Rule-based System for Automatic Grammar Correction Using Syntactic N-grams for English Language Learning (L2)

Grigori Sidorov[†], Anubhav Gupta[‡], Martin Tozer[‡], Dolors Catala[‡], Angels Catena[‡] and Sandrine Fuentes[‡]
[†]Centro de Investigación en Computación, Instituto Politécnico Nacional (IPN), Mexico
[‡] Departament de Filologia Francesa i Romànica, Universitat Autònoma de Barcelona, Spain
www.cic.ipn.mx\~sidorov,
{anubhav.gupta, tozer.martin}@e-campus.uab.cat,
{dolors.catala, angels.catena, sandrine.fuentes}@uab.cat

Abstract

We describe the system developed for the CoNLL-2013 shared task—automatic English L2 grammar error correction. The system is based on the rule-based approach. It uses very few additional resources: a morphological analyzer and a list of 250 common uncountable nouns, along with the training data provided by the organizers. The system uses the syntactic information available in the training data: this information is represented as syntactic n-grams, i.e. n-grams extracted by following the paths in dependency trees. The system is simple and was developed in a short period of time (1 month). Since it does not employ any additional resources or any sophisticated machine learning methods, it does not achieve high scores (specifically, it has low recall) but could be considered as a baseline system for the task. On the other hand, it shows what can be obtained using a simple rule-based approach and presents a few situations where the rule-based approach can perform better than ML approach.

1 Introduction

There are two main approaches in the design of the modern linguistic experiments and the development of the natural language processing applications: rule-based and machine learning-based. In practical applications of machine learning (ML), the best results are achieved by the methods that use supervised learning, i.e., that are based on manually prepared training data for learning. It is also worth mentioning what can be considered a general rule for the combination of these two approaches: a system based on the mixed approach should obtain better results if each part of the system is applied according to its "competence". Specifically, some problems are better solved by the application of the rules—like the rules for choosing the correct allomorph of the article "a" vs. "an", while other problems are better solved by the usage of ML methods—such as deciding the presence or absence of a definite or an indefinite determiner.

This paper describes the system developed for the CoNLL-2013 shared task. The task consists of grammar correction in texts written by people learning English as a second language (L2). There are five types of errors considered in the task: noun number, subject-verb agreement, verb form, article/determiner and choice of preposition. The training data processed by the Stanford parser (de Marneffe et al., 2006) is provided. This data is part of the NUCLE corpus (Dahlmeier et al., 2013). The data also contains the error types and the corrected version.

Development of the system was started only two months before the deadline, so it is also an interesting example of what can be done in a rather short period of time and with relatively little effort: only one person-month joint effort in total.

In our system, we considered mainly the rule-based approach. Note that we used the ConLL data to extract preposition patterns, which can be considered as a very reduced form of machine learning with yes/no classifier, as well as to construct rules directly from the data.

Another feature of our system is the widespread use of the syntactic information present in the provided data. In our previous works, we generalized the use of syntactic information in NLP by introducing the concept of **syntactic n-grams**, i.e. n-grams constructed by following the dependency paths in a syntactic tree (Sidorov et al., 2012; Sidorov et al., 2013). Note that they are not n-grams of POS tags, as could be assumed from the name; the name refers to the manner in which they

Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pages 96–101,
Sofia, Bulgaria, August 8-9 2013.

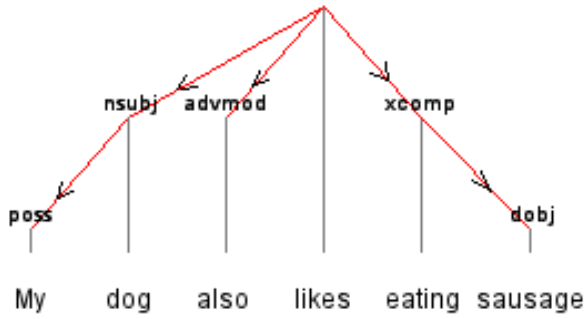

Figure 1: Example of syntactic tree (for extraction of syntactic n-grams).

are constructed. That is to say, in a dependency relation, there is always a head word and a dependent word. In the syntactic tree, this relation is graphically represented by an arrow: head → dependent. As it can be observed in Fig. 1, we can also use the tree hierarchy—the head word is always "higher" in the syntactic tree.

The algorithm for the construction of syntactic n-grams is as follows: we start from the root word and move to each dependent word following the dependency relations. At each step, the sequence of previous elements in the route taken are taken into account. The last *n* words in the sequence correspond to the syntactic n-gram. This could be reformulated as: we should take the last *n* words of the (unique) path from the root to the current word.

In other words, we start from the root and reach one of the dependent words. If we want to construct bigrams, then we have a bigram already. If we need other elements of the n-gram, then we move to the word that is dependent and continue to the words that are dependent on it. If a word has several dependent words, we consider them one after another and thus, obtain several syntactic n-grams. Note that the head word always appears before the dependent word in the syntactic n-gram during the construction process.

For example, from the tree presented in Fig. 1, the following syntactic bigrams can be extracted: *likes-also, likes-dog, dog-my, likes-eating, eating-sausage*. Note that only two syntactic 3-grams can be constructed: *likes-dog-my, likes-eating-sausage*. The construction process is the following: we start with the root word *like*. It has several dependent words: *dog, also, eating*. Considering them one after another, we obtain three syntactic bigrams. Then we move on to the word *dog*. It has only one dependent word: *my*. This is another bigram *dog-my*. However, the path from *like* also goes through it, so this is also the 3-gram *like-dog-my*, etc.

The reader can compare these syntactic n-grams with traditional n-grams and consider their advantages: there are a lot less syntactic n-grams, they are less arbitrary, they have linguistic interpretation, etc.

Note that syntactic n-grams can be formed by words (lemmas, stems), POS tags, names of dependency relations, or they can be mixed, i.e., a combination of the mentioned types. Being n-grams, they can be applied in any machine learning task where traditional n-grams are applied. However, unlike traditional n-grams, they have a clear linguistic interpretation and can be considered as an introduction of linguistic (syntactic) information into machine learning methods. Previously, we obtained better results by applying the syntactic n-grams to opinion mining and authorship attribution tasks compared to the traditional n-grams. Further in this paper, it is described how we use syntactic n-grams for the formulation of rules in our system and for the extraction of patterns.

The system described in this paper does not obtain high scores. In our opinion, it could be considered a **baseline system** for the grammar correction task due to its simplicity, its use of very few additional resources and the speed of its development. Concretely, if a more sophisticated system outperforms ours, it reflects well upon that system. If it performs more poorly, its design should be revised. On the other hand, this paper also discusses the few situations where the rule-based system can outperform an ML approach. As we mentioned earlier, the ideal system would combine both these approaches. To quote Tapanainen and Voutilainen (1994), "don't guess if you know".

Further below, we describe the lexical resources that we used, the processing of each type of error and the evaluation of the system.

2 The System's Linguistic Resources

The system consists of several program modules written in the Python programming language. We used only three types of **linguistic resources**:

- The provided corpus NUCLE data was processed with the Stanford parser. It was used for the extraction of patterns to identify

preposition errors and for the formulation of rules.

- A list of the 250 most common uncountable nouns[1]. This list was used for processing the possibility of using the nouns in plural form.

- A morphological analysis system for English that in our case was based on the FreeLing morphological dictionary (Padró et al., 2010).

The FreeLing dictionary is a freely available text file which contains more than 71,000 word forms with standard POS tags. It has the following data: for each word form, it contains a list of lemmas and POS tags. An example of the entries:
...abandon abandon VB abandon VBP
abandoned abandon VBD abandon VBN
abandoning abandon VBG
abandonment abandonment NN
abandons abandon VBZ...

This list can also be easily reordered by lemmas. It is therefore very easy to apply this word list to both morphological analysis and generation. The morphological analysis simply consists of searching for a word form in the list, while the morphological generation involves searching the list of lemmas and then finding the word form with the necessary POS tag, i.e., for the generation, the input consists in the lemma and the POS tag. For example, if we want to generate the *VBZ* form of the verb *take*, then we search in the list ordered according to the lemma *take*; there are several forms: *take took VBP, take taken VBN, take takes VBZ* and choose the form that has the POS tag *VBZ*.

3 Error Processing

In accordance with the rules of the ConLL shared task, only five types of errors were considered: noun number, incorrect preposition, choice of determiner or article, subject-verb agreement and verb form. More error types are marked in the corpus, but they are much more complex, being related to the meaning and content.

Let us see examples of the errors:

- Preposition error: "...the need of habitable environment...", where "for" should be used.

- Nn error: "...people are getting more conscious of the damages...", the word "damage" in singular should be used.

- SVA error: "...relevant information are readily available...", where "is" should be used instead.

- Vform error: "The solution can be obtain by using technology...", where "obtained" should appear.

- ArtOrDet error: "...It is also important to create a better material...", where "a" should not be used.

The total number of errors marked in the training and the test data for ConLL 2013 are presented by type in Table 1.

Table 1: Numbers of errors in training and test data listed by type.

Error type	Training	Test
Vform (Verb form)	1,451	122
SVA (Subject-verb agreement)	1,529	124
ArtOrDet	6,654	690
Nn (Noun number)	3,773	396
Prep (Preposition)	2,402	311

Note that the errors related to the noun number should be processed first since later, an agreement error could be produced if the noun number is changed. If the agreement error is introduced by the modification of the noun number, it is not the error committed by the student, however it is considered as such in the current version of the task. Probably, it can be considered as some sort of secondary error. The order in which other errors are processed is irrelevant.

3.1 Noun Number Error Processing

The only rule we implemented in this case was that uncountable nouns do not have a plural. We used a list of the 250 most common uncountable nouns (as mentioned in the Section 2) to determine the possibility of a plural form for a noun. For example: *...ethics, evidence, evolution, failure, faith, fame, fiction, flour, flu, food, freedom...*

We made an exception for the noun "time" and do not consider it as uncountable, because its use in the common expressions such as "many times"

[1]List of 250 most common uncountable nouns. www.englishclub.com>Learn English>Vocabulary>Nouns.

is much more frequent than its use as an uncountable noun as in "theory of time" or "what time is it now?". More sophisticated systems should analyze the contexts obtained from vast data sets (corpora), i.e. consider n-grams or syntactic n-grams. Note that word sense disambiguation would be helpful in the resolution of the mentioned ambiguities. Also, the rule that considers the presence of the dependent words like "many, a lot of, amount of" could be added.

3.2 Subject-Verb Agreement and Verb Form Error Processing

We consider these two types of errors together because they are related to a similar and a rather simple grammatical phenomenon. To correct these errors we used syntactic information to formulate the rules. This is logical because we cannot rely on the context words (neighbours) as they appear in texts (traditional n-grams). Note that the rules are also related to the modal verbs and the passive constructions.

The rules for the agreement are very simple: 1) if the noun is in plural and the VBZ tag is present, then change the tag to VB, 2) if the noun is in singular and the VB tag is present, then change the tag to VBZ. The corresponding morphological generation is also performed.

The rules for verb form correction are as follows: 1) if we have a modal verb, then the depending verb should have a VB tag, 2) if we have an auxiliary verb "have", then the main verb should have a VB tag (perfect tense), etc. Moreover, the FreeLing morphological dictionary is utilized to identify the correct verb form. Note that there are some assumptions here about what drives the verb form, e.g., that a noun or a modal verb are correct and the verb needs to change. This appears to be a reasonable assumption, but may not always be correct.

3.3 Preposition Error Processing

It is well-known that prepositions depend on lexical units that are their heads, see (Eeg-Olofsson and Knutsson, 2003). But what should be done if we want to consider the dependent word? Say, that in the PP attachment task, the lexical unit is the preferred solution as well. In general, it would be an ideal solution in grammar correction, but in the case of our system, very little training data was used. If we consider that the dependent word is a lexical unit, we will have less recall. We are therefore practically obliged to consider that it is a POS tag.

To process the prepositions, we used the training data provided by the organizers. Specifically, we extracted preposition patterns. We apply the concept of syntactic n-grams to include both the head word of the preposition and the dependent word into the pattern. The pattern data corresponds to syntactic n-grams because they are constructed using syntactic dependencies. As we mentioned previously, syntactic n-grams can consist of words, POS tags or a combination. In our case, we used mixed syntactic n-grams: the head word is the lexical unit, while the dependent word is the POS tag, as shown in Table 2.

For example, the first line corresponds to the erroneous phrase "...unwelcomed among public...", where "among" should be substituted by "by". Note that there can be other words between these three words in the surface representation of the sentence, but the parser allows the extraction of the syntactic n-gram, which represents the "pure" pattern.

In order to choose the syntactic n-gram type, our first consideration was that the head word should be a lexical unit (word), because this determines the choice of the preposition. We used a POS tag for the dependent element, because we considered that using a word there would be too specific. Thus, our final syntactic n-gram for the first line was "...unwelcomed among NN...", which should be changed to "...unwelcomed by NN...". The syntactic n-gram for the second line was "...trouble for NN...", which should be changed to "...trouble in NN...", etc. Note that insertion of prepositions is not considered, but deletion can be performed, i.e., changing the preposition to nothing.

The rule for the system is formulated in the following way: if we find a relation "preposition" in the dependency tree, then for the preposition that corresponds to this relation, we search the list of the extracted patterns. If we find the pattern, then we change the preposition. It is quite clear that the training data is too limited to obtain patterns for a great majority of words. Our list contained only 1,896 elements. These patterns should be extracted from a very large corpus or a dictionary.

3.4 Article or Determiner Error Processing

In this case, we found only two clear rules, both related to the article "a": 1) choice of the allo-

Table 2: Examples of patterns for prepositions.

Preposition (error)	Preposition (correction)	Head word (lemma)	Head word (POS)	Dependent word (lemma)	Dependent word (POS)
among	by	unwelcomed	VBN	public	NN
for	in	trouble	NN	development	NN
on	in	practice	NN	October	NNP
on	in	face	VBG	field	NN

morph "a/an", and 2) the fact that the article "a" cannot be used with nouns in plural. Other rules would be too complex for a manually created rule-based system. The first rule takes into the account the immediate neighbor: the choice depends on its phonetic properties. The second rule considers the syntactically related head word, which cannot be in plural if we use the indefinite article.

4 Evaluation of the System

For the evaluation, the organizers provided data similar to the training data from the same NUCLE corpus, which also contained syntactic information. The evaluation results were provided by the organizers using their evaluation script in Python (Dahlmeier and Ng, 2012). The results obtained with this script for our system are: precision 17.4 %, recall 1.8%, and F1 measure 3.3% (the preliminary scores were: 12.4%, 1.2% and 2.2% correspondingly). See the final remarks in this section, where we argue that the real values should be: precision 25%, recall 2.6%, and F1 measure 4.7%.

The results are low, but as we mentioned previously, our system uses a rule-based approach with very few additional resources, so it cannot compete with ML based approaches that additionally rely on vast lexical resources and the Internet. Due to its simplicity, low use of additional resources, and very short development time, we consider our system a possible baseline system for the task. On the other hand, we showed that in some cases the rules should be used as a complementary technique for ML learning methods: don't guess if you know.

The low recall of the system is to be expected as we process only clearly defined errors, ignoring more complex cases.

It is always interesting to perform an analysis of the errors committed by a system. Let us analyze the supposed errors committed by our system for the noun number error type. It performed 18 corrections, 3 of which coincide with the marks in the corpus data. Two of them are clear errors of the system: "traffic jam", where the word "jam" is used in a sense other than that of the "substance", and "many respects", where again the word "respect" has a different meaning to that of the uncountable noun. There are 13 cases listed below, that our system marked as errors, because they are uncountable nouns in plural, but they are not marked in the corpus. Let us consider the nouns in capital letters:
...peaceful(JJ) LIVINGS(NNS)[2]*...,*
...life(NN) QUALITIES(NNS)...,
...Many(JJ) science(NN) FICTIONS(NNS)...,
...does(VBZ) not(RB) have(VB) enough(JJ) LANDS(NNS)...,
...indicates(VBZ) that(IN) the(DT) FOODS(NNS) the(DT) people(NNS) eat(VBP)...,
...problem(NN) of(IN) public(JJ) TRANSPORTATIONS(NNS)...,
...healthcare(NN) consume(VBP) large(JJ) QUANTITIES(NNS) of(IN) energy...,
...this(DT) society(NN) may(MD) lack(VB) of(IN) LABOURS(NNS)...

Note that the words "equipment" and "usage" in plural were marked as errors in the corpus. In our opinion, it is inconsistent to mark these two as errors, and not to mark the words from this list as such. While it is true that their use in plural is possible, it is clearly forced and is much less probable. At least, students of English should learn to use these words in singular only. Some of these mistakes (but not all) were corrected by the organizers for the final scoring data. If we consider all these cases as correctly marked errors, then the precision of our system is around 25%, recall 2.6%, and F1 measure 4.7%.

[2] "LIVINGS" is encountered 5 times and "QUANTITIES" is encountered 2 times

5 Conclusions

In this paper we have described the system presented for the CoNLL-2013 shared task for grammar correction in English (L2). The system uses a rule-based approach and relies on very few additional resources: a list of 250 uncountable nouns, a morphological analyzer and the training data from the NUCLE corpus provided by the organizers. The system uses syntactic n-grams for rule formulation, i.e., n-grams that are constructed by following the dependency paths in a parsed tree.

We analyzed various situations in which a rule based technique can give better results than ML techniques: don't guess if you know. These cases are: 1) two rules for the article "a", and 2) the rules for uncountable nouns (in this case, word sense disambiguation would help to determine if the sense in the text is an uncountable noun or has some other use), and 3) the subject-verb agreement rule. In the case of prepositions, ML learning is definitely better. Otherwise, vast resources would need to be used, which in any case, would resemble machine learning. We are not sure about verb form errors: the rules which we formulated are rather simple, but the performance of various ML methods should be analysed in order to decide which technique is better.

The system is simple and was developed in a very short time. It does not obtain high scores and could be considered as a baseline system for the task.

Acknowledgements

This work was done under partial support of the Mexican Government (CONACYT, SNI, COFAA-IPN, SIP-IPN 20120418, 20121823), CONACYT-DST India ("Answer Validation through Textual Entailment"), Mexico City Government (ICYT PICCO10-120), and FP7-PEOPLE-2010- IRSES: Web Information Quality - Evaluation Initiative (WIQ-EI) European Commission project 269180.

References

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

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner English: The NUS corpus of learner English.

M.C. de Marneffe, B. MacCartney, and C.D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *LREC 2006*.

Jens Eeg-Olofsson and Ola Knutsson. 2003. Automatic grammar checking for second language learners – the use of prepositions. In *Proceedings of Nodalida'03*.

Llus Padró, Miquel Collado, Samuel Reese, Marina Lloberes, and Irene Castellón. 2010. Freeling 2.1: Five years of open-source language processing tools. In *Proceedings of 7th Language Resources and Evaluation Conference (LREC 2010)*, ELRA.

G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernandez. 2012. Syntactic dependency-based n-grams as classification features. *LNAI 7630*, pages 1–11.

G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, and L. Chanona-Hernandez. 2013. Syntactic dependency-based n-grams: More evidence of usefulness in classification. *LNCS 7816 (Proc. of CICLing)*, pages 13–24.

Pasi Tapanainen and Atro Voutilainen. 1994. Tagging accurately - don't guess if you know. In *Proceedings of ANLP '94*.