UdS at the CoNLL 2013 Shared Task

Desmond Darma Putra, Lili Szabó
Saarland University
Faculty of Computational Linguistics and Phonetics
{ddputra, lilis}@coli.uni-saarland.de

Abstract

This paper describes our submission for the CoNLL 2013 Shared Task, which aims to to improve the detection and correction of the five most common grammatical error types in English text written by non-native speakers. Our system concentrates only on two of them; it employs machine learning classifiers for the *ArtOrDet*-, and a fully deterministic rule based workflow for the *SVA* error type.

1 Introduction

Grammatical error correction is not a new task in Natural Language Processing field. Many previous research was done to solve the problem. Most of these works focus on article and preposition correction.

In this paper we present our implementation of our system that participated in the CoNLL 2013 Shared Task for grammatical error correction. Out of the 28 annotated error types in the training data, this year's task focuses on 5 error types: article or determiner (*ArtOrDet*), preposition (*Prep*), noun number (*Nn*), verb form (*Vform*) and subject-verb agreement (*SVA*). This error proportion can be seen in Table 1.

From these error types we focused on *ArtOrDet* and *SVA* mistakes only.

Error type	Counts
ArtOrDet	6658
Nn	3779
Prep	2404
Vform	1453
SVA	1527

Table 1: Error types in NUCLE corpus

The remaining part of this paper is organized as follows. Chapters refcorp and 3 describe the data and system architecture. Chapter 4.2 explains the *ArtOrDet* classification task. Our experimental setup for *ArtOrDet* error is presented in Section 4.3. Chapter 4.4 describes the results from our experiments and some analysis regarding the results. Chapters 5.1 and 5.1.1 describe the task and issues respectively, Chapter 5.2 explains the how the subject-verb pairs are extracted, Chapter 5.3 is about the evaluation of the pairs. Lastly, Chapter 8 will conclude our work.

2 Corpora and Tools

The training corpus (Dahlmeier, 2013) consists of approx. 1400, 40-sentence long essays (summing up to overall 1161567 tokens), written by non-native speakers, and annotated by professional English language instructors for error tags and corrections.

The tokenized, POS-tagged and dependency and constituency parsed version of the corpus was also provided, along with the tools (tokenization - NLTK, POS-tagging and parsing - Stanford parser (Marie-Catherine de Marneffe, 2011)).

The other NLP-tools used in our implementation (described in the relevant sections) are the LIBLINEAR classifier and NodeBox.

For evaluation of the system results the M2 Scorer (Dahlmeier, 2012) was used.

3 System and Pipeline

Our system consists of two independent subsystems, which are combined serially. The parsed version of the input text first goes through the *ArtOrDet* subsystem whose output is re-parsed, and serves as the input for the *SVA* subsystem:

1. Article and determiner correction
2. Re-parsing of the data
3. Subject-verb agreement correction

In the following 2 Chapters we present the workflows for the *ArtOrDet* and *SVA* mistake types separately.

4 ArtOrDet Correction

4.1 *ArtOrDet* Mistake Type

The *ArtOrDet* error type is the most common mistake. We pose this *ArtOrDet* error correction as a multi-class classification task. The output from the classification task will be used to correct the data.

Both sentences *'girls like flowers'* and ***'the** girls like flowers'* can be accepted as correct, depending on the context - whether the noun refers to a specific group or it is a general statement. Another example like '*he ate*

Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pages 88–95,
Sofia, Bulgaria, August 8-9 2013.

the *cake*' and '*he ate* ***a*** *cake*' are also grammatically correct depending on the context whether the cake has been introduced before or not.

4.2 *ArtOrDet* Classification

An article or a determiner is followed by an NP. This article often refers to a definite or indefinite element of a class or pointing to something specific or general. There are many examples article/determiner that follows an NP, for example, *the, a, some, any, this, these, that, those,* etc. According to (Huddleston, 1984), one NP can hold up to three determiners e.g. *all her many* ideas. Moreover, each NP has a head which is noun type class. This noun consists of three subclasses including common noun (e.g. book, car, dog), proper noun (e.g. Larry, Sarah, Germany) and pronoun (e.g. you, we, they, them, it). Since we are working with *ArtOrDet* errors, then there is no point of checking NP which contains pronoun subclass because an article can never be followed by pronoun.

We classify these *ArtOrDet* errors into several types which are described in Table 2. The most common error is caused by missing *the* (around 39%). Additionally, unnecessary use of *the* contributes 26% of error. Furthermore, confusion between using *the* or *or a/an* bring 4.3% error. We classified around 15% as undefined error due to several reason. First, the error does not appear in front of the NP itself, sometimes it appears in the middle of the NP. Second, the error appears in other type phrase like adjective phrase, this makes the problem is more difficult to trace. For example, a clause "*...such invention helps to prevent elderly from falling down.*" The word *elderly* is recognized as adjective phrase and the correction happens in front of that word (adding article *the*). Third, the correction involves other articles for example *this*, *that*, and many more.

Besides the above error, there is another error which we have to handle such as confusion between *a* or *an*. This problem can be solved using a rule-based approach which will be discussed in the next section. To simplify this, we normalize article *a* and *an* into *a*. Later on, after the classification is done, we will use this rule-based to return the correct article.

4.3 Experimental setup

After defining the error types, we split the corpus into training and testing dataset. We select 50 documents from the corpus as a held-out test data and the rest is used for the training data. For the training part, we extract the NP (which is not headed by pronoun) using the information from constituent parse tree and POS tags. Each NP that is extracted represents one training example. Thus, if an NP is incorrect then we label it to one of the label from Table 2. We consider this task as a multi-class classification task, that one NP finds a mapping $f : x \rightarrow \{c_1, c_2, \ldots, c_8\}$ that maps $x \in NPs$ into one of the 8 labels.

For the first experiment, we select two well known classification methods such as LIBLINEAR (Fan et al., 2008) and Naive Bayes (McCallum and Nigam, 1998). Both of these methods are trained using the same training data and features which we are going to discuss in Subsection 4.3.5. In the testing part, our classifier will predict a label for each NP. If the classifier predicts that the observed NP is already correct or it needs to add article *a* then we apply a rule-based approach to make sure it puts the right article (*a/an*). This rule-based will utilize CMU pronouncing dictionary from NLTK to do the checking and put conditional constraints such as checking whether this NP is an acronym or not.

Classification label	Training
Correct NP	97.91%
Missing *the*	0.92%
Missing *a/an*	0.30%
Unnecessary *the*	0.07%
Unnecessary *a/an*	0.61%
Use *the* instead of *a/an*	0.03%
Use *a/an* instead of *the*	0.06%
Undefined	0.11%

Table 3: Training data

The second and third experiments are inspired by (Dahlmeier et al., 2012; Rozovskaya et al., 2012). We realize that the proportion of observed NP without article error outnumbers the observed NP with an article error (see Table 3). Therefore, this huge proportion of correct NP may affect the classifier accuracy. To justify this claim, we will utilize error inflation method for the second experiment and do re-sampling and undersampling NP as the third experiment.

4.3.1 Naive Bayes

Naive Bayes is a famous classification method which applies Bayes theorem's with naive assumptions. This assumptions believe that all features that are use to describe the data are independent (McCallum and Nigam, 1998). The advantages of this method are fast and easy to implement. This method has shown to be a good classification tool in NLP field (e.g. spam filtering, news classification, etc.). To classify an instance $D = \langle f_1, f_2, \ldots, f_n \rangle$ according to one of the classes $c_j \in C$, we calculate the maximum likelihood estimation of a prior probability c_j times the product of every features $f_{1,\ldots,n}$ given class c_j times as described below:

$$c = \arg\max_{c_j \in C} P(c_j) \prod_i P(f_i|c_j) \quad (1)$$

For this task, we utilize naive bayes package from NLTK. This method is trained using the features which are already described in Table 4.

4.3.2 LIBLINEAR

LIBLINEAR provides a large-scale classification library to handle sparse data that contains a large numbers of instances and features (Fan et al., 2008).

***ArtOrDet* errors**	**Proportion**	**Example(s)**
Missing *the*	38.9%	Working class Singaporean would be motivated to work hard as they know ***the*** government would contribute...
Missing *a/an*	12.8%	If China can come up with ***an*** effective policy to change its education system and stimulate innovation
Unnecessary *the*	26%	~~The~~ innovators, who are normally work under Research and Development department, have to recognize...
Unnecessary *a/an*	2.7%	It would no longer be able to ~~a~~ have constant economic growth which places a detrimental effect on the country
Use *the* instead of *a/an*	2.9%	The government budgets should be diverted to other areas of ~~the~~ ***a*** country's development since resources are limited
Use *a/an* instead of *the*	1.4%	As a result of ~~a~~ ***the*** growing aging population...
Undefined	15.3%	...such invention helps to prevent ***the*** elderly from falling down. Of course, ~~it~~ ***this*** is not possible. This caused problem like the appearance of slums which most of the time is not safe due to ~~the~~ ***their*** unhealthy environment

Table 2: *ArtOrDet* errors distribution from NUCLE corpus

It supports two binary linear classifiers such as L2-regularized logistic regression (LR), L1-loss and L2-loss linear SVM. Given a pair training set instance (x_i, y_i), where $i = 1, \ldots, l$, $x_i \in R^n$ and $y \in \{+1, -1\}^l$. This data will be considered as optimization problem:

$$\min_w \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \quad (2)$$
$$\text{subject to } y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i$$

where $C > 0$ as a penalty parameter.

LIBLINEAR not only supports binary class problems but also multi-class problems via one-vs-the-rest strategy. For our purpose, we will use this LIBLINEAR package with C = 0.125. This penalty value is come from the grid search which is provided in the package to find the best parameter C.

Both of these classification methods are evaluated by calculating the number of corrects prediction compare to the annotation label which is defined as:

$$Accuracy = \frac{\#\ of\ correct\ predictions}{\#\ of\ predictions} \quad (3)$$

4.3.3 Error Inflation Method

Since the *ArtOrDet* errors that we have is sparse and increase the errors proportion in the training data can help the classifier to perform better then we apply this error inflation method (Rozovskaya et al., 2012). We select some positive constant (less than 1.0) to reduce the proportion of the correct example and adding this proportion to the other error types by generating the artificial error. We found that probability among the corrections are still similar.

4.3.4 Re-sampling and Undersampling

Besides error inflation method, we are also interested in re-sampling NP with *ArtOrDet* error and undersampling without *ArtOrDet* error. Some combination will be selected to see whether it can help the classifier in detecting and correcting the *ArtOrDet* errors. we select some constant number to re-sample the NP which contains *ArtOrDet* error and some threshold to undersampling the NP which is correct. The results from these two approaches are discussed in the next section.

4.3.5 Feature Extraction

We adopt some features from (Dahlmeier et al., 2012; Rozovskaya et al., 2012) which are described in Table 4. Most of the features are coming from lexical and POS. If the NP contains an article, then we will separate it and consider as as additional feature.

wNb and wNa in Table 4 represent word at position *N* before the NP and word at position *N* after the article position. If there is no article in the beginning of NP then first word in the NP is recognize as $w1a$. pNb and pNa describe the POS of wNb and wNa. NC is a noun compound and this compound is generated by the last two words inside the NP which have noun POS. head of the NP is identified with *headWord* feature and it is determined using the information from dependency tree. NP is a noun phrase which is extracted from the constituent parse tree. $posX$ is a POS feature of X where $x \in \{NC, NP, headWord\}$. *verb* feature and *prep* are determined from the POS information. *wordAfterNP* is activated if there is another word after the NP.

4.4 Results & Discussion

The result from the first experiment can be seen in Table 6. We compare the baseline with Naive Bayes and LIBLINEAR classifier. The baseline that we choose for this task has similar definition with (Rozovskaya and Roth, 2010) which is 'do nothing'. The reason behind of this is because the proportion of NP using correct article is more than 90% and this is better than state-of-the-art classifier for article selection (with article selection, usually the baseline is set by majority class which is zero article). The result shows that LIBLINEAR produces a minor improvement than the baseline. This increase is influenced by the rule based approach that we develop to correct the use of *a* and *an*. Naive Bayes doesn't perform well due to the dependent features that

Feature Type	Description
Observed article	article
Word n-grams	w1b, w2b, w3b, w2b_w1b, w3b_w2b_w1b, w1a, w2a, w3a, w1a_w2a, w1a_w2a_w3a, w1b_w1a, w2b_w1b_w1a, w1b_w1a_w2a, w2b_w1b_w1a_w2a, w3b_w2b_w1b_w1a, w1b_w1a_w2a_w3a
POS features	p1b, p2b, p3b, p2b_p1b, p3b_p2b_p1b, p1a, p2a, p3a, p1a_p2a, p1a_p2a_p3a, p1b_p1a, p2b_p1b_p1a, p1b_p1a_p2a, p2b_p1b_p1a_p2a, p3b_p2b_p1b_p1a, p1b_p1a_p2a_p3a, p1b_w1b, p1b_w1a, p2b_w2b, p2b_w2a
NP	NC, posNC, headWord, posHeadWord, headWord_posHeadWord, w1b_posNP_posHeadWord, w1b_headWord, w1b_headWord_wordAfterNP
Verb	verb, verb_headWord, verb_NC, verb_NP, verb_posNP_headWord, verb_posNP_NC
Preposition	prep, prep_headWord, prep_NC, prep_NP, prep_posNP_headWord, prep_posNP_NC

Table 4: Features set

	1	0.9	0.8	0.7	0.6	0.5
acc.	98.64%	98.63%	98.14%	97.12%	95.10%	92.36%

Table 5: *ArtOrDet* accuracy using error inflation

Method	Accuracy
Baseline	98.5%
Naive Bayes	82 %
LIBLINEAR	98.67 %

Table 6: Classifier performance on correcting *ArtOrDet* errors

Classification label	#	Accuracy
Missing *the*	45	96%
Missing *a/an*	26	38%
Unnecessary *the*	47	100%
Unnecessary *a/an*	4	100%
Use *the* instead of *a/an*	4	0%
Use *a/an* instead of *the*	1	0%
Undefined	5	0%
TOTAL	**132**	**79%**

Table 7: Error Correction distribution

we employs.

Our second experiment tests the use of error inflation method on LIBLINEAR classifier. This test is applied to LIBLINEAR classifier with since it has a higher accuracy than Naive Bayes. The results from this experiment is described in Table 5. The smaller the constant number will result in larger article errors. Nonetheless, if we introduce too many error it will reduce the accuracy.

The last experiment test the effect of re-sampling NP with *ArtOrDet* several error times and reducing the number of observed NP that is already correct can be seen in Table **??**. The re-sampling parameter is put on the first column (5, 10, 15, 20 and 25 times) determine how many duplicates are made for each NP. On the row side we use a threshold to reduce the proportion of the observed NP which is already correct. So for each correct NP, we generate a random number and if it is higher than the threshold, then it is included in the training dataset. Table **??** reveals that re-sampling some NP that has *ArtOrDet* error does not increase the accuracy. On the other hand, reducing the threshold improve the accuracy.

If we look deeper, we found that increasing the threshold and re-sampling may have a positive correlation with correcting the error. However, the number of false positives also increased.

4.5 Further analysis

Inspired by (Gamon et al., 2008) to make two classifiers for detecting and correcting article errors. If we consider that our classifier can detect correctly the error, then we only need to train another classifier to make the correction by using the same features as described in Table 4. The training for this classifier comes from all NP with *ArtOrDet* error. Our result proves that 79% of the *ArtOrDet* can be corrected (see Table 7)

On one hand, our classifier does a good job in a sense of detecting missing article and removing unnecessary article. On the other hand, it is hard to predict either choosing between *a/an* or *the*. We found that our classifier labels this confusion as unnecessary *the* or *a/an*. This means that we have to remove the article for both of these confusions.

This may be caused by lack of training data for particular errors such as confusion between *the* & *a/an*. We realize that this mistake occurs often when the article would appear in front of an adjective - and in our feature sets there is no explicit adjective feature.

5 SVA Correction

5.1 *SVA* Mistake Type

Subject-verb agreement is the fourth most common mistake type in texts written by English language learners. It is also the highest done by machine translation systems, yet still an unsolved problem.
The English verb inflection paradigm is relatively simple, and only the misuse of third person singular and finite form of the verb (the form coinciding with the infinitive form) are of interest for the SVA error correction:

**John and Mary* goes *to work every day.*
**Mary* go *to work every day.*

Nevertheless, it is not a straightforward task, mainly because of the difficulties of linking the corresponding

subjects and verbs together. The detection of the disagreement is relatively simple, compared to the task of recognizing the number of the subject and verb.

This mistake type is different in nature form the error types (e.g. determiner and preposition) as the scope of the analysis cannot be determined as easily, therefore it has to be the whole sentence. The verb and its corresponding subject can be quite distant from one another in the sentence, and by no means have predictable positions.

In English the verbs and their subjects have no fixed positions; in indicative sentences the verb most of the times (not immediately) but follows the subject, although not necessarily, e.g. in sentences with expletives the subject follows the verb:

However, there/EXPL are/VERB still many problems/SUBJ hampering engineering design process for innovations.

5.1.1 Issues on the Syntactic Level

There are two types of syntactic phenomena that make the recognition and agreement evaluation of subject-verb difficult.

These issues are explained on dependency parsing examples, but can be generalized to any kind of grammar.

5.1.2 Multiple Subjects

When there are multiple subjects in the sentence, only the first one is labeled as a subject, the ones following it get the conj label. Even if all of them are in singular form, the verb has to be in its plural form, as multiple subjects mean plural number in English. If these type of sentences are not taken care of, that can lead to many missed corrections and to even more faulty ones. Figure 5.1.2 visualizes the problem.

5.1.3 Subject Coreference

If a sentence contains a *wh*-subordinate clause, the verb in the subordinate clause has to agree with the antecedent of the subject, but the subject is a WH-determiner (*that, what, which, who*, etc.) that can refer to both singular and plural antecedents.

The referent (ref) of the head of an NP is the relative word introducing the relative clause modifying the NP is an existing label in dependency parsing, but not available with the parser used here.

There are multiple ways to resolve the coreference, the one simplistic method[1] applied here is based on the assumption that the antecedent of the wh-subject is the closest preceding noun or pronoun to it.

Another competing method is to use the head of the verb in the subordinate clause, which is exactly the antecedent of the wh-subject (see in Figure 5.1.3). This relation is labeled as rcmod, the relative clause modifier.

When the verb is an auxiliary, its head can be a verb (*which have shaped/VBN*), an adjective (*which is effective/JJ*) or a noun (*which is a competitive funding scheme/NN*), whose head is the antecedent of the relative clause.

The second method, apart from being challenging to implement, yields to significantly worse results than the first one, most probably because of the dependency annotation mistakes in the corpus. The other problem with it is, that it requires the subjects and verbs to be paired before they the pairing is done in the pipeline.

5.2 Subject-Verb Pair Extraction

In order to being able to evaluate their agreement, the first task in finding SVA errors is identifying matching subjects and verbs. This is done in two steps:

1. extracting all predicate verbs and subjects from the sentence,
2. identifying which subject(s) belongs to which verb(s).

For recognizing inflected verb forms in **1.** the POS-tags are used; all inflected verb forms (VBZ, VBP, VBD, MD) are extracted from the sentence. As for the subjects, the dependency labels nsubj, nsubjpass, csubj are used to recognize them.

This is also the place where the multiple subject identification and coreference resolution is done. Pronoun- and determiner subjects are classified as singular or plural subjects, based on a finite list. Noun subjects are classified based on their POS-tags: NN and NNP as singular, NNS and NNPS as plural.

Once all subjects and verbs were extracted from the sentence, they have to be paired.

In **2.**, depending on how many subjects and verbs were extracted, POS templates were used to pair them.

It has to be noted here that in dependency parsing the subjects are not always dependent on the predicate verb itself, but rather on the main verb in the sentence, such as in Figure 5.2, so the head of the subject information couldn't be used.

There is no straightforward solution in the constituency parse trees either; it is not sufficient to take the head of the NP under the ROOT as the subject, as this solution wouldn't handle relative clauses properly.

5.2.1 Patterns

Only patterns, which can be almost exhaustively correctly classify subject-verb pairs are used.

Each verb is paired with the subject that is assigned an identical index. The following patterns are used:

```
Subject1  Verb1
Verb1 Subject1
Subject1/2 Verb1 Verb2
Subject1 Verb1 Subject2
Subject1 Verb1 Subject2 Verb2
Subject1 Verb1 Verb2 Subject2
```

[1] In sentences, where the wh-subject is a clausal subject, like *What engineers should do is to invent new machines.* are handled separately.

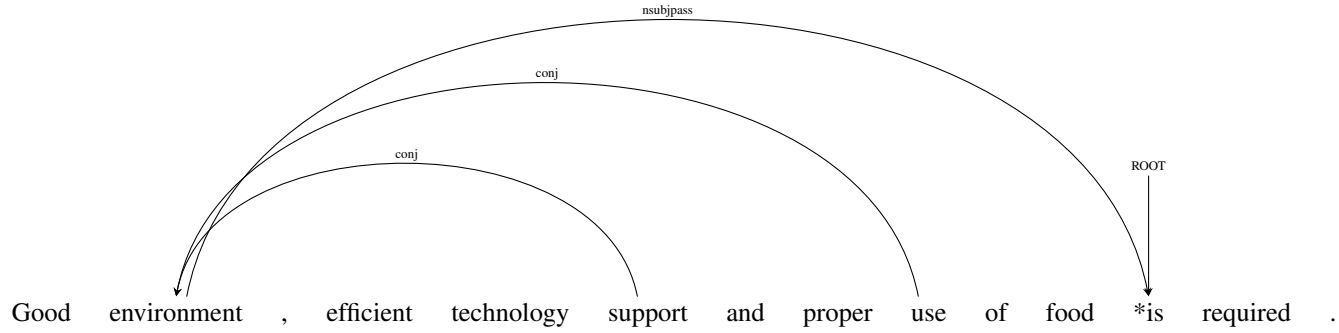

Figure 1: Dependency relations in a sentence with conjunct subjects. Only the relevant dependencies are marked. There is an original SVA mistake (made by the author) in the sentence due to the missed identification of the conjunct subjects.

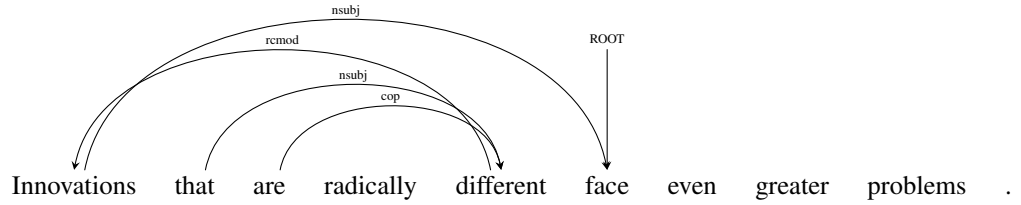

Figure 2: Sentence with subordinate clause. Only relevant dependencies are marked. The subject of the subordinate sentence is headed by the adjective, which is headed by the subject of the main clause.

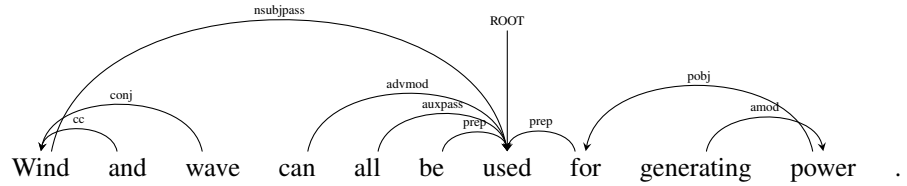

Figure 3: Sentence with labeled dependency relations. The first subject is not headed by the finite verb of the sentence *can*, but rather by the verb in the participial form *used*.

All other patterns (with 5 and more subjects or verbs in the sentence) were discarded from the evaluation, due to the far too many pairing possibilities. These long sentences generally contain a lot of modifiers, and make up 34% of the development data.

5.3 SV-Agreement Evaluation: Rule-based System

After the pairing is complete, only the pairs which include VBP[2]/VBZ[3] tags for the verbs, or verb forms in the past tense of the copula (was/were) are retained for the agreement evaluation.
If the number of the subject and verb don't agree, the verb form gets corrected.

[2]plural verb form
[3]third person singular verb form

5.3.1 Correction

The correction is done by using NodeBox, which is a tool that generates the morphologically correct singular or plural form of a given English verb.

5.4 SVA Results

On development set, only SVA-corrections, with other error types not being corrected we get a precision of 0.18.25% and a recall of 22.20%.

5.4.1 System Error Analysis

The following patterns emerged. False negatives (missed corrections) are mostly, but not exclusively due to non-accurate POS-tags, non-accurate parse trees (including many titles of the documents), dependency on other mistake types: especially on the noun number

type mistakes, mistake annotation errors and other specific cases.

6 Integrating the Systems

The systems, handling separately the mistake types, are combined in a sequential order.
The *SVA* mistake type heavily depends on the correction of the other mistake types, most prominently on the noun number (*Nn*) mistakes, as the example sentence below shows.
**This will , if not already , caused/Vform problems as there are/SVA very limited spaces/Nn for us .*
This will , if not already , cause problems as there is very limited space for us .
Although we don't deal with *Nn*-mistakes, the *SVA*-system is still the last in the row. After each iteration, the test data is re-parsed, to become the input for the next system.

7 Joint Results on Blind Data

Our final results (run on the M2 scorer) are as shown in Table 7.

Precision	0.2769
Recall	0.110
F1	0.0211

Table 8: System results on blind data

8 Conclusion

Correcting *ArtOrDet* errors for this task is not an easy job especially the number of NP using correct article is really high (more than 95%). However our LIBLINEAR classifier performance is slightly better than the baseline and Naive Bayes. Besides comparing between Naive Bayes and LIBLINEAR classifiers for this task we also adapt two approaches from (Dahlmeier et al., 2012) and (Rozovskaya et al., 2012). Our result explains that neither re-sampling method nor error inflation method contribute to the increase of accuracy.

There are several directions that can be pursued to improve the classifier accuracy. Adding language model feature which is mentioned by (Gamon et al., 2008; Dahlmeier et al., 2012) might be useful to filter the result. However using language model like Google N-gram corpus would need some extra treatment since the data is really big and need a lot of computation time to build the language model.

The hardest part of the SVA-correction task is to extract the matching subject-verb pairs; with sufficient amount of data annotated for that purpose (there is one out there, for Swedish), the rule-based approach could be turned into a statistical learning one, which might improve the recall of the system. I have found no previous research pointing to this direction. Long and complex sentences, with more than one subject-verb pairs, are frequent in corpora specific to life sciences and technology literature, such as the corpus used in this shared task. The system definitely works better on shorter sentences.

References

Dan Roth Alla Rozovskaya. 2010. Training paradigms for correcting errors in grammar and usage. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, pages 154–162.

Daniel Dahlmeier, Hwee Tou Ng, and Eric Jun Feng Ng. 2012. Nus at the hoo 2012 shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 216–224, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ng Hwee Tou Dahlmeier, Daniel. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, (NAACL 2012)*, pages 568–572, Montreal, Canada.

Ng Hwee Tou Wu Siew Mei Dahlmeier, Daniel. 2013. Building a large annotated corpus of learner english: The nus corpus of learner english. In *To appear in Proceedings of the 8th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2013)*, Atlanta, Georgia, USA.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. Hoo 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 54–62, Montréal, Canada, June. Association for Computational Linguistics.

Rachele De Felice and Stephen G. Pulman. 2007. Automatically acquiring models of preposition use. In *Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions*, SigSem '07, pages 45–50, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rachele De Felice and Stephen G. Pulman. 2008. A classifier-based approach to preposition and determiner error correction in l2 english. In *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 169–176, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xuelei Miao Yan Song Dongfeng Cai, Yonghua Hu. 2009. Dependency grammar based english subject-verb agreement evaluation. *23rd Pacific Asia Conference on Language, Information and Computation*, pages 63–71.

Mark Dredze and Koby Crammer. 2008. Confidence-weighted linear classification. In *In ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 264–271. ACM.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, June.

Michael Gamon, Jianfeng Gao, Chris Brockett, Alexandre Klementiev, William B. Dolan, Dmitriy Belenko, and Lucy Vanderwende. 2008. Using contextual speller techniques and language modeling for esl error correction. In *IJCNLP*, pages 449–456.

Na-Rae Han, Martin Chodorow, and Claudia Leacock. 2006. Detecting errors in english article usage by non-native speakers. *Nat. Lang. Eng.*, 12(2):115–129, June.

Rodney D. Huddleston. 1984. *Introduction to the grammar of English / Rodney Huddleston*. Cambridge University Press Cambridge [Cambridgeshire] ; New York.

Stephanie Seneff John Lee. 2008. Correcting misuse of verb forms. *Proceedings of ACL-08: HLT*, 12:174–182.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *ACL*, pages 423–430.

Kevin Knight and Ishwar Ch. 1994. Automated postediting of documents. In *In Proceedings of AAAI*.

Gerard Lynch, Erwan Moreau, and Carl Vogel. 2012. A naive bayes classifier for automatic correction of preposition and determiner errors in esl text. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 257–262, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christopher D. Manning Marie-Catherine de Marneffe. 2011. Stanford typed dependencies manual.

Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive bayes text classification. In *IN AAAI-98 WORKSHOP ON LEARNING FOR TEXT CATEGORIZATION*, pages 41–48. AAAI Press.

Alla Rozovskaya and Dan Roth. 2010. Training paradigms for correcting errors in grammar and usage. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 154–162, Stroudsburg, PA, USA. Association for Computational Linguistics.

Alla Rozovskaya, Mark Sammons, and Dan Roth. 2012. The ui system in the hoo 2012 shared task on error correction. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 272–280, Stroudsburg, PA, USA. Association for Computational Linguistics.