LFG-based Features for Noun Number and Article Grammatical Errors

Gábor Berend[1], **Veronika Vincze**[2], **Sina Zarriess**[3], **Richárd Farkas**[1]
[1]University of Szeged
Department of Informatics
{berendg,rfarkas}@inf.u-szeged.hu
[2]Research Group on Artificial Intelligence
Hungarian Academy of Sciences
vinczev@inf.u-szeged.hu
[3]University of Stuttgart
Institute for Natural Language Processing
zarriesa@ims.uni-stuttgart.de

Abstract

We introduce here a participating system of the CoNLL-2013 Shared Task "Grammatical Error Correction". We focused on the noun number and article error categories and constructed a supervised learning system for solving these tasks. We carried out feature engineering and we found that (among others) the f-structure of an LFG parser can provide very informative features for the machine learning system.

1 Introduction

The CoNLL-2013 Shared Task aimed at identifying and correcting grammatical errors in the NUCLE learner corpus of English (Dahlmeier et al., 2013). This task has become popular in the natural language processing (NLP) community in the last few years (Dale and Kilgariff, 2010), which manifested in the organization of shared tasks. In 2011, the task Helping Our Own (HOO 2011) was held (Dale and Kilgariff, 2011), which targeted the promotion of NLP tools and techniques in improving the textual quality of papers written by non-native speakers of English within the field of NLP. The next year, HOO 2012 (Dale et al., 2012) specifically focused on the correction of determiner and preposition errors in a collection of essays written by candidates sitting for the Cambridge ESOL First Certificate in English (FCE) examination. In 2013, the CoNLL-2013 Shared Task has continued this direction of research.

The CoNLL-2013 Shared Task is based on the NUCLE corpus, which consists of about 1,400 student essays from undergraduate university students at The National University of Singapore (Dahlmeier et al., 2013). The corpus contains over one million words and it is completely annotated with grammatical errors and corrections. Among the 28 error categories, this year's shared task focused on the automatic detection and correction of five specific error categories.

In this paper, we introduce our contribution of the CoNLL-2013 Shared Task. We propose a supervised learning-based approach. The main contribution of this work is the exploration of several feature templates for grammatical error categories. We focused on the two "nominal" error categories:

1.1 Article and Determiner Errors

This error type involved all kinds of errors which were related to determiners and articles (`ArtOrDet`). It required multiple correction strategies. On the one hand, superfluous articles or determiners should be deleted from the text. On the other hand, missing articles or determiners should be inserted and at the same time it was sometimes also necessary to replace a certain type of article or determiner to an other type. Here is an example:

> For nations like Iran and North Korea, the development of nuclear power is mainly determined by **the** political forces. → For nations like Iran and North Korea, the development of nuclear power is mainly determined by political forces.

Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task, pages 62–67,
Sofia, Bulgaria, August 8-9 2013.

1.2 Wrong Number of the Noun

The wrong number of nouns (`Nn`) meant that either a singular noun should occur in the plural form or a plural noun should occur in the singular form. A special case of such errors was that sometimes uncountable nouns were used in the plural, which is ungrammatical. The correction involved here the change of the number. Below we provide an example:

> All these measures are implemented to meet the safety expectation of the operation of nuclear power **plant**. → All these measures are implemented to meet the safety expectation of the operation of nuclear power **plants**.

2 System Description

Our approach for grammatical error detection was to construct supervised classifiers for each candidate of grammatical error locations. In general, our candidate extraction and features are based on the output of the preprocessing step provided by the organizers which contained both the POS-tag sequences and the constituency phrase structure outputs for every sentence in the training and test sets determined by Stanford libraries. We employed the Maximum Entropy based supervised classification model using the MALLET API (McCallum, 2002), which was responsible for suggesting the various corrections.

The most closely related approach to ours is probably the work of De Felice and Pulman (2008). We also employ a Maximum Entropy classifier and a syntax-motivated feature set. However, we investigate deeper linguistic features (based on the f-structure of an LFG parser).

In the following subsections we introduce our correction candidate recognition procedure and the features used for training and prediction of the machine learning classifier. We employed the same feature set for each classification task.

2.1 Candidate Locations

We used the following heuristics for the recognition of the possible locations of grammatical errors. We also describe the task of various classifiers at these candidate locations.

Article and Determiner Error category We handled the beginning of each noun phrase (NP) as a possible location for errors related to articles or determiners. The NP was checked if it started with any definite or indefinite article. If it did, we asked our three-class classifier whether to leave it unmodified, change its type (i.e. an indefinite to a definite one or vice versa) or simply delete it. However, when there was no article at all at the beginning of a noun phrase, the decision made by a different three-class classifier was whether to leave that position empty or to put a definite or indefinite article in that place.

Wrong Number of the Noun Error category Every token tagged as a noun (either in plural or singular) was taken into consideration at this subtask. We constructed two – i.e. one for the word forms originally written in plural and singular – binary classifiers whether the number (i.e. plural or singular) of the noun should be changed or left unchanged.

2.2 LFG parse-based features

We looked for the minimal governing NP for each candidate location. We reparsed this NP without context by a Lexical Functional Grammar (LFG) parser and we acquired features from its f-structure. In the following paragraph, LFG is introduced briefly while Table 1 summarizes the features extracted from the LFG parse.

Lexical Functional Grammar (LFG) (Bresnan, 2000) is a constraint-based theory of grammar. It posits two levels of representation, c(onstituent)-structure and f(unctional)-structure.

C-structure is represented by context free phrase-structure trees, and captures surface grammatical configurations. F-structures approximate basic predicate-argument and adjunct structures.

The experiments reported in this paper use the English LFG grammar constructed as part of the ParGram project (Butt et al., 2002). The grammar is implemented in XLE, a grammar development environment, which includes a very efficient LFG parser. Within the spectrum of approaches to natural language parsing, XLE can be considered a hybrid system combining a hand-crafted grammar with a number of automatic ambiguity management techniques:
(i) c-structure pruning where, based on information from statistically obtained parses, some trees are ruled out before f-structure unification (Cahill et al., 2007)

COORD	NP/PP is coordinated +/-
COORD-LEVEL	syntactic category of coordinated phrase
DEG-DIM	dimension for comparitive NPs, (”equative”/”pos”/”neg”)
DEGREE	semantic type of adjectival modifier (”positive”/”comparative”/”superlative”)
DET-TYPE	type of determiner (”def”/”indef”/”demon”)
LOCATION-TYPE	marks locative NPs
NAME-TYPE	”first_name”/”last_name”
NSYN	syntactic noun type (”common”/”proper”/”pronoun”)
PRON-TYPE	syntactic pronoun type (e.g. ”pers”, ”refl”, ”poss”)
PROPER-TYPE	type of proper noun (e.g. ”company”, ”location”, ”name”)

Table 1: Short characterization of the LFG features incorporated in our models designed to correct noun phrase-related grammatical errors

(ii) an Optimality Theory-style constraint mechanism for filtering and ranking competing analyses (Frank et al., 2001),
and (iii) a stochastic disambiguation component which is based on a log-linear probability model (Riezler et al., 2002) and works on the packed representations.

Although we use a deep, hand-crafted LFG grammar for processing the data, our approach is substantially different from other grammar-based approaches to CALL. For instance, Fortmann and Forst (2004) supplement a German LFG developed for newspaper text with so-called malrules that accept marked or ungrammatical input of some predefined types. In our work, we apply an LFG parser developed for standard texts to get a rich feature representation that can be exploited by a classifier. While malrules would certainly be useful for finding other error types, such as agreement errors, the NP- and PP-errors are often analyzed as grammatical by the parser (e.g. “the political forces” vs. “political forces”). Thus, the grammaticality of a phrase predicted by the grammar is not necessarily a good indicator for correction in our case.

2.3 Phrase-based contextual features

Besides the LFG features describing the internal structure of the minimal NP that dominates a candidate location, we defined features describing its context as well. Phrase-based contextual features searched for those minimal prepositional and noun phrases that governed a token at a certain candidate location of the sentence where a decision was about to be taken. Then features encoding the types of the phrases that preceded and succeeded a given minimal governing noun or prepositional phrase were extracted.

	Final results	Corrected output
P	0.0552	0.1260
R	0.0316	0.0292
F	0.0402	0.0474

Table 2: Overall results aggregated over the five error types

The length of those minimal governing noun and prepositional phrases as well as those of the preceding and succeeding ones were taken into account as numeric features. The motivation behind using the span size of the minimal governing and neighboring noun and prepositional phrases is that it was assumed that grammatical errors in the sentence result in unusual constituency subtree patterns that could manifest in minimal governing phrases having too long spans for instance. The relative position of the candidate position inside the smallest dominating noun and prepositional phrases was also incorporated as a feature since this information might carry some information for noun errors.

2.4 Token-based contextual features

A third group of features described the context of the candidate location at the token level. Here, two sets of binary features were introduced to mark the fact if the token was present in the four token-sized window to its left or right. Dedicated nominal features were introduced to store the word form of the token immediately preceding a decision point within a sentence and the POS-tags at the preceding and actual token positions.

Two lists were manually created which consisted of entirely uncountable nouns (e.g. `blood`) and nouns that are uncountable most of the times (e.g. `aid` or `dessert`). When generating features for those classifiers that can modify the plurality of a noun, we marked the fact in a binary feature if they were present in any of these lists. Another binary feature indicated if the actual noun to be classified could be found at an earlier point of the document.

	Only erroneous	All sentences
P	0.1260	0.1061
R	0.0292	0.0085
F	0.0474	0.0158

Table 3: Overall results aggregated over the five error types

	Only erroneous	All sentences
P	0.2500	0.0167
R	0.0006	0.0006
F	0.0012	0.0012

Table 4: Overall results aggregated over the five error types, not using the LFG parser based features

3 Results

It is important to note that our officially submitted architecture included an unintended step which meant that whenever our system predicted that at a certain point an indefinite article should be inserted or (re-)written, the indefinite article `an` was put at that place erroneously when the succeeding token started with a consonant (e.g. outputting `an serious` instead of `a serious`).

Since the output that contained this kind of error served as the basis of the official ranking we include in Table 2 the results achieved with the output affected by this unintended behavior, however, in the following we present our results in such a manner where this kind of error is eliminated from the output of our system.

Upon training our systems we followed two strategies. For the first approach we used all the sentences regardless if they had any error in them at all. However, in an alternative approach we utilized only those sentences from the training corpus that had at least one error in them from the five error categories to be dealt with in the shared task. The different results achieved on the test set according to the two approaches are detailed in Table 3. Turning off the LFG features ended up in the results detailed in Table 4.

Since our framework in its present state only aims at the correction of errors explicitly related to noun phrases, no error categories besides `ArtOrDet` and `Nn` (for more details see Sections 1.1 and 1.2, respectively) could be possibly corrected by our system. Note that these two error categories covered 66.1% of the corrections on the test set, so with our approach this was the highest possibly achievable score in recall.

In order to get a clearer picture on the effectiveness of our proposed methodology on the two error types that we aimed at, we present results focusing on those two error classes.

	`Nn`	`ArtOrDet`
P	0.4783 (44/92)	0.0151 (4/263)
R	0.1111 (44/396)	0.0058 (4/690)
F	0.1803	0.0084

Table 5: The scores achieved and the number of true positive, suggestions, real errors for the Noun Number (`Nn`) and Article and Determiner Errors (`ArtOrDet`) categories.

4 Error Analysis

In order to analyze the performance of our system in more detail, we carried out an error analysis. As our system was optimized for errors related to nouns (i.e. `Nn` and `ArtOrDet` errors), we focus on these error categories in our discussion and neglect verbal and prepositional errors.

Some errors in our system's output were due to pronouns, which are conventionally tagged as nouns (e.g. *something*), but were incorrectly put in the plural, resulting in the erroneous correction *somethings*. These errors would have been avoided by including a list of pronouns which could not be used in the plural (even if they are tagged as nouns).

Another common source of errors was that countable and uncountable uses of nouns which can have both features in different senses or metonymic usage (e.g. *coffee* as a substance is uncountable but *coffee* meaning "a cup of coffee" is countable) were hard to separate. Performance on this class of nouns could be ameliorated by applying word sense disambiguation/discrimination or a metonymy detector would also prove useful for e.g. mass nouns.

A great number of nominal errors involved cases where a singular noun occurred in the text without any article or determiner. In English, this is only grammatical in the case of uncountable nouns which occur in generic sentences, for instance:

> **Radio-frequency identification** is a technology which uses a wireless non-contact system to scan and transfer the data [...]

The above sentence offers a definition of radio-frequency identification, hence it is a generic statement and should be left as it is. In other cases, two possible strategies are available for correction. First, the noun gets an article or a determiner. The actual choice among the articles or determiners depends on the context: if the noun has been mentioned previously and thus is already known (definite) in the context, it usually gets a definite article (or a possessive determiner). If it is mentioned for the first time, it gets an indefinite article (unless it is a unique thing such as *the sun*). The difficulty of the problem lies in the fact that in order to adequately assign an article or determiner to the noun, it is not sufficient to rely only on the sentence. Thus, is also necessary to go beyond the sentence and move on the level of text or discourse, which requires natural language processing techniques that we currently lack but are highly needed. With the application of such techniques, we would have probably achieved better results but this remains now for future work.

Second, the noun could be put in the plural. This strategy is usually applied when either there are more than one of the thing mentioned or it is a generic sentence (i.e. things are discussed in general and no specific instances of things are spoken of). In this case, the detection of generic sentences/events would be helpful, which again requires deep semantic processing of the discourse and is also a possible direction for future work.

To conclude, the successful identification of noun number and article errors would require a much deeper semantic (and even pragmatic) analysis and representation of the texts in question.

5 Discussion and further work

Comparing the columns of Table 3 we can conclude that restricting the training sentences to only those which had some kind of grammatical error in them had a useful effect on the overall effectiveness of our system.

In a similar way, it can be stated based on the results in Table 4 that composing features from the output of an LFG parser is essentially beneficial for the determination of `Nn`-type errors. Table 5 reveals, however, that those features which work relatively well on the correction of `Nn` type errors are less useful on `ArtOrDet`-type errors without any modification.

As our only target at this point was to suggest error corrections related to noun phrases, our obvious future plans include the extension of our system to deal with error categories of different types. Simultaneously, we are planning to utilize large scale corpus statistics, such as the Google N-gram Corpus to build a more effective system.

Acknowledgements

This work was supported in part by the European Union and the European Social Fund through the project FuturICT.hu (grant no.: TÁMOP-4.2.2.C-11/1/KONV-2012-0013).

References

Joan Bresnan. 2000. *Lexical-Functional Syntax*. Blackwell, Oxford.

Miriam Butt, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. 2002. The Parallel Grammar Project. In *Proceedings of COLING-2002 Workshop on Grammar Engineering and Evaluation, Taipei, Taiwan*.

Aoife Cahill, John T. Maxwell III, Paul Meurer, Christian Rohrer, and Victoria Rosén. 2007. Speeding up LFG Parsing using C-Structure Pruning. In *Coling 2008: Proceedings of the workshop on Grammar Engineering Across Frameworks*, pages 33 – 40.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English. In *Proceedings of the 8th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2013)*, Atlanta, Georgia, USA. Association for Computational Linguistics.

Robert Dale and Adam Kilgariff. 2010. Helping Our Own: Text massaging for computational linguistics as a new shared task. In *Proceedings of the 6th International Natural Language Generation Conference*, pages 261–265, Dublin, Ireland.

Robert Dale and Adam Kilgariff. 2011. Helping Our Own: The HOO 2011 Pilot Shared Task. In *Proceedings of the 13th European Workshop on Natural Language Generation*, Nancy, France.

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. 2008. A Classifier-Based Approach to Preposition and Determiner Error Correction in L2 English. In *Proceedings of the 22nd International Conference on Com-*

putational Linguistics (Coling 2008), pages 169–176.

Christian Fortmann and Martin Forst. 2004. An LFG Grammar Checker for CALL. In *Proceedings of ICALL 2004*.

Anette Frank, Tracy Holloway King, Jonas Kuhn, and John T. Maxwell. 2001. Optimality Theory Style Constraint Ranking in Large-Scale LFG Grammars. In Peter Sells, editor, *Formal and Empirical Issues in Optimality Theoretic Syntax*, pages 367–397. CSLI Publications.

Andrew Kachites McCallum. 2002. MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu.

Stefan Riezler, Tracy Holloway King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques. In *Proceedings of ACL 2002*.