", "VBG", "VBN", "VBP", "VBD" and "VBZ". Second, the function words and time words that we use form a key word list. If a word in a training sentence is not contained in the key word list, then the word will be replaced by its POS. The processed sentence consists of POS and the words of key word list. For example, after the processing, the sentence "*In the past, John was kind to his sister*" is converted into "*In the past, NNP was JJ to his NN*", where the words "*in*", "*the*", "*was*", "*to*" and "*his*" are function words, the word "*past*" is time word, and "*NNP*", "*JJ*", and "*NN*" are POS tags.

Mining LSPs. The length of the discovered LSPs is flexible and they can be composed of contiguous or distant words/tags. Existing frequent sequential pattern mining algorithms (e.g. (Pei et al., 2001)) use *minimum support* threshold to mine frequent sequential patterns whose support is larger than the threshold. These algorithms are not sufficient for our problem of mining LSPs. In order to ensure that all our discovered LSPs are discriminating and are capable of predicting correct or erroneous sentences, we impose another constraint *minimum confidence*. Recall that the higher the confidence of a pattern is, the better it can distinguish between correct sentences and erroneous sentences. In our experiments, we empirically set *minimum support* at 0.1% and minimum confidence at 75%.

Mining LSPs is nontrivial since its search space is exponential, althought there have been a host of algorithms for mining frequent sequential patterns. We adapt the frequent sequence mining algorithm in (Pei et al., 2001) for mining LSPs with constraints.

Converting LSPs to Features. Each discovered LSP forms a binary feature as the input for classification model. If a sentence includes a LSP, the corresponding feature is set at 1.

The LSPs can characterize the correct/erroneous sentence structure and grammar. We give some examples of the discovered LSPs. (1) LSPs for erroneous sentences. For example, "*<this, NNS>*"(e.g. contained in "*this books* is stolen."), "*<past, is>*"(e.g. contained in "in the *past*, John *is* kind to his sister."), "*<one, of, NN>*"(e.g. contained in "it is *one of* important working *language*", "*<although, but>*"(e.g. contained in "*although* he likes it, *but* he can't buy it."), and "*<only, if, I, am>*"(e.g. contained in "*only if* my teacher has given permission, *I am* allowed to enter this room"). (2) LSPs for correct sentences. For instance, "*<would, VB>*"(e.g. contained in "he *would buy* it."), and "*<VBD, yeserday>*"(e.g. contained in "I *bought* this book *yesterday*.").

3.3 Other Linguistic Features

We use some linguistic features that can be computed automatically as complementary features.

Lexical Collocation (LC) Lexical collocation error (Yukio et al., 2001; Gui and Yang, 2003) is common in the writing of ESL learners, such as "*strong tea*" but not "*powerful tea.*" Our LSP features cannot capture all LCs since we replace some words with POS tags in mining LSPs. We collect five types of collocations: verb-object, adjective-noun, verb-adverb, subject-verb, and preposition-object from a general English corpus[4]. Correct LCs are collected

[1]http://www.marlodge.supanet.com/museum/funcword.html

[2]http://www.wjh.harvard.edu/%7Einquirer/Time%40.html

[3]http://www.cogsci.ed.ac.uk/∼jamesc/taggers/MXPOST.html

[4]The general English corpus consists of about 4.4 million native sentences.

by extracting collocations of high frequency from the general English corpus. Erroneous LC candidates are generated by replacing the word in correct collocations with its confusion words, obtained from WordNet, including synonyms and words with similar spelling or pronunciation. Experts are consulted to see if a candidate is a true erroneous collocation.

We compute three statistical features for each sentence below. (1) The first feature is computed by $\sum_{i=1}^{m} p(co_i)/n$, where m is the number of CLs, n is the number of collocations in each sentence, and probability $p(co_i)$ of each CL co_i is calculated using the method (Lü and Zhou, 2004). (2) The second feature is computed by the ratio of the number of unknown collocations (neither correct LCs nor erroneous LCs) to the number of collocations in each sentence. (3) The last feature is computed by the ratio of the number of erroneous LCs to the number of collocations in each sentence.

Perplexity from Language Model (PLM) Perplexity measures are extracted from a trigram language model trained on a general English corpus using the SRILM-SRI Language Modeling Toolkit (Stolcke, 2002). We calculate two values for each sentence: lexicalized trigram perplexity and part of speech (POS) trigram perplexity. The erroneous sentences would have higher perplexity.

Syntactic Score (SC) Some erroneous sentences often contain words and concepts that are locally correct but cannot form coherent sentences (Liu and Gildea, 2005). To measure the coherence of sentences, we use a statistical parser Toolkit (Collins, 1997) to assign each sentence a parser's score that is the related log probability of parsing. We assume that erroneous sentences with undesirable sentence structures are more likely to receive lower scores.

Function Word Density (FWD) We consider the density of function words (Corston-Oliver et al., 2001), i.e. the ratio of function words to content words. This is inspired by the work (Corston-Oliver et al., 2001) showing that function word density can be effective in distinguishing between human references and machine outputs. In this paper, we calculate the densities of seven kinds of function words [5] respectively as 7 features.

Dataset	Type	Source	Number
JC	(+)	the Japan Times newspaper and Model English Essay	16,857
	(-)	HEL (Hiroshima English Learners' Corpus) and JLE (Japanese Learners of English Corpus)	17,301
CC	(+)	the 21st Century newspaper	3,200
	(-)	CLEC (Chinese Learner Error Corpus)	3,199

Table 1: Corpora ((+): correct; (-): erroneous)

4 Experimental Evaluation

We evaluated the performance of our techniques with support vector machine (SVM) and Naive Bayesian (NB) classification models. We also compared the effectiveness of various features. In addition, we compared our technique with two other methods of checking errors, Microsoft Word03 and ALEK method (Chodorow and Leacock, 2000). Finally, we also applied our technique to evaluate the Machine Translation outputs.

4.1 Experimental Setup

Classification Models. We used two classification models, SVM[6] and NB classification model.

Data. We collected two datasets from different domains, Japanese Corpus (JC) and Chinese Corpus (CC). Table 1 gives the details of our corpora. In the learner's corpora, all of the sentences are erroneous. Note that our data does not consist of parallel pairs of sentences (one error sentence and its correction). The erroneous sentences includes grammar, sentence structure and lexical choice errors, but not spelling errors.

For each sentence, we generated five kinds of features as presented in Section 3. For a non-binary feature X, its value x is normalized by z-score, $norm(x) = \frac{x-mean(X)}{\sqrt{var(X)}}$, where `mean`(x) is the empirical mean of X and `var`(X) is the variance of X. Thus each sentence is represented by a vector.

Metrics We calculated the `precision`, `recall`, and `F-score` for correct and erroneous sentences, respectively, and also report the overall accuracy.

[5] including determiners/quantifiers, all pronouns, different pronoun types: Wh, 1^{st}, 2^{nd}, and 3^{rd} person pronouns, prepositions and adverbs, auxiliary verbs, and conjunctions.

[6] http://svmlight.joachims.org/

All the experimental results are obtained thorough 10-fold cross-validation.

4.2 Experimental Results

The Effectiveness of Various Features. The experiment is to evaluate the contribution of each feature to the classification. The results of SVM are given in Table 2. We can see that the performance of *labeled sequential patterns* (LSP) feature consistently outperforms those of all the other individual features. It also performs better even if we use all the other features together. This is because other features only provide some relatively abstract and simple linguistic information, whereas the discovered LSPs characterize significant linguistic features as discussed before. We also found that the results of NB are a little worse than those of SVM. However, all the features perform consistently on the two classification models and we can observe the same trend. Due to space limitation, we do not give results of NB.

In addition, the discovered LSPs themselves are intuitive and meaningful since they are intuitive features that can distinguish correct sentences from erroneous sentences. We discovered 6309 LSPs in JC data and 3742 LSPs in CC data. Some example LSPs discovered from erroneous sentences are *<a, NNS>* (*support*:0.39%, *confidence*:85.71%), *<to, VBD>* (*support*:0.11%, *confidence*:84.21%), and *<the, more, the, JJ>* (*support*:0.19%, *confidence*:0.93%) [7]; Similarly, we also give some example LSPs mined from correct sentences: *<NN, VBZ>* (*support*:2.29%, *confidence*:75.23%), and *<have, VBN, since>* (*support*:0.11%, *confidence*:85.71%) [8]. However, other features are abstract and it is hard to derive some intuitive knowledge from the opaque statistical values of these features.

As shown in Table 2, our technique achieves the highest accuracy, e.g. 81.75% on the Japanese dataset, when we use all the features. However, we also notice that the improvement is not very significant compared with using LSP feature individually (e.g. 79.63% on the Japanese dataset). The similar results are observed when we combined the features *PLM, SC, FWD,* and LC. This could be explained by two reasons: (1) A sentence may contain several kinds of errors. A sentence detected to be erroneous by one feature may also be detected by another feature; and (2) Various features give conflicting results. The two aspects suggest the directions of our future efforts to improve the performance of our models.

Comparing with Other Methods. It is difficult to find benchmark methods to compare with our technique because, as discussed in Section 2, existing methods often require error tagged corpora or parallel corpora, or focus on a specific type of errors. In this paper, we compare our technique with the grammar checker of Microsoft Word03 and the ALEK (Chodorow and Leacock, 2000) method used by ETS. ALEK is used to detect inappropriate usage of specific vocabulary words. Note that we do not consider spelling errors. Due to space limitation, we only report the `precision`, `recall`, `F-score` for erroneous sentences, and the overall accuracy.

As can be seen from Table 3, our method outperforms the other two methods in terms of overall accuracy, F-score, and recall, while the three methods achieve comparable precision. We realize that the grammar checker of Word is a general tool and the performance of ALEK (Chodorow and Leacock, 2000) can be improved if larger training data is used. We found that Word and ALEK usually cannot find sentence structure and lexical collocation errors, e.g., "*The more you listen to English, the easy it becomes*." contains the discovered LSP *<the, more, the, JJ>* $\rightarrow$ *Error*.

Cross-domain Results. To study the performance of our method on cross-domain data from writers of the same first-language background, we collected two datasets from Japanese writers, one is composed of 694 parallel sentences (+:347, -:347), and the other 1,671 non-parallel sentences (+:795, -:876). The two datasets are used as test data while we use JC dataset for training. Note that the test sentences come from different domains from the JC data. The results are given in the first two rows of Table 4. This experiment shows that our leaning model trained for one domain can be effectively applied to independent data in the other domains from the writes of the same first-language background, no matter whether the test data is parallel or not. We also noticed that

[7] a + plural noun; to + past tense format; the more + the + base form of adjective

[8] singular or mass noun + the 3^{rd} person singular present format; have + past participle format + since

Dataset	Feature	A	(-)F	(-)R	(-)P	(+)F	(+)R	(+)P
JC	LSP	79.63	80.65	85.56	76.29	78.49	73.79	83.85
	LC	69.55	71.72	77.87	66.47	67.02	61.36	73.82
	PLM	61.60	55.46	50.81	64.91	62	70.28	58.43
	SC	53.66	57.29	68.40	56.12	34.18	39.04	32.22
	FWD	68.01	72.82	86.37	62.95	61.14	49.94	78.82
	$LC + PLM + SC + FWD$	71.64	73.52	79.38	68.46	69.48	64.03	75.94
	$LSP + LC + PLM + SC + FWD$	81.75	81.60	81.46	81.74	81.90	82.04	81.76
CC	LSP	78.19	76.40	70.64	83.20	79.71	85.72	74.50
	LC	63.82	62.36	60.12	64.77	65.17	67.49	63.01
	PLM	55.46	64.41	80.72	53.61	40.41	30.22	61.30
	SC	50.52	62.58	87.31	50.64	13.75	14.33	13.22
	FWD	61.36	60.80	60.70	60.90	61.90	61.99	61.80
	$LC + PLM + SC + FWD$	67.69	67.62	67.51	67.77	67.74	67.87	67.64
	$LSP + LC + PLM + SC + FWD$	79.81	78.33	72.76	84.84	81.10	86.92	76.02

Table 2: The Experimental Results (A: overall accuracy; (-): erroneous sentences; (+): correct sentences; F: F-score; R: recall; P: precision)

Dataset	Model	A	(-)F	(-)R	(-)P
JC	Ours	81.39	81.25	81.24	81.28
	Word	58.87	33.67	21.03	84.73
	ALEK	54.69	20.33	11.67	78.95
CC	Ours	79.14	77.81	73.17	83.09
	Word	58.47	32.02	19.81	84.22
	ALEK	55.21	22.83	13.42	76.36

Table 3: The Comparison Results

Dataset	A	(-)F	(-)R	(-)P
JC(Train)+nonparallel(Test)	72.49	68.55	57.51	84.84
JC(Train)+parallel(Test)	71.33	69.53	65.42	74.18
JC + CC	79.98	79.72	79.24	80.23
JC(Train)+ CC(Test)	55.62	41.71	31.32	62.40
CC(Train)+ JC(Test)	57.57	23.64	16.94	39.11

Table 4: The Cross-domain Results of our Method

LSPs play dominating role in achieving the results. Due to space limitation, no details are reported.

To further see the performance of our method on data written by writers with different first-language backgrounds, we conducted two experiments. (1) We merge the JC dataset and CC dataset. The 10-fold cross-validation results on the merged dataset are given in the third row of Table 4. The results demonstrate that our models work well when the training data and test data contain sentences from different first-language backgrounds. (2) We use the JC dataset (resp. CC dataset) for training while the CC dataset (resp. JC dataset) is used as test data. As shown in the fourth (resp. fifth) row of Table 4, the results are worse than their corresponding results of Word given in Table 3. The reason is that the mistakes made by Japanese and Chinese are different, thus the learning model trained on one data does not work well on the other data. Note that our method is not designed to work in this scenario.

Application to Machine Translation Evaluation. Our learning models could be used to evaluate the MT results as an complementary measure. This is based on the assumption that if the MT results can be accurately distinguished from human references by our technique, the MT results are not natural and may contain errors as well.

The experiment was conducted using 10-fold cross validation on two LDC data, low-ranked and high-ranked data[9]. The results using SVM as classification model are given in Table 5. As expected, the classification accuracy on low-ranked data is higher than that on high-ranked data since low-ranked MT results are more different from human references than high-ranked MT results. We also found that LSPs are the most effective features. In addition, our discovered LSPs could indicate the common errors made by the MT systems and provide some suggestions for improving machine translation results.

As a summary, the mined LSPs are indeed effective for the classification models and our proposed technique is effective.

5 Conclusions and Future Work

This paper proposed a new approach to identifying erroneous/correct sentences. Empirical evaluating using diverse data demonstrated the effectiveness of

[9]One LDC data contains 14,604 low ranked (score 1-3) machine translations and the corresponding human references; the other LDC data contains 808 high ranked (score 3-5) machine translations and the corresponding human references

Data	Feature	A	(-)F	(-)R	(-)P	(+)F	(+)R	(+)P
Low-ranked data (1-3 score)	*LSP*	84.20	83.95	82.19	85.82	84.44	86.25	82.73
	LSP+LC+PLM+SC+FWD	86.60	86.84	88.96	84.83	86.35	84.27	88.56
High-ranked data (3-5 score)	*LSP*	71.74	73.01	79.56	67.59	70.23	64.47	77.40
	LSP+LC+PLM+SC+FWD	72.87	73.68	68.95	69.20	71.92	67.22	77.60

Table 5: The Results on Machine Translation Data

our techniques. Moreover, we proposed to mine LSPs as the input of classification models from a set of data containing correct and erroneous sentences. The LSPs were shown to be much more effective than the other linguistic features although the other features were also beneficial.

We will investigate the following problems in the future: (1) to make use of the discovered LSPs to provide detailed feedback for ESL learners, e.g. the errors in a sentence and suggested corrections; (2) to integrate the features effectively to achieve better results; (3) to further investigate the application of our techniques for MT evaluation.

References

Rakesh Agrawal and Ramakrishnan Srikant. 1995. Mining sequential patterns. In *ICDE*.

Emily M. Bender, Dan Flickinger, Stephan Oepen, Annemarie Walsh, and Timothy Baldwin. 2004. Arboretum: Using a precision grammar for grammmar checking in call. In *Proc. InSTIL/ICALL Symposium on Computer Assisted Learning*.

Chris Brockett, William Dolan, and Michael Gamon. 2006. Correcting esl errors using phrasal smt techniques. In *ACL*.

Peter E Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19:263–311.

Jill Burstein, Karen Kukich, Susanne Wolff, Chi Lu, Martin Chodorow, Lisa Braden-Harder, and Mary Dee Harris. 1998. Automated scoring using a hybrid feature identification technique. In *Proc. ACL*.

Martin Chodorow and Claudia Leacock. 2000. An unsupervised method for detecting grammatical errors. In *NAACL*.

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proc. ACL*.

Simon Corston-Oliver, Michael Gamon, and Chris Brockett. 2001. A machine learning approach to the automatic evaluation of machine translation. In *Proc. ACL*.

P.W. Foltz, D. Laham, and T.K. Landauer. 1999. Automated essay scoring: Application to educational technology. In *Ed-Media '99*.

Michael Gamon, Anthony Aue, and Martine Smets. 2005. Sentence-level mt evaluation without reference translations: Beyond language modeling. In *Proc. EAMT*.

Shicun Gui and Huizhong Yang. 2003. *Zhongguo Xuexizhe Yingyu Yuliaohu. (Chinese Learner English Corpus)*. Shanghai: Shanghai Waiyu Jiaoyu Chubanshe. (In Chinese).

George E. Heidorn. 2000. *Intelligent Writing Assistance*. Handbook of Natural Language Processing. Robert Dale, Hermann Moisi and Harold Somers (ed.). Marcel Dekker.

Emi Izumi, Kiyotaka Uchimoto, Toyomi Saiga, Thepchai Supnithi, and Hitoshi Isahara. 2003. Automatic error detection in the japanese learners' english spoken data. In *Proc. ACL*.

Nitin Jindal and Bing Liu. 2006. Identifying comparative sentences in text documents. In *SIGIR*.

Ding Liu and Daniel Gildea. 2005. Syntactic features for evaluation of machine translation. In *Proc. ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

Yajuan Lü and Ming Zhou. 2004. Collocation translation acquisition using monolingual corpora. In *Proc. ACL*.

Lisa N. Michaud, Kathleen F. McCoy, and Christopher A. Pennington. 2000. An intelligent tutoring system for deaf learners of written english. In *Proc. 4th International ACM Conference on Assistive Technologies*.

Ryo Nagata, Atsuo Kawai, Koichiro Morihiro, and Naoki Isu. 2006. A feedback-augmented method for detecting errors in the writing of learners of english. In *Proc. ACL*.

Jian-Yun Nie, Michel Simard, Pierre Isabelle, and Richard Durand. 1999. Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the web. In *SIGIR*, pages 74–81.

Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, and Helen Pinto. 2001. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proc. ICDE*.

Yongmei Shi and Lina Zhou. 2005. Error detection using linguistic features. In *HLT/EMNLP*.

Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *Proc. ICSLP*.

Guihua Sun, Gao Cong, Xiaohua Liu, Chin-Yew Lin, and Ming Zhou. 2007. Mining sequential patterns and tree patterns to detect erroneous sentences. In *AAAI*.

Tono Yukio, T. Kaneko, H. Isahara, T. Saiga, and E. Izumi. 2001. The standard speaking test corpus: A 1 million-word spoken corpus of japanese learners of english and its implications for l2 lexicography. In *ASIALEX: Asian Bilingualism and the Dictionary*.