Towards Transformation-based Annotation of Norm Deviations in an Infrastructure for Research on Swedish as a Second Language

Dan Rosén[1], Mats Wirén[2], Elena Volodina[1]
[1]Språkbanken, University of Gothenburg, [2]Stockholm University
[1]Box 200, 40530 Göteborg, [2]SE-10691 Stockholm
dan.rosen@svenska.gu.se, mats.wiren@ling.su.se, elena.volodina@svenska.gu.se

Abstract

This paper describes ongoing work on a tool for annotation of norm deviations (errors) in second-language learner texts, a key component in an intended infrastructure for research on Swedish as a second language. Unlike traditional approaches which treat this as a single task applied to a static learner text, our approach is divided into two steps to reflect the conceptual structure of the problem: a) normalisation of the learner text by transforming (editing) it to reflect the target hypotheses, and b) the actual norm-deviation annotation, supported by visualisation of the differences between source and normalisation. Another distinct feature of our approach is that a parallel text is generated in the transformation step, with word alignments inferred from the editing operations. This parallel text is a useful resource in its own right, by allowing for search in either or both of the source and normalised texts, and for training of systems for automated annotation of norm deviations. We describe the normalisation component of our tool and outline the associated component for annotation of norm deviations currently being implemented.

Keywords: second language learner corpora, error annotation, normalisation

1. Introduction

The need for data is omnipresent in language-technology projects, not the least when it comes to data produced by second-language learners. Automatic analysis of learner data is very challenging, however, since part-of-speech taggers and syntactic parsers are almost always trained on the standard language. Annotation of norm deviations (that is, errors[1]) is therefore predominantly a manual process. Like all manual annotation, it is very time-consuming, but the problem is exacerbated by the high degree of subjectivity in deciding what the deviation is and of what the correct form would then be.

This paper describes our tool which aims at a more efficient handling of normalisation and annotation of norm deviations, thereby alleviating some of the data bottleneck. The work is carried out in the context of a project for constructing an infrastructure for research on Swedish as a second language (SweLL[2]), in which our tool will constitute a key component.

Annotation of norm deviations in second-language learner texts is typically done with a purpose-built editing tool, using a hierarchical set of error categories (Granger, 2008). For example, in ASK (Tenfjord et al., 2006), the editor used is the XML editor Oxygen, customised with pop-up menus that reflect the chosen set of error categories. Specifically, annotation of an error involves inserting an XML *sic* tag with a *type* attribute encoding the error category, a *desc* attribute encoding the error subcategory, and a *corr* attribute encoding the correct (normalised) form of the token or tokens. By running a set of XSLT scripts, the XML file can be translated to an HTML format which can be viewed in a web browser, thereby facilitating proofreading of the annotation.

The basic problem with this kind of approach is that it conflates the conceptually separate stages of (traditionally speaking) error coding, namely, error detection, correction and annotation (Ellis, 1994; Granger, 2008, page 266). This is because correction (normalisation, based on a target hypothesis) is carried out as a sub task of error annotation instead of as an independent task, prior to and separate from error annotation. Also, since normalisations are only a by-product of error annotations, the corrected text as a whole is not readily available for inspection after each annotation step, and as a result it may be more difficult to produce a consistent normalisation.

In Falko (Lüdeling et al., 2005) and MERLIN (Boyd et al., 2014), target hypothesis corrections have a more independent status in the sense that they are entered in a separate step prior to error annotation, but the normalised text still does not seem to be available in its entirety.

To circumvent these problems, and to reduce subjectivity and increase inter-annotator agreement, our approach is based on dividing error coding into two steps: normalising the text by transforming (editing) it into a corrected form according to some given criteria, and annotating the norm deviations corresponding to the changes. Furthermore, as a result of this process, a parallel corpus consisting of the learner source text and the normalised text is incrementally constructed. Thus, our contribution is a system for transformation-based annotation of norm deviations which includes the following two components:

- An editor in which a learner source text can be normalised and a parallel corpus of the two texts is simultaneously generated (Sections 2. and 3.).
- Integrated with this, a facility for annotating the normalisations (changes) with norm-deviation categories. (Section 4.)

[1]We use the term *norm deviation* rather than "error" as a neutral term with respect to the idiosyncrasies of the interlanguage of the learner (Selinker, 1972). Similarly, we refer to the changes imposed on a learner text to reflect the norms of the target language as *normalisation*.

[2]https://spraakbanken.gu.se/eng/swell_infra

2. Target Hypotheses

As pointed out above, annotation of a norm deviation always involves an interpretation of what the deviation is – the target hypothesis, here realised by editing of the text. A complication is that there may be multiple competing target hypotheses for a deviation. An example of this from Lüdeling et al. (2005, Section 2.1) is "die Erklärung für diese Phänomen ist einfach", where "diese Phänomen" in absence of other information could equally likely be interpreted as a gender error (with the target hypothesis "dieses Phänomen") or as a number error (with the target hypothesis "diese Phänomene"). Although our tool only handles one target hypothesis, in principle we could deal with multiple hypotheses by maintaining different normalisations for a given part of the source text.

Furthermore, some systems allow for several levels of target hypotheses. In the Czech learner corpus of Hana et al. (2012) a first level normalises grammatical forms without context information. All other deviations (e.g. word order) belong to a second level and take the whole sentence into consideration. In MERLIN (Boyd et al., 2014) two levels are used: a) minimal corrections pertaining only to orthographic and grammatical errors and b) corrections to handle sociolinguistic, lexical and pragmatic deviations to arrive at what is referred to as an acceptable text. Again, our tool does not handle this, but it would be possible to do so by maintaining several levels of aligned normalisations. The first level (with basic corrections) would then be aligned to the source text, and the second level (with additional corrections) would be aligned to the first level.

3. The Normalisation Tool

We wanted our tool to have the feel of a conventional text editor and yet retain information about how a word was normalised or where it was moved to. None of the existing tools corresponded to our expectations, hence we set off to build our own normalisation tool.

Our first key observation was to view the construction of a target hypothesis as creating a parallel corpus where one of the texts is the the source learner text, and the words in this text is linked to the other text: the target hypothesis of the normalisation. However, we realised that these links can be maintained automatically for the annotator while they edit the source text into the target text in text editing area if proper logging of the editing operations are put into place.

During normalisation, the user will need to pay attention to how their edits affect the links between the source text and the target hypothesis. We facilitate this by providing views of their difference and the links between them: we use four panes (see Figure 1): one containing the frozen source text, one in which the editing takes place, one in which changes are highlighted, and finally one which shows links between tokens in the source text and normalised text. To normalise a source text using our tool the annotator only uses the text area in the middle of Figure 1. It works just like a text field in a normal text editor.

Each edit the user does in this text area is reflected in the underlying parallel structure. This is shown in two ways: 1) as an inline diff, the rightmost pane of Figure 1, 2) pictorially, as the graph below it, which shows the links as edges from the words in the source text (on the top row) to the words in the target text (the bottom row).

We will use an example sentence in English to illustrate what editing operations our tool supports and how to execute them: "*Examples high light here lotsof futures*". We demonstrate our tool by normalising this using our tool. Initially, the source and target hypothesis text are the same and each word is connected to itself and the diagram looks as Figure 2a. Now we start editing, addressing these deviations:

- *Deviant spelling*: this is normalised by editing the text. In Figure 2b the user can put the cursor in "*futures*", delete the *u* character with backspace and type in *ea* to get *features*.

  We show differences inside tokens by calculating a diff automatically using standard techniques for finding edit scripts. Deletions are shown in red with strikethrough, and insertions in green. This intra-token diff is recalculated after every edit (internally only replacements on the token level are stored.) This highlighting is simply an aid for the user.

- *Over splitting*: normalised by removing spaces. The user removes the space in *high light*, obtaining the state in Figure 2c.

- *Over compounding*: conversely, this is normalised by inserting a space inside a word. This is how the word *lotsof* has been split into two in Figure 2d.

- *Word order*: normalised by drag and drop using the mouse. An alternative is to use the standard keyboard shortcuts for cut and paste. The word in the source text remain connected to the target word. Using either of these methods, the user has moved one word to the end of the sentence in Figure 2e. Note that it is the complexity and ambiguities from word order changes are the reason why we log the editing operations.

- *Missing word*: normalised by inserting the missing word. The new target word is not connected to any source word. See the insertion of *to* in Figure 2f.

- *Redundant word*: normalise by deleting the word. The source word is now not connected to anything. In Figure 3 we exemplify with an alternative analysis to Figure 2e where the word *here* has been removed.

Implementation We have implemented a working prototype as a single-page web application. It does not rely on any server backend: all processing is done in the client's browser. The editor has support for unlimited chronological undo and redo operations. For convenience we also support undoing the normalisation at a specific position. Undoing is very useful to get links exactly right, which occasionally takes a few tries. The editor may also be used as a library and can in this way be included in other tools and webpages. The implementation is free open source software released under the MIT license[3], and may be tried online[4].

[3] https://github.com/spraakbanken/swell-editor

[4] https://demo.spraakdata.gu.se/dan/swell-editor

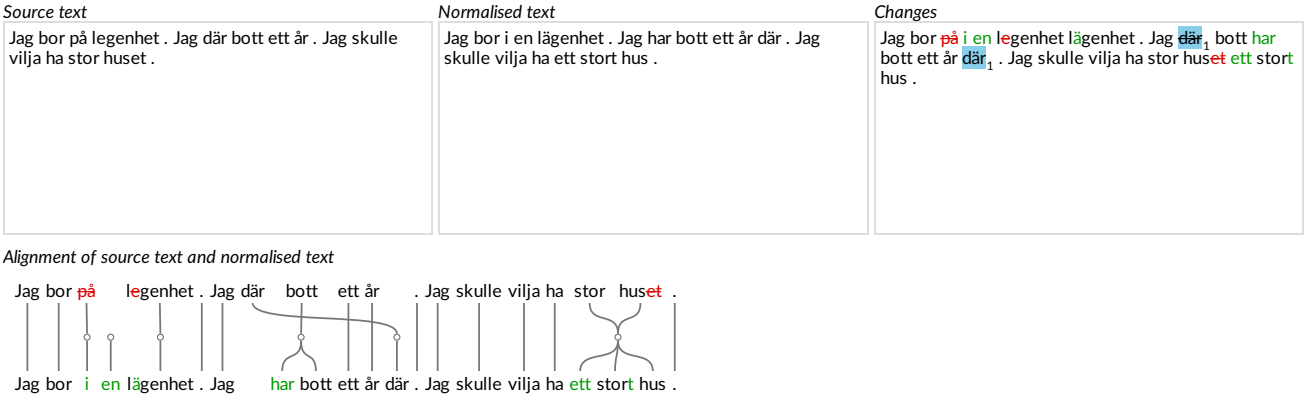
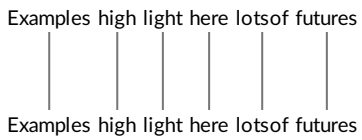
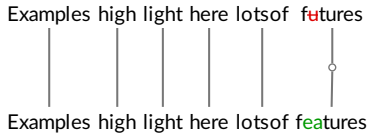


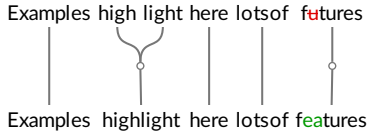
Figure 1: Screenshot of an editing session in progress. The three panes in the upper row are from left to right: 1) the source text which cannot be edited, 2) the target hypothesis which is where the annotators do all their edits, 3) a calculated view of the differences between the two texts obtained from the history of edits. Below the panes we display the the differences pictorially: the source text on top and the target hypothesis below, with edges showing how words have been moved.



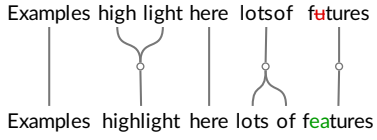
(a) The initial state.



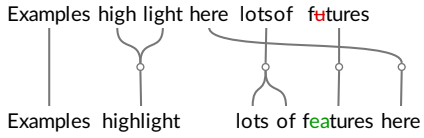
(b) Editing within a word normalises spelling.



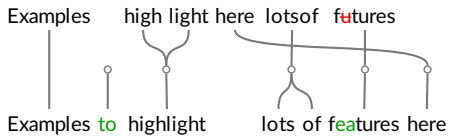
(c) Removing white space normalises over splitting.



(d) Adding space within a word normalises over compounding.



(e) Drag and drop (or cut and paste) normalises word order.



(f) New words can be inserted to normalise missing word.

Figure 2: Editing operations are reflected in the links between the learner text and the annotator's normalised text.

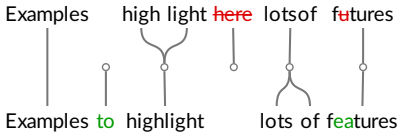


Figure 3: Alternative analysis to redundant word normalisation.

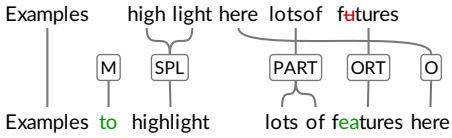


Figure 4: Example annotated sentence using ASK taxonomy. From left to right: missing word, over splitting, over compounding, orthographic deviation and word order.

4. Annotation of Norm Deviations

The norm-deviation annotation is carried out by selecting material, either one or several tokens that have been changed in *Normalised text*, or the corresponding component of joint source and normalisation in *Alignment* in Figure 1. When the annotator has selected a change, they add a deviation category from a pop-up menu, whereupon the system displays this category on the corresponding link in *Alignment*, as shown in Figure 4. Annotators are free to perform the annotation for each new change, or to carry out normalisation of a longer text before performing the annotation, if a separation of these tasks is preferred.

The taxonomy of deviations is based on the error categories in the Norwegian learner corpus ASK (Tenfjord et al., 2006), by which we take advantage of the experiences from the closely related target language there. So far we have made a few extensions of the ASK categories for morphological deviations. The particular taxonomy used is independent of our tool, though.

A possibility that we have not yet made use of is that a preliminary error annotation could be inferred from the se-

quence of editing operations. For example, in a suffixing language like Swedish we could automatically suggest relevant labels when noticing changes in common suffixes for tense or definiteness. Many deviation labels are easy to infer even without any linguistic insight: missing or redundant words, deviant word order, over compounding and over splitting. This is something that we plan to explore later on in the project.

5. Discussion

Although only the normalisation component of our tool has been finished so far (whereas the annotation component is being implemented), we believe that the overall transformation-based annotation of norm deviations will increase the efficiency and consistency of this process, for the following reasons: First, the conceptually different tasks of normalisation (deciding on target hypotheses) and annotation of the norm deviations are separated, with the normalisation arguably being more transparent than in previous approaches. Since the source text and normalised text are displayed separately, and the normalised text is updated incrementally, the annotator has access to all information for familiarising themselves with the specific interlanguage of the learner, and thus to make consistent decisions on target hypotheses. An additional advantage is the parallel text which is incrementally constructed based on the editing operations, allowing for search in either or both of the source and normalised texts, and for training of systems for automated annotation of norm deviations (Sproat and Jaitly, 2016). Finally, as an aside, we expect our approach to be useful also for other kinds of problems which involve parallel texts and manual annotation, such as text simplification, essay grading and anonymisation.

6. Acknowledgements

This work has been supported by an infrastructure grant from Riksbankens Jubileumsfond (SweLL – Research Infrastructure for Swedish as a Second Language, project IN16-0464:1). The ideas advanced here owe much to an earlier pilot developed by Felix Hultin (Hultin, 2017), supervised by Robert Östling and Mats Wirén. We have received valuable comments and feedback on the tool from Markus Forsberg, Lars Borin and Beáta Megyesi.

7. Bibliographical References

Boyd, A., Hana, J., Nicolas, L., Meurers, D., Wisniewski, K., andrea Abel, Schöne, K., Štindlová, B., and Vettori, C. (2014). The MERLIN corpus: Learner Language and the CEFR. In Nicoletta Calzolari (Conference Chair), et al., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Ellis, R. (1994). *The Study of Second Language Acquisition*. Oxford University Press, Oxford.

Granger, S. (2008). Learner corpora. In Anke Lüdeling et al., editors, *Corpus Linguistics. An International Handbook*, volume 1, chapter 15, pages 259–275. Mouton de Gruyter, Berlin.

Hana, J., Rosen, A., Štindlová, B., and Jäger, P. (2012). Building a learner corpus. In Nicoletta Calzolari (Conference Chair), et al., editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

Hultin, F. (2017). Correct-Annotator: An Annotation Tool for Learner Corpora. CLARIN Annual Conference 2017 in Budapest, Hungary.

Lüdeling, A., Walter, M., Kroymann, E., and Adolphs, P. (2005). Multi-level error annotation in learner corpora. In *In Proceedings of Corpus Linguistics 2005*.

Selinker, L. (1972). Interlanguage. *IRAL-International Review of Applied Linguistics*, 10(3):209–231.

Sproat, R. and Jaitly, N. (2016). RNN Approaches to Text Normalization: A Challenge. *arXiv preprint arXiv:1611.00068*.

Tenfjord, K., Meurer, P., and Hofland, K. (2006). The ASK corpus: A language learner corpus of Norwegian as a second language. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pages 1821–1824.