Semi-automated typical error annotation for learner English essays: integrating frameworks

Andrey Kutuzov
National Research University
Higher School of Economics
akutuzov@hse.ru

Elizaveta Kuzmenko
National Research University
Higher School of Economics
eakuzmenko_2@edu.hse.ru

Abstract

This paper proposes integration of three open source utilities: *brat* web annotation tool, *Freeling* suite of linguistic analyzers and *Aspell* spellchecker. We demonstrate how their combination can be used to pre-annotate texts in a learner corpus of English essays with potential errors and ease human annotators' work.

Spellchecker alerts and morphological analyzer tagging probabilities are used to detect students' possible errors of most typical sorts. F-measure for the developed pre-annotation framework with regard to human annotation is 0.57, which already makes the system a substantial help to human annotators, but at the same time leaves room for further improvement.

1 Introduction

Nowadays, learner corpora accumulating typical learner texts together with typical errors often support language learning. They allow to study interrelation of L1 and L2, and the process of language acquisition in general. Error annotation of such corpora is particularly valuable as it can provide various insights into the features of learners' interlanguage and contribute to error analysis. For example, errors made by a learner convey a lot of information about how (s)he acquires a foreign language, and which categories are most problematic (Corder, 1981). Another promising feature of error annotation is the possibility to detect L1-specific errors (Nesselhauf, 2004). Also, error-tagged corpora help human annotators and teachers who are grading students' works. All this consequently leads to more efficient language learning process.

Annotating learner texts with common linguistic annotation layers (tokens, morphology, syntax, etc) is challenging because of the non-conventional nature of such texts. It is not easy to find out what was the author's intended utterance (target hypothesis) and how it should be marked up in the corpus. Sometimes several 'readings' are possible, further complicating the situation. As for the error annotation in learner corpora, being a very complicated and a time-consuming process, it is often put aside.

Meanwhile, these two problems can be merged into one solution. Non-canonical features of learner texts can be of use when finding and correcting errors and revealing text structure. 'Strange', unconventional spelling or morphological forms provide clues about mismatches between the target hypothesis and surface form of the text (Ragheb and Dickinson, 2012). Therefore, it is possible to perform some types of error annotation automatically, disregarding its complexity.

In this paper we demonstrate our approach towards semi-automated pre-annotation of typical errors in learner English texts. We propose a solution to facilitate learner corpora error annotation based on integrating three well-known open-source frameworks, particularly, *Aspell*, *Freeling* and *brat*.

The paper is structured as follows. In Section 2 we give an overview of other approaches to automatic error annotation, and how our approach differs from them. In Section 3 we describe the tools employed in the framework, testing corpus and general work-flow. Section 4 gives details on the system performance in comparison to human-annotated texts. Section 5 points at a working prototype available online and briefly describes implementing the same tool-chain in one's own environment. Finally, in Section 6 we conclude and describe directions of further research.

2 Related work

The idea of automatic error annotation is not new. Overview of approaches to automated error detection in learner corpora can be found, for example, in (Leacock et al., 2010). In the recent years, there have been a few attempts to solve this problem, and all of them proposed unique solutions, Particularly, one should mention those deployed in the CzSL corpus (Hana et al., 2010) and in the Falko corpus (Reznicek et al., 2013).

In the CzSL corpus (the corpus of Czech as a Second Language) the work-flow of annotating errors is bound by the peculiarities of the annotation scheme. The annotation scheme consists of the two tiers or layers. The first tier includes errors dealing with the form of a word instance, so spelling and orthographic errors are defined to this tier as well as morphological errors (words with incorrect inflectional affixes). The second tier contains errors that can be derived from the context. Therefore, lexical and syntactic errors fall into this category.

As for the process of automatic error annotation, it is applied mostly to the errors from the first tier (Jelínek et al., 2012): words are compared to the dictionaries of canonical Czech, and if discrepancies are found, such word form is marked as an error. It is specific for the devised automatic annotation tools that possible morphological errors are not only manifested by tags, but the tags are further subspecified by the word part in which the possible error is found. An original word form and a word form from the dictionary are compared symbol by symbol, and if alternations are found in the inflectional part of the word, this counts as a morphological error; if the word form contains mistake in its stem part, it is considered to be a made-up word (Rosen et al., 2014).

This automatic annotation system is used not only to extend the manual annotation, but also to verify it. If the system finds some words that are unknown to the morphological analyzer but are unmarked with tags, the errors was possibly overlooked by a human annotator. If the changes proposed by the system concern pronunciation, the presence of the tag denoting inflection or word base is checked.

All texts in CzSL are also pre-processed with *Korektor* spell-checker (Hana et al., 2014). It is applied to both original and corrected versions of the text.

This automatic spell-checking is similar in part to what we do in this research. However, we additionally introduce automatic error-tagging using morpho-syntactic tags (see Section 3)

Errors from the second tier are annotated manually in CzSL; however, some information is added to them automatically, based on the context of the error, or, in case of an error in a compound verb form, on the morphological analyses assigned to the word. It happens only when a human annotator has already initially marked the errors.

Our approach is different in two ways. First, our framework detects not only errors from the first tier, but also the errors from the second tier (e.g., agreement errors), which are annotated manually in CzSL. The mismatch in the context of word form in the case of disagreement reflected in morphological analysis allows us to detect more error types than by using only spellchecker. Second, we do not distinguish between different types of spelling errors. As English is not a highly inflective language like Czech, spelling errors convey less information about their nature; most often it means that the word detected by a spellchecker simply does not exist.

The Falko corpus (Reznicek et al., 2013) performs error-annotation using the mismatch between target hypothesis (speaker's intention) and the actual learner's text. For example, in the sentence '*The girl sing loudly.*' the target hypothesis formulated by a sequence of queries into a corpus of native speakers' texts states that such noun phrase should be accompanied by a verb with the *-s* ending, and there are no cases when such combination of word forms is met in the native language. Nevertheless, if this form is found in the learner's text, this span is marked as an error.

This approach is partly similar to a component of our framework, the one which is based on morphological analysis. As we will demonstrate in Section 3, we derive the target hypothesis from the PoS tags probabilities, and not from a corpus of canonical English, but the nature of the approach stays the same.

3 Mixing tools and the corpus

To construct our framework, we used three tools: an annotation framework, a set of linguistic analyzers and a spellchecker.

Brat (Stenetorp et al., 2012) is an open-source framework for web-based text annotation. It sep-

arates documents from their markup (see below), and allows several people to annotate a text simultaneously, using only their web browsers. It also provides an important possibility to easily define new annotation schemes. In this paper, it serves as a basis for all other tools.

Freeling (Padro and Stanilovsky, 2012) is a set of open source linguistic analyzers for several languages. It features tokenizing, sentence splitting, morphology analyzers with disambiguation, syntax parsing, named entity recognition, etc. In this research, we use only morphological analyzer for English.

Finally, *GNU Aspell*[1], currently maintained by Kevin Atkinson, is one of the most popular open source spelling correction utilities. It compares an input word to a set of dictionaries and if the word is out-of-vocabulary (possible typo), provides a list of words similar in spelling.

The tools are tested on REALEC, Russian Error Annotated Learner Corpus[2]. REALEC is a corpus of Russian students' essays written in English (Kuzmenko and Kutuzov, 2014). The works in the corpus are written by 2, 3 and 4 year students from National Research University Higher School of Economics, Faculty of Philology, together with students of the first year of Masters program, Faculty of Psychology. The texts are mostly routine assignments or exam-type essays. Most of the works are written with the premises to prepare for the IELTS examination and have the structure similar to that of IELTS writing tasks (Moore and Morton, 2005). Essays in this corpus are manually error-annotated in *brat* by human experts (mostly English teachers). They output a substantial amount of quality annotation, but the process of error spotting is rather cumbersome and time-consuming. Thus, there is a certain need to at least semi-automatize this annotation task and make computers do the most monotonous part of the work.

The work flow we propose is as follows. When a document (an essay) is uploaded to the system, it is processed by *Freeling*. Processing includes tokenizing, sentence splitting and morphological analysis (lemmatizing and PoS-tagging).

Then, we detect possible errors. First, all tokens and lemmas generated by *Freeling* are checked with *Aspell*. If neither token nor lemma are known English words, we assign this token an attribute '*Possible spelling error or typo*', which is visible and searchable in the annotators' web interface. We also add a note to this token with the first correction suggested by *Aspell*. Thus, L2 (English in this case) spelling rules are the basis for this annotation.

It is important that by design *Aspell* does not make any difference between non-words or unknown neologisms and typos (misspelled words). This sometimes may lead to false flags: for example, the word '*polysemy*' is out of vocabulary and marked as a spelling error, with '*polysemous*' suggested as a correction. We plan to deal with this issue in the future, most probably using evaluation of Damerau-Levenshtein distance (Damerau, 1964) between words and suggestions.

After annotating spelling errors, we move on to the Part-of-Speech (PoS) tags for all tokens.

In the course of morphological analysis, *Freeling* outputs probabilities of different PoS tags for each token, depending on its lexical environment. For example, in the sentence

'*He plays with his phone.*'

Freeling assigns the token '*plays*' the PoS tag **VBZ** (Verb, 3rd person singular present) with probability as high as 0.663934. However, if we introduce an error in the same sentence and transform it into

'*He **play** with his phone.*',

the token '*play*' is assigned the **VBP** tag (Verb, non-3rd person singular present) with the probability as low as 0.163539.

The reason of such a low value is that other tagging variants for this word form are much more probable. We can get all the possible morphological 'readings' of the given word with their default probabilities in the model. Continuing our example with '*play*', *Freeling* had to choose from three variants (given with their respective probabilities):

1. **VB 0.565684** (Verb, base form)
2. **NN 0.270777** (Noun, singular)
3. **VBP 0.163539** (Verb, non-3rd person singular present)

Most probable tag for '*play*' is an infinite verb form. However, a variant with low default probability was chosen because of the context (preceding '*He*'), thus signaling that something erroneous may be happening here. Naturally, in the case of

[1] http://aspell.net/

[2] http://realec.org

the correct sentence, the PoS tag **VBZ** for the word '*plays*' has the maximum default probability:

1. **VBZ 0.663934** (Verb, 3rd person singular present)

2. **NNS 0.336066** (Noun, plural)

This information gives some clues as to which words manifest possible errors. Particularly, we check whether there are other possible tagging variants with default probability greater than the probability of the variant *Freeling* actually chose. If it is true, we suppose that *Freeling* met difficulties in choosing between tag variants, and there can be a mismatch between word surface form and its distributional features (lexical environment). In this case we assign an attribute '*Possible grammar or morphology error*' to this token. As such tokens can be highly ambiguous with regard to their tagging variants, a note with other tags (rejected by *Freeling*) is added to the token annotation.

Of course, this issue is not tackled with 100% precision, and low default probability of the chosen tag variant does not always mean that there is an error in the sentence. However, as we show below, in most cases this is a good indicator of inconsistencies in the word sequence, and this can help an annotator a lot. Some proportion of mistakes is necessarily acceptable, and the output will afterwards be checked by a human, so that incorrectly flagged instances will be removed from the annotation.

After having conducted the pre-annotation of errors, the output of *Freeling* and *Aspell* is converted to the standard CONLL format and then to the *brat* standoff annotation format. At this stage text and annotations are separated (consistent with the data structure adopted in Falko). The only change in the text is introduced by tokenization, which extracts all punctuation marks and surrounds them with spaces, so that they can be considered full-fledged tokens. All annotations are kept in a separate annotation file for each document, linked to the actual text by character offsets.

Surprisingly, the shallow analysis described above returns quite satisfactory results with regards to recall and the number of false flags; see Section 4 for evaluation of our technique.

As a result, human expert receives a document which is not only tokenized and POS-tagged, but also pre-annotated with possible errors. The errors caught by this method are mostly limited to misspellings, typos and morpho-syntactic ones. Nevertheless, these types constitute a substantial share of errors in a real learner corpus.

Consequently, our system allows annotators to spend less time on spotting spans to pay attention to, and additionally lessens the risk of overlooking errors. The latter turned out to be particularly useful, as human annotators tend to miss the spelling errors in which some letter doubles or, on the contrary, double lettering is absent. For example, errors like '*signalling*' (gerund form), or '*posess*' were overlooked in human annotation, but found by the framework.

Also, paradoxically, automatic error annotation helps to detect errors which are not spotted by humans because of the transfer effect from their L1. For instance, Russian learners of English often make an error concerning the verb *consider* control pattern. Many learners generate erroneous *consider smth as smth*, which comes from the analogous structure in Russian, but is ungrammatical for English. Human annotators tend to omit this error, but it is always found by the framework.

4 Evaluation

Our pre-annotation was tested against errors spotted by human annotators in 800 documents from REALEC corpus (213 694 word tokens in total). After applying the framework, we encountered 10490 morphological errors 'issued' by *Freeling* and 3018 spelling errors by *Aspell*. This is consistent with the ratio of spelling mistakes in human annotations of the same texts (Kuzmenko and Kutuzov, 2014).

Initially, we checked strict coincidences of automatically detected 'pre-errors' with human-annotated error spans, so that only the tokens from our pre-annotation that exactly match those assigned by humans were counted. Quite expected, performance was not very impressive, with F-measure only 0.05 (see Table 1).

The reason for such low values is that human experts often mark spans ranging across several words or even parts of words. In fact, tagging several words is necessary for particular types of errors, for example, word order errors. At the same time, our system annotates only separate words, and thus lags behind humans.

The figures for *Aspell* and *Freeling* parts of the framework separately were discouraging as well.

While the *Freeling* component in general performs slightly better that the *Aspell* component, both tools demonstrate low recall and even more discouraging precision.

However, in fact we do not need precise hits into human annotated spans. What we expect is that pre-annotation will help an expert or a language teacher in spotting problematic areas in the text, and then they will be properly annotated.

Hence, we measured how good our system is at hitting right sentences, that is, generating errors at the same sentences where human experts found various mistakes.

First, we carried out evaluation of our system with regard to a simple baseline, within which we assigned an error mark to every sentence in the corpus with the probability of 50%. This alone resulted in increased performance, with F-measure 0.123 (see Table 1).

When we applied the real *Freeling* and *Aspell* output, we received results seriously outperforming the baseline, with precision and recall at values allowing real-world usage (0.46 and 0.75 respectively).

Table 1: Performance in comparison with human judgments

	Precision	Recall	F-measure
Strict matches			
Overall	0.04	0.07	0.05
Aspell only	0.007	0.04	0.01
Freeling only	0.046	0.06	0.05
Sentence-wise matches			
Baseline	0.0973	0.169	0.123
Overall	0.4637	**0.7479**	0.57
Freeling only	**0.7643**	0.5383	**0.63**

This is already a decent result as precision is relatively high, therefore, most of the errors spotted by the system are flagged correctly, and an annotator only needs to define a proper error type for them.

It can be seen that the integration of *Aspell* slightly spoils the precision figures. *Freeling* method without spell-checking provides better precision and F-measure. This is due to the fact that *Aspell* assigns erroneous tags to the instances, being driven by the wide definition of an error as a word form absent in its dictionary. At the same time, *Aspell* helps achieving very high recall values.

It should be mentioned that *Korektor* spellchecking system for Czech is reported in (Hana et al., 2014) to perform with an accuracy of 74%. It is difficult to compare performance of spellcheckers for English and Czech. However, increasing the performance of our spellchecker part should definitely be an important step towards enhancing our framework in general.

Nevertheless, recall has increased, and almost 75% of sentences containing errors are already flagged even before experts take to their job; this reduces human efforts. The overall precision is lower, meaning that about a half of flags are false: we pre-annotate an error within a sentence, where according to human experts there are no errors.

For example, in the sentence

'*Since that period modern human started to tame animals and use them for the good of primitive society.*'

our framework finds three errors: in the words *that*, *tame*, and *use*. Meanwhile, only one error was identified by manual annotation: erroneous choice of a lexeme *started* in this context. For now, we do not set up a goal to identify lexical errors, but the annotation of redundant tokens clearly is a disadvantage for an annotator.

There are also positive examples. In the sentence

'*Hen was spread worldwide by humans, and that's why domestication was useful for these species.*'

the number of errors found by our system and by human annotators equals to four in both cases, and in two cases (the words *these* and *spread*) pre-annotation and manual annotation coincide (Actually *was spread* and *these species* are annotated by humans, but the problematic area is identified correctly).

We plan to improve precision in future research. For now, this issue is mitigated by the fact that in the case of incorrect pre-annotation, an expert can easily change or ignore it. We consider precision at the value of 0.5 to be acceptable for the time being.

5 Implementation

Our implementation of the described system can be found at `http://dev.rus-ltc.org/learner_preprocess/index.xhtml#/integration/`. It is possible to browse through

a sample of REALEC texts with possible errors marked by red. After logging in with the user name '*learner*' and identical password, one can upload own texts. They will be tagged and annotated with possible mistakes.

Deploying this framework on one's own server is as easy as installing *brat* and *Freeling* and slightly fixing *brat* document work-flow to include pre-processing stage. *GNU Aspell* is usually already installed on any Unix/Linux system. All the source code for our converters and detectors together with instructions is available online at Github[3].

6 Conclusions

We presented a framework integrating morphological analyzer, spellchecker and web annotation tool in order to pre-annotate learner English texts with possible errors. While already providing a significant relief to human experts, with F-measure 0.57 in relation to human annotations, it is yet to be extended and improved.

It is important that unlike other automatic error-tagging systems (for example, in (Hana et al., 2010)), our framework functions without any knowledge about target hypotheses or correct forms of words in the analyzed texts. Its input is raw learner-generated sentences and it does its work before any human intervention. Additionally, the errors we detect are not limited to incorrect word forms, but also include error classes related to complex syntactic patterns.

One of prospective directions for improving our system performance is to differentiate between a larger number of error types, for example, taking into account syntax trees constructed by *Freeling* parser module, finding non-typical dependencies. Supposedly, this can help in spotting errors on supra-lexical levels.

Tracking lexical errors can be done comparing neighbors of a given unit in canonical English language corpora and our learner corpus. Also, spell-checking part can be augmented with additional dictionaries, especially containing gazetteers of named entities, in order to prevent it from incorrectly marking proper names as typos.

Also, we plan to investigate the relationship between the language level of learners' and the features of their mistake from the perspective of automatic annotation. It is expected that the architecture of the automatic annotation system is heavily dependent on the linguistic characteristics of texts. For example, in the beginners' level it is possible that more mistakes concerning morphology and syntax are found, whereas advanced learners make more lexical mistakes. Therefore, we plan to adapt different algorithms and approaches towards automatic error annotation to different levels on language knowledge.

Another improvement that is needed to be done in future is to test human reaction on the errors spotted automatically. For now, our system was not deeply tested and checked with English language teachers, and we need to measure to what extent such pre-annotation facilitates human efforts and how many errors spoil the process of correct error annotation.

[3] `https://github.com/akutuzov/error_annotation`

References

Stephen Pit Corder. 1981. *Error analysis and interlanguage*, volume 112. Oxford Univ Press.

Fred J. Damerau. 1964. A technique for computer detection and correction of spelling errors. *Commun. ACM*, 7(3):171–176, March.

Jirka Hana, Alexandr Rosen, Svatava Škodová, and Barbora Štindlová. 2010. Error-tagged learner corpus of czech. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 11–19. Association for Computational Linguistics.

Jirka Hana, Alexandr Rosen, Barbora Štindlová, and Jan Štěpánek. 2014. Building a learner corpus. *Language Resources and Evaluation*, 48(4):741–752.

Tomáš Jelínek, Barbora Štindlová, Alexandr Rosen, and Jirka Hana. 2012. Combining manual and automatic annotation of a learner corpus. In *Text, Speech and Dialogue*, pages 127–134. Springer.

Elizaveta Kuzmenko and Andrey Kutuzov. 2014. Russian error-annotated learner english corpus: a tool for computer-assisted language learning. *NEALT Proceedings Series Vol. 22*, page 87.

Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault. 2010. Automated grammatical error detection for language learners. *Synthesis lectures on human language technologies*, 3(1):1–134.

Tim Moore and Janne Morton. 2005. Dimensions of difference: a comparison of university writing and ielts writing. *Journal of English for Academic Purposes*, 4(1):43–66.

Nadja Nesselhauf. 2004. Learner corpora and their potential for language teaching. *How to use corpora in language teaching*, 12:125–156.

Llus Padro and Evgeny Stanilovsky. 2012. Freeling 3.0: Towards wider multilinguality. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uur Doan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

Marwa Ragheb and Markus Dickinson. 2012. Defining syntax for learner language annotation. In *Proceedings of COLING 2012: Posters*, pages 965–974, Mumbai, India, December.

Marc Reznicek, Anke Lüdeling, and Hagen Hirschmann. 2013. Competing target hypotheses in the falko corpus. *Automatic Treatment and Analysis of Learner Corpus Data*, 59.

Alexandr Rosen, Jirka Hana, Barbora Štindlová, and Anna Feldman. 2014. Evaluating and automating the annotation of a learner corpus. *Language Resources and Evaluation*, 48(1):65–92.

Pontus Stenetorp, Sampo Pyysalo, Goran Topic, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for nlp-assisted text annotation. In *EACL*, pages 102–107.