SVALA: Annotation of Second-Language Learner Text Based on Mostly Automatic Alignment of Parallel Corpora

Mats Wirén
Department of Linguistics
Stockholm University, Sweden
mats.wiren@ling.su.se

Arild Matsson
Språkbanken, Department of Swedish
University of Gothenburg, Sweden
arild.matsson@gu.se

Dan Rosén
Språkbanken, Department of Swedish
University of Gothenburg, Sweden
dan.rosen@svenska.gu.se

Elena Volodina
Språkbanken, Department of Swedish
University of Gothenburg, Sweden
elena.volodina@svenska.gu.se

Abstract

Annotation of second-language learner text is a cumbersome manual task which in turn requires interpretation to postulate the intended meaning of the learner's language. This paper describes SVALA, a tool which separates the logical steps in this process while providing rich visual support for each of them. The first step is to pseudonymize the learner text to fulfil the legal and ethical requirements for a distributable learner corpus. The second step is to correct the text, which is carried out in the simplest possible way by text editing. During the editing, SVALA automatically maintains a parallel corpus with alignments between words in the learner source text and corrected text, while the annotator may repair inconsistent word alignments. Finally, the actual labelling of the corrections (the postulated errors) is performed. We describe the objectives, design and workflow of SVALA, and our plans for further development.

1 Introduction

Corpus annotation, whether manual or automatic, is typically performed in a pipeline that includes tokenization, morphosyntactic tagging, lemmatization and syntactic parsing. Because of the deviations from the standard language, however, learner data puts special demands on annotation. Automatic tools trained on the language of native speakers can sometimes be applied with more or less satisfactory results even to learner language. Where available, spell and grammar checking tools providing suggestions can be used to approximate a corrected version of the text. More commonly, however, an additional manual step is added before applying a standard annotation pipeline, namely, normalization, which means changing the original learner text to a grammatically correct version. A problem with this is that there is seldom a unique grammatically correct version, and, related to this, that the agreement between different annotators is often low. For this reason, Lüdeling (2008) argues for making the corrections — in other words, construction of the *target hypotheses* — explicit so as to factor the problem of correction and the ensuing step, which is the actual error annotation, that is, labelling the type of correction that has been made to the learner text according to a taxonomy.

The aim of this paper is to describe SVALA[1], a tool for pseudonymization[2], normalization and correction annotation of learner texts. The work is part of the SweLL project (Volodina et al., 2018), whose

[1]"Svala" in Swedish means "swallow" (noun, in the sense of the bird), but is here an acronym which is a concatenation of SVA (*SV*enska som *A*ndraspråk; in English: *Swedish as a Second Language*) and LA (*L*inking and *A*nnotation).

[2]Pseudonymization of personal data means that it can be restored to its original state (re-identified) by keeping keys that are stored securely. In contrast, anonymization means that no such keys are kept, and that the data can therefore never be restored (unanonymized). For more discussion of this in the context of the project, see Megyesi et al. (2018).

Mats Wirén, Arild Matsson, Dan Rosén, and Elena Volodina 2019. SVALA: Annotation of Second-Language Learner Text Based on Mostly Automatic Alignment of Parallel Corpora. *Selected papers from the CLARIN Annual Conference 2018*. Linköping Electronic Conference Proceedings 159: 227–239.

aim is to create a platform for collecting, digitizing, pseudonymizing, normalizing and annotating learner texts, and to use this to construct a corpus of 600 texts written by learners of Swedish to be made available for research.[3] In addition, our intent is to make the methods and tools from the project as generally applicable as possible. To this end, SVALA is free software under the MIT license.[4]

In setting out this work, our objective was to arrive at a tool with the following characteristics:

1. *Uniform environment.* The tool should provide a single environment for pseudonymization, normalization and correction annotation, with a uniform representation of data to avoid problems with conversions between different formats.

2. *Lightweight data format.* The data format should be easy to work with programmatically and to convert to formats used by other systems.

3. *Intuitive interface.* From the point of view of usability, the tool should primarily target researchers in Learner Corpus Research (LCR) and Second Language Acquisition (SLA). In particular, the interface should have separate components that mirror the conceptual tasks of the problem (Ellis, 1994; Granger, 2008, page 266): detecting sensitive information for the purpose of pseudonymization, detecting learner deviations from the standard language, correcting the deviations (normalization), and finally annotating them.[5] It should be possible for an annotator to perform the different types of tasks separately, and it should even be possible to assign the tasks to annotators with different skills. The tool should provide rich visual support for the different tasks, specifically:

   (a) *Target hypotheses via text editing.* The annotator should be able to construct the target hypotheses in the simplest possible way by using text editing.

   (b) *Parallel corpus.* To make the differences between the layers of the text (pseudonymization, normalization, correction annotation) fully explicit, they should be visualized and represented as word-aligned texts in a parallel corpus (see Figure 1). The word alignments should be constructed automatically from the editing operations, but it should be possible for the annotator to correct them.

4. *Administration and scheduling.* The tool should be designed so as to make it straightforward to link it to facilities for administration and scheduling of the work of multiple annotators, as well as statistical analyses of the results of their work, including inter-annotator agreement.

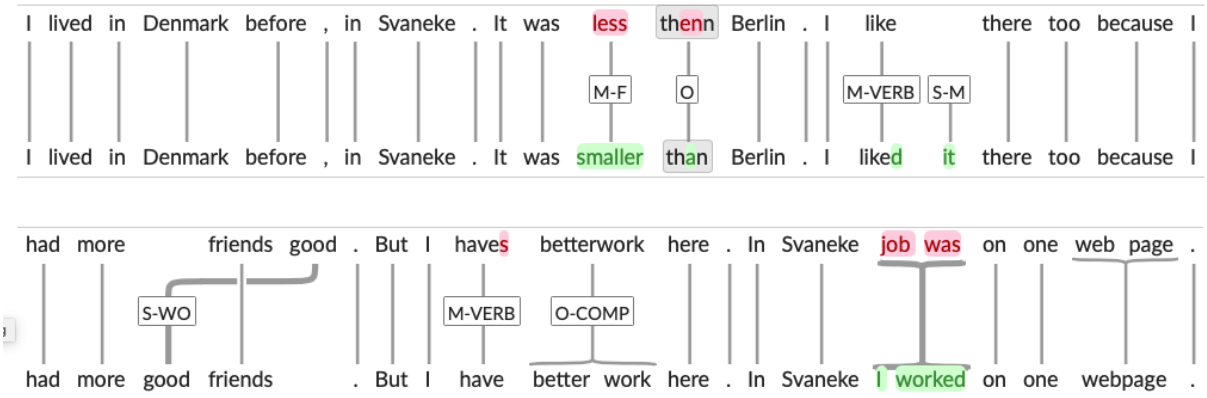


Figure 1: Example learner sentence (top rows), corrected version representing the target hypotheses (bottom rows), and correction annotation on the edges aligning the words (middle areas).

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

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

[5] Deviations may be viewed as errors relative to the standard language (Ellis, 1994; Granger, 2008). However, we annotate the corrections rather than the postulated errors, which is a slightly different notion (see Section 3.5).

The rest of this paper is organized as follows: We provide an overview of related work (Section 2), introduce the way in which the system works from the user perspective (Section 3), describe the technical solutions and data format (Section 4), and discuss what we have achieved and our plans for future work (Section 5).

2 Related work

Annotation of second-language learner texts is typically carried out using a purpose-built editing tool (or a pipeline of tools) with a hierarchical set of error categories (Granger, 2008), for example, the XML editor Oxygen in the ASK project (Tenfjord et al., 2006) and the TEITOK editor in the COPLE2 project (Mendes et al., 2016). However, the conceptually different tasks referred to in item 3 in the list in Section 1 are not always clearly separated. For example, in ASK (Tenfjord et al., 2006), the Oxygen editor used is customized 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 that can be viewed in a web browser, by which the annotation and corrections can be proofread.

In contrast, there are three learner corpora for which normalization and error annotation have been separated. The first two are Falko, a learner corpus for German (Reznicek et al., 2012), and MERLIN, a learner corpus for Czech, German and Italian (Boyd et al., 2014). Both of these use tools that support annotation in a tabular stand-off format with separate layers for transcription, target hypotheses and error tags. Furthermore, they both use the guidelines developed for Falko, in which two types of target hypotheses are distinguished: the Minimal Target Hypothesis, in which corrections are restricted to orthography, morphology and syntax, and the Extended Target Hypothesis, in which semantics, pragmatics and style are corrected. These hypotheses are representend in different layers in the stand-off format.

The third is CzeSL, a learner corpus for Czech (Rosen et al., 2014), for which the graphical annotation tool Feat has been used (Hana et al., 2012). This again includes two levels of target hypotheses, though with different scopes compared to Falko and MERLIN. At the lower level, only the orthography of isolated tokens is corrected, whereas at the higher level, ”the grammatical and lexical aspects of the learner’s language in a narrow sense” are corrected (Rosen et al., 2014, page 75). However, this level may include corrections of words that do not fit the context, and may thus affect semantics; in this sense it goes a bit further than the Minimal Target Hypothesis in Falko and MERLIN.[6] Feat displays three levels: the tokenized text (called level 0) and the two levels of target hypothesis (levels 1 and 2), with edges between the words at the different levels. Changes are made between the levels, and the annotator can join, split, add, delete and move words, as well as insert error labels for these. Feat saves the three levels of text and annotation as three word-aligned files. The texts are downloadable in that format, but are searchable only with SeLaQ, a tool which does not display the edges between levels.[7]

As for normalization, none of the systems mentioned above let the annotator correct the learner data using text editing. There is one such tool that we are aware of, however, namely, QAWI which was used in creating the Qatar Arabic Language Bank (QALB), a corpus of Arabic text with manual corrections (Obeid et al., 2013). Here, the annotator can insert, delete and move words while correcting orthographic, lexical and syntactic errors. No parallel corpus is constructed, however.

There has recently been some work on manual construction of parallel treebanks for learner and normalized texts. This work has not made use of any special-purpose (graphical) tools, however. For example, Berzak et al. (2016) added part-of-speech tags and dependency analyses to CoNLL-based textual templates for 5,124 learner sentences from the Cambridge First Certificate in English (FCE) corpus. By then adding dependency analyses and error codes to a corrected version of each of these sentences, they obtained a parallel treebank based on the pairs of learner and corrected sentences. Li and Lee (2018) took a somewhat similar approach for a corpus of sentences written by learners of Chinese. However,

[6] Alexandr Rosen, personal communication, 1 February 2019.
[7] Alexandr Rosen, personal communication, 1 February 2019.

their corpus also has word alignments between the learner and corrected texts and does not require strict pairing at the sentence level, thus comprising 600 learner sentences and 697 corrected sentences.

In translation and related fields, several tools for manual or semi-automatic word alignment of parallel corpora have been developed, for example, Interactive Linker (Ahrenberg et al., 2002; Merkel et al., 2003), ICA (Tiedemann, 2006), the Hypal4MUST interface, adapted in 2016 from the Hypal system (Obrusník, 2012) for the MUST project (Granger and Lefer, 2018), and, for multiparallel corpora, Hierarchical Alignment Tool (Graën, 2018). An assumption in our system is that the differences between the source and target texts are relatively small, which makes it possible for an automatic aligner to rely only on orthographic differences. Such differences, for example, the longest common subsequence ratio by Melamed (1999), have previously been used for the alignment of parallel corpora, but not for the alignment of learner corpora as far as we know. Also, we are not aware of any tool which includes all the tasks of pseudonymization, normalization and correction annotation in a uniform environment.

3 Functionality and user interface of SVALA

3.1 Prestudy

Before embarking on the design of SVALA, we studied existing tools and discussed them with SLA researchers to see if there were any possibilities of re-use. First, we wished to avoid linking different tools in a pipeline, as in the MERLIN project (Boyd et al., 2014) (item 1 in Section 1). This would have required conversions between the formats, which have been reported to be a source of multiple problems, including loss of information.[8] Secondly, we wanted a tool which allowed annotators to work on the different tasks independently as far as possible, supported by visualization (item 3 in Section 1), as in CzeSL (Hana et al., 2012). Thirdly, we wanted a tool that allowed normalization of a text in a traditional text editing mode (item 4), as in QAWI (Obeid et al., 2013). Finally, to make the differences explicit (particularly word order changes), we wanted a tool that represented and visualized the different layers of the text as a parallel corpus (item 5). It should also be mentioned that we deemed the formats used by other tools for text annotation, such as Brat[9] and WebAnno (Eckart de Castilho et al., 2016) not to be suitable for the intended users. Motivated by these considerations, we decided to design our own tool to fulfil the objectives described in Section 1.

As a first step, we made a proof-of-concept implementation based on some of the desiderata in Section 1, which confirmed the suitability for SLA researchers of normalization based on text editing (Hultin, 2017).[10] We then began a full-scale implementation, specifically including the incremental construction and visualization of the different layers of the text as a parallel corpus, resulting in SVALA. The tool has then undergone several design iterations based on annotation experiments with participants from the project team, resulting in gradual improvements of the functionality.

3.2 Overview of the user interface

In the normalization and correction annotation modes, the SVALA interface includes three panels related to the texts (see Figure 2): from the top, a display of the learner text (*Source text*), an editable display of the version of the text being corrected (*Target text*), and a graphic rendering of the parallel corpus, featuring word alignments of the source and target texts as well as annotations (which we refer to as the *spaghetti area*). Initially, the target text is the same as the source text, and each word is aligned to its copy, thus forming a trivial parallel corpus. To correct a text, the annotator uses the editable text panel *Target text* in Figure 2, which works like a standard display editor. Changes (that is, editing operations) can be made in any order, and upon each change, the system immediately updates the word alignments between the two texts, thereby maintaining the parallel corpus. Figure 2 shows the effects of several changes; note, in particular, the edges corresponding to the insertion of ”it”, the move of ”good” and the

[8] Adriane Boyd (2017), presentation at the Workshop on Interoperability of L2 Resources and Tools, https://sweclarin.se/sites/sweclarin.se/files/event_atachements/L2wsh_Boyd_slides.pdf.

[9] http://brat.nlplab.org/.

[10] Originally a student project at Stockholm University supervised by Mats Wirén and Robert Östling.

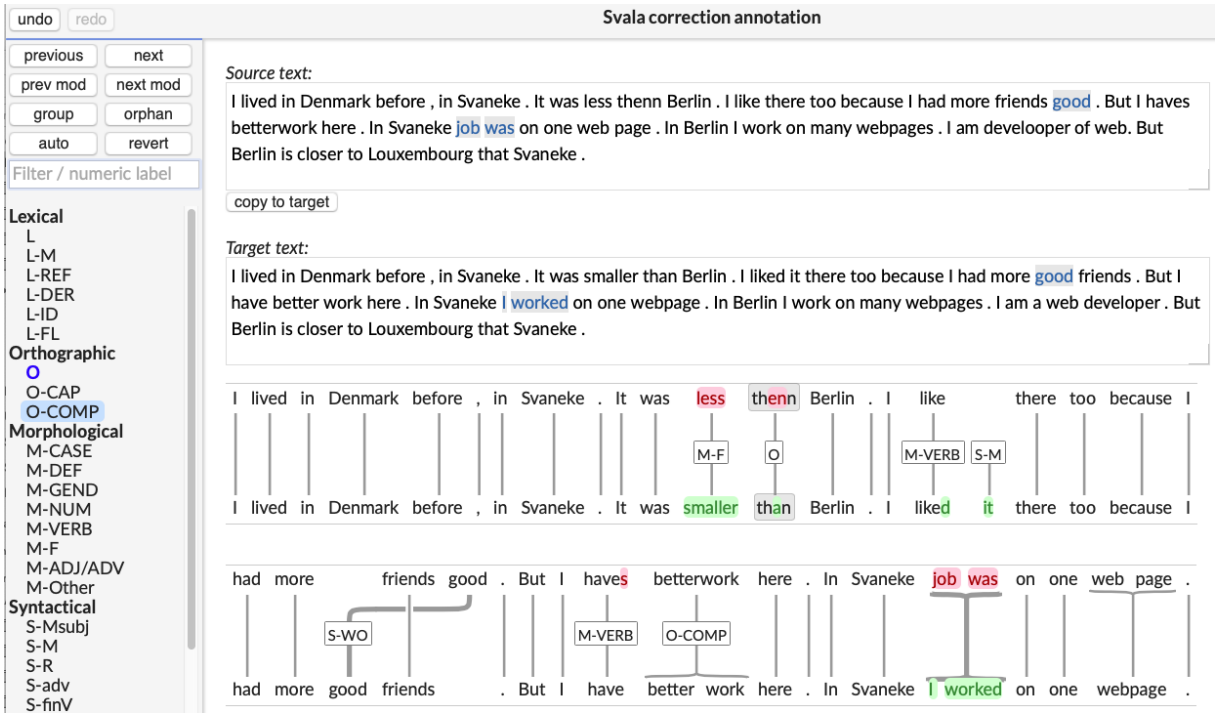


Figure 2: The SVALA graphical user interface (correction annotation mode).

replacement of ”job was” with ”I worked”. Correction annotation is made by clicking on a word or edge in the spaghetti area, and choosing a relevant label that appears in the menu to the left.

On the right of the interface, shown in Figure 4, there is a drop-down menu to switch between the different modes, open relevant guidelines, and inspect the underlying representation of the annotated data. To facilitate effective work, users can leave comments for selected sections of the text, and can browse these if the need for dicussion comes up. The comments may eventually be saved for the final version to facilitate the work for future users of the corpus.

3.3 Pseudonymization

Pseudonymization starts in a special environment that we call *Kiosk*. This is a computer with a pre-installed database and minimal support for administration of the project tasks. The Kiosk supports work on the non-pseudonymized original learner data, and is encrypted to prevent intruders from accessing learner texts. In the Kiosk, two main preprocessing steps are performed, namely, transcription and pseudonymization, the latter of which is carried out using SVALA. Once these steps have been made, hand-written and digital non-pseudonymized texts are sent to a secure storage, and the pseudonymized texts are committed to a server with which regular communication is possible.

The (sequences of) tokens that users label with pseudonymization categories are highlighted in both the *Source text* and *Target text* areas, and the pseudonyms are shown in both the spaghetti area and the *Target text* area. Edges in the spaghetti area that contain pseudonymization labels are given the status ”manually manipulated” in the data representation. This secures their highlighting in the next steps of text processing. Each pseudonymization segment which is labelled in the text is assigned a running index number, which is incremented as new entities are being pseudonymized. A display of these is provided in the area to the right; see Figure 3, where, for example, index 2 corresponds to all occurrences of the *Svaneke*.

We plan to experiment with automatic linguistic annotation of non-pseudonymized texts for two purposes: detecting candidates for pseudonymization using Named Entity Recognition (NER), and projecting morphological and grammatical information to the pseudonymized segments (for example, retaining genitive case when replacing ”John’s” with ”Adam’s”).

SVALA is pre-installed in its pseudonymization mode in the Kiosks, and works as in the usual (on-line)

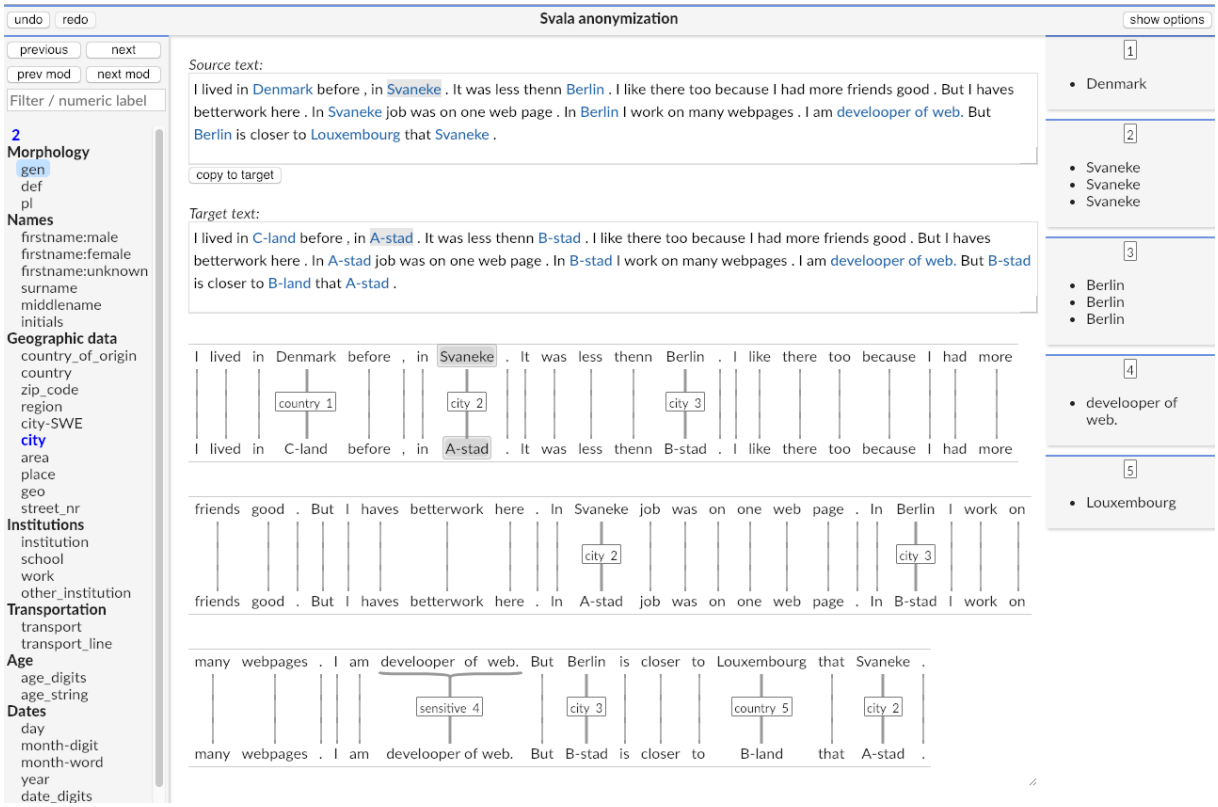


Figure 3: The SVALA pseudonymization mode.

environment in spite of the security precautions. In contrast, normalization and correction annotation are performed in the on-line version of SVALA. During the normalization step, annotators may discover errors in the pseudonymization, and because of this the pseudonymization mode is also available in the on-line version of SVALA.

A detailed description of the pseudonymization taxonomy in SVALA, the legal background of pseudonymization, and the data management in SweLL for the purpose of complying with ethical and legal demands is provided by Megyesi et al. (2018).

3.4 Normalization

The aim of normalization in SweLL is to render the text in a version which remains as close as possible to the original, but which is correct with respect to orthography, morphology and syntax, as well as semantically and stylistically coherent. To unify these incompatible requirements, the tolerance with respect to deviations from the standard language is relatively high. Correction of orthography, morphology and syntax is similar to the Minimal Target Hypothesis in the German learner corpus Falko (Reznicek et al. 2012, page 42 ff.). In contrast, the latter points go beyond this by allowing changes that affect meaning, mainly for the purpose of correcting sentences that are semantically anomalous in the context and maintaining a stylistic level which on the whole is consistent.

The purpose of normalization is twofold:

1. To render the text in a version which is amenable to automatic linguistic annotation. For Swedish, two standard pipelines for this are Sparv (Borin et al., 2016) and efselab.[11]

2. To obtain an explicit representation of the deviations relative to a postulated standard variant of the language (the target hypotheses), based on which correction annotation can be performed.

In SVALA, normalization is carried out as editing of what is initially a copy of the learner source text which appears in the *Target text* area (see Figure 4). Pseudonymized segments are highlighted to make

[11] https://github.com/robertostling/efselab.

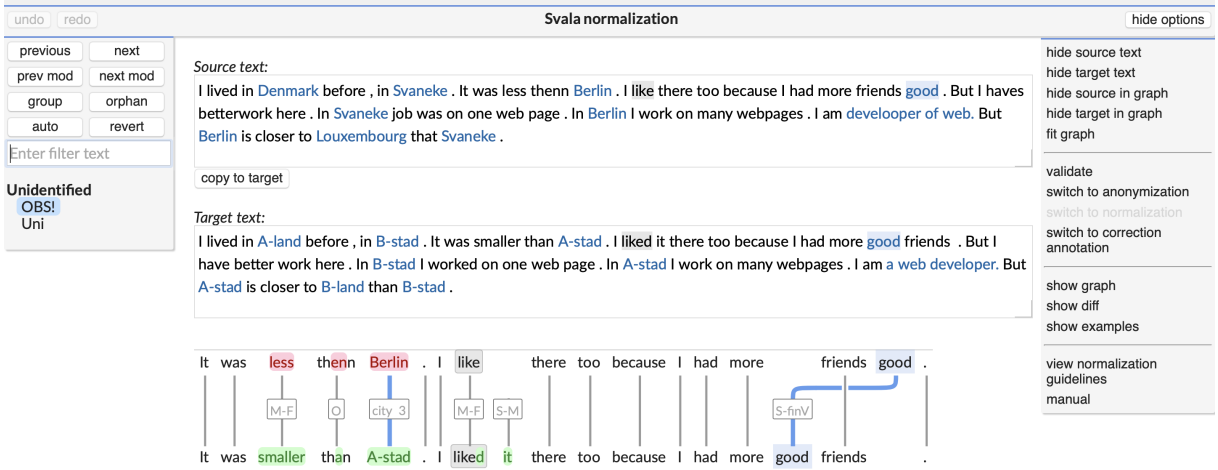


Figure 4: The SVALA normalization mode.

it clear that they were not written as such by learners. The spaghetti area displays the sentence on which the annotator is working. Differences between the source and target texts are colour-coded, creating a kind of colour map of the learner language. For usability purposes, hovering over a segment in any of the panels (source, target, spaghetti) causes the corresponding segment to be highlighted in the other areas as well, as for the word ”good” in Figure 4.

3.5 Correction annotation

By ”correction annotation” we specifically mean that we annotate the *changes* made to the target text rather than the postulated errors that occur in the learner text. The distinction is subtle, and typically there are no differences between the two notions. What it means, however, is that when a correction involves a replacement, we do not annotate errors that appear solely in the replaced material, but only the replacement as such. For example, if ”She go to the park” is replaced by ”She walks to the park”, what we annotate is the lexical replacement, but not the agreement error of ”go”, since this token does not occur in the corrected text.

Correction annotation is carried out by selecting one or several (not necessarily contiguous) tokens of the learner and corrected texts, or the edges that connect them, in the spaghetti panel. A pop-up menu is displayed to the left (see Figures 2 and 4), which contains correction labels as well as some function buttons for purposes such as grouping of tokens in the spaghetti panel. The annotator selects one or more correction labels, each of which will be displayed on the corresponding edges. The rationale for labelling the edges is that we view correction labels as relations between the learner and corrected versions of the text; because of this, the annotation is directly accessible from both. For ergonomics, and particularly for annotators who use the tool for extended periods, we provide keyboard bindings (shortcuts) for labelling and navigating between edges.

Our correction taxonomy is inspired by the error categories of ASK (Tenfjord et al., 2006), with two hierarchical levels consisting of main categories and subcategories. However, following several rounds of testing of the taxonomy in pilots, the original ASK-like taxonomy has been modified substantially. To facilitate experimentation, and to allow SVALA to be used in other projects, the error taxonomy is fully customizable.

3.6 SweLL workflow

To put SVALA into the larger context of the intended SweLL workflow, the overall text preparation and annotation steps are outlined below.

1. *Essay collection*. The original learner texts, consisting of hand-written and born-digital material, respectively, are collected. Hand-written material is scanned and transcribed.

2. *Pseudonymization*. Names, locations, institutions, etc., are annotated and automatically replaced with placeholder tokens (Megyesi et al., 2018).

3. *Normalization*. The learner text is edited for the purpose of providing a corrected version. Based on this, the system automatically constructs a parallel corpus of the learner and normalized texts, displayed in a third panel with word alignments that may need to be corrected by the annotator.

4. *Correction annotation*.

   (a) Misaligned edges between words are corrected.

   (b) Labels describing the deviations of the original text relative to the corrected version are added.

5. *Linguistic annotation*. Automatic annotation is made by way of part-of-speech tagging, lemmatization, dependency parsing, word sense disambiguation, etc. The annotation is added to the normalized version of the text, but we plan to experiment with automatic annotation of the learner version as well with the goal of obtaining a parallel treebank (Berzak et al., 2016; Li and Lee, 2018).

6. *Import to a corpus search interface.*

Work on steps 5 and 6 above have not yet begun, but for step 1, we have currently (January 2019) collected 388 essays from speakers of more than 30 native languages with different Swedish-language proficiency levels. To guide the design of SVALA, we have made several iterations (pilots) in which each time we have annotated around 30–40 essays through steps 1–4 of the workflow outlined above. Each time, we measured inter-annotator agreement (IAA) as a way of evaluating reliability and quality of the annotations, and assessed the consistency of our guidelines, correction label taxonomy and the tool functionality.

These experiments brought into focus the fact that merging normalization and correction annotation into one step produces a lot of uncertainty as to what is measured by IAA: whether annotators agree on a) which token sequence to change, and/or b) how to change it, and/or c) which correction label to assign. Similar conclusions have been drawn by Rosen et al. (2014) and Boyd (2018). As a consequence, pseudonymization, normalization and correction annotation will be assessed separately. In particular, for normalization, we need to know to what extent annotators agree on the token-level changes (target hypotheses), and for correction annotation, we need to know to what extent annotators agree on a correction label given a particular normalization.

This separation of components is also reflected in the construction of the SweLL corpus. To begin with, we allow only one master version of an pseudonymized text (see Figure 5). Thus, while pseudonymization of a text may be assigned to several assistants, only one version, the *master*, is saved for use in the successive normalization step. If it is discovered during normalization that corrections of the pseudonymization are needed, these are made in the pseudonymized master version.

The same principle is used in the normalization step. While several independent normalizations will be produced for the purpose of comparison, only one consensus annotation will be saved as the master version for use in correction annotation, and as part of the SweLL corpus. No further normalization changes will be possible once this version has been assigned for correction labelling.

We argue that normalization is possibly the most critical step and needs to be performed by highly competent staff with SLA training. This is also why step 4 a (correction of misaligned edges) is not done as part of the normalization, even though that is possible: the annotators need to concentrate on the text as such. As a basis of normalization, we have developed thorough guidelines and have had at least two people with an SLA profile discuss the normalization decisions for the same texts until they reach consensus. Only then each annotator works individually with normalization. Because of the amount interpretation involved, we believe that previous training in SLA theory as well as practical work with second-language learner texts reduce the subjectivity. Further discussions of a number of texts are a prerequisite for agreement on target hypotheses. This way we are also decreasing the complexity level of the correction annotation step.

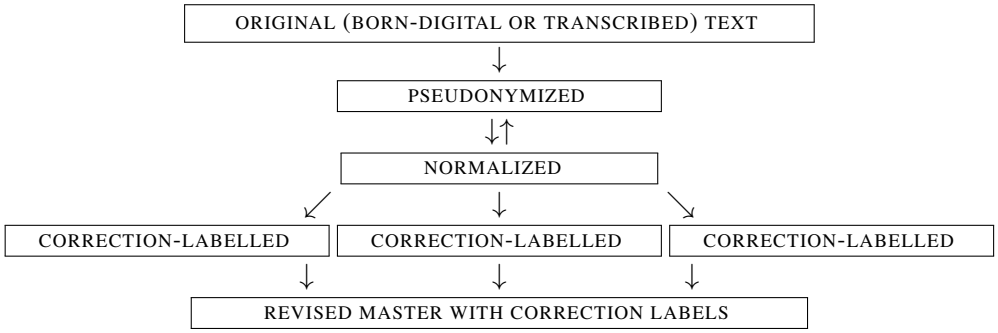


Figure 5: Versions of processed texts stored in SVALA.

Unlike in steps 2 and 3, we see a need for several versions of the same text in step 4 (correction annotation) as a prerequisite to secure consistent annotation in the whole corpus. Not all of the texts need to have several correction labelled versions, however, but for a least 20% of the corpus this will be done in order to report inter-annotator agreement.

To summarize, the above described decisions and considerations of reliability and consistency of the SweLL corpus annotation have influenced the design of the SVALA tool.

4 System design

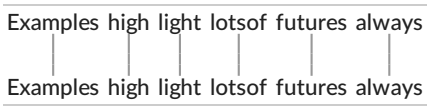
4.1 Overview

The tool is developed as a web page in TypeScript, a backwards-compatible statically typed version of JavaScript. This allows the tool to be run on all major operating systems using a browser without any installation on the user side. The interface is built using the modern and commonplace library React. The parallel corpus editing state is internally stored as a plain old JavaScript object which can be trivially serialised to and from the JavaScript object notation format JSON. Although it is conventional to use XML as a representation format in corpora, we argue that the difference between XML and JSON as a structured format is mostly superficial as there are libraries for lossless conversion between the two. Note that users with already labelled corpora can use our tool to visualize their corpus and edit it further by exporting it to the lightweight format outlined below. We are considering providing our format in a corpus conversion tool such as Pepper (Zipser and Romary, 2010).

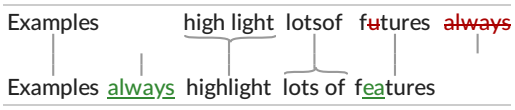
For illustration of the format, here is a small contrived example of a source (learner) text and a target text with one token each in our format:

```
{ "source": [{"id": "s0", "text": "Example "}],
  "target": [{"id": "t0", "text": "Token "}],
  "edges": {
    "e-s0-t0": {
      "id": "e-s0-t0",
      "ids": ["s0", "t0"],
      "labels": ["L"],
      "manual": false
    }
  }
}
```

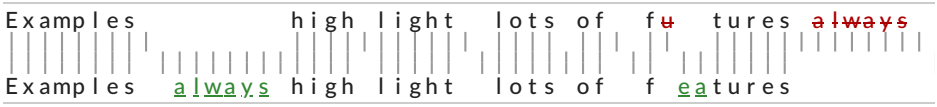
Each text is an array of tokens under their respective key (`source` and `target`). Each token consists of its text (including whitespace) and a unique identifier. Edges are stored in an unordered object since they have no order. These refer to the tokens they connect by their identifiers in the `ids` field. Annotations are put on the edges (not on tokens themselves) and each edge can have multiple labels so they are represented as an array of strings. Here the user has used the label `L`. The `manual` field is used to tell the linker to not consider it for automatic linking. This is explained in Section 4.3. Edges are strictly speaking multi-edges since they may link multiple tokens together (not just two). Edges have an identifier which is by default derived automatically from its token identifiers. This is not strictly needed as nothing in the format ever refers to this identifier, and can be considered an implementation convenience (as well as it



(a) State before editing.



(b) State after editing.



(c) Character based diff: this is lifted to the token level in Figure 6b. (not shown to the user)

Figure 6: Before and after editing. The automatic aligner gets most words right and the user will later manually connect the unlinked words.

being reflected twice). The rest of this section describes how the mostly automatic alignment procedure works.

4.2 Alignment procedure

Our system builds a corrected version aligned with the learner text using the tokens, the groups of edges between these, and any labels attached to the groups. How is this alignment calculated? We start with a standard diff edit script on the *character level*. Internally, the representation in Figure 6c is generated, which is calculated using Myers's diff algorithm (Myers, 1986) provided by the diff-match-patch[12] library. Each character is associated with the token it originates from. Next, these character-level alignments are lifted to the token level. Spaces are not used for alignment to avoid giving rise to too many false positives. We can now read off from this representation which tokens should be aligned. For each pair of matching characters, we add an edge to their corresponding tokens. For example, since there is an edge between the *h* in *high* and *highlight*, these two words are linked. Furthermore, there is an edge between the *l* in *light* to this target word too, so all these three words should be linked. There are no other characters linked to characters from these tokens, so exactly these three will become a group. The other tokens are connected analogously.

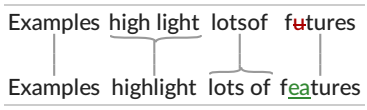
4.3 Manual alignments: word order changes and inconsistent edges

In Figure 6b, the two occurrences of the word *always* are not aligned. The user can correct this error by selecting both occurrences of *always* and clicking the *group* button (not shown here). After this grouping we are in a state where the parallel structure has one manual alignment pertaining to the word *always*, with all other words being candidates for automatic (re-)alignment. To (re-)align these we carry out the same procedure as before but excluding the manually aligned *always*: We first *remove* manually aligned words, align the rest of the text automatically (see Figure 7a), and then *insert* the manually aligned words again in their correct position (Figure 7b). Here the correct position is where they were removed from the respective texts. The editor indicates that this edge is manually aligned by colouring it blue and (for the purposes of this article) making it dotted. These edges interact with automatically aligned (grey) edges differently. How this works is explained in the next section.

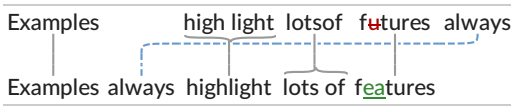
4.4 Editing in the presence of both manual and automatic alignments

The tokens that have been manually aligned are remembered by the editor. The user may now go on editing the target hypothesis. Things are straightforward as long as the edit takes place wholly in either an automatically aligned passage or in a manually aligned passage. When editing across these boundaries, the manual segment is contagious in the sense that it extends as much as needed. For example, if we select *always highlight* in Figure 7b and replace it with *alwayXighligt* the state becomes as shown in Figure 7c. However, if we cross the boundary of *of features* in the same starting state of Figure 7b to *oXeatures* we get the state of Figure 7d. Here the edit is not contagious: the automatic alignment decided to not make

[12] `https://github.com/google/diff-match-patch`.

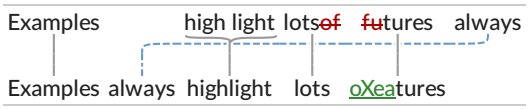


(a) Diff without *always*.



(b) Diff with *always* correctly inserted from Figure 7a.

(c) Editing from Figure 7b between *always* and *highlight*.



(d) Editing from Figure 7b between *of* and *features*.

Figure 7: Aligning and editing in the presence of a manually aligned word (here *always*).

a big component, instead it chose to split to align the words independently. In case the manual aligner has absorbed too much, our editor provides a way of removing the manual alignment tag. The editor will then fall back to the automatic aligner for those tokens.

5 Discussion

We have described SVALA, a tool for pseudonymization, normalization and correction annotation of second-language learner text. In the SVALA interface and workflow, these tasks are separate but interdependent, with rich visual support for each of them. Normalization is carried out in the simplest possible way by text editing, in the course of which the tool automatically maintains a parallel corpus with alignments between words in the learner source text and corrected text.

SVALA is being used in the SweLL project, which aims at constructing a corpus of 600 texts from learners of Swedish. However, given the relative lack of established infrastructure for research in second-language learning in CLARIN, it is also targeted as a general utility. To this end, SVALA is free software under the MIT license, and the taxonomy of correction labels is customizable. Alternatively, users of corpora that are annotated with different labels could export their format to the lightweight format of SVALA in order to make use of the system.

Several avenues for further development of SVALA are possible. In particular, the notion of parallel corpus is useful for several reasons. First, it would be straightforward to extend SVALA with additional levels of target hypotheses, for example, a purely orthographic level as in the Feat tool (Hana et al., 2012) or something akin to the Extended Target Hypothesis in Falko (Reznicek et al., 2012, page 51). Secondly, the parallel corpus is relevant for automatic linguistic annotation (step 5 in the SweLL workflow as outlined in Section 3.6). Although annotation of the corrected text is likely to be most straightforward, the alignments of the parallel corpus open up the possibility of using annotation transfer to project relevant labels (also) to the learner source text. Thirdly, we expect that a parallel corpus of learner and corrected texts will be an independently valuable resource for later development of automatic tools for identification of learner errors.

As for correction annotation, it is possible to infer some of this automatically from the editing operations and the differences between the learner and corrected texts, for example, changes of orthography, missing or redundant words, and word order. Specifically, in an agglutinating language the system might suggest relevant morphosyntactic labels upon detecting changes in suffixes for definiteness, number or tense. We experimented with automatic correction annotation in the pilot system by Hultin (2017), and the Feat tool (Rosen et al., 2014, Section 5.5) provides it for several cases.

We have argued that SVALA, with its intuitive interface, uniform environment, lightweight data format and mostly automatic word alignment, provides a rational tool and methodology for pseudonymization, normalization and correction annotation of second-language learner text. By virtue of this, we also expect it to provide a useful starting-point for the construction of parallel treebanks for learner language, in which each of the learner and normalized texts are linguistically analysed.

Acknowledgements

This work has been supported by Riksbankens Jubileumsfond through the SweLL project (grant IN16-0464:1). We want to thank the three reviewers for helpful comments, and Robert Östling, Markus Forsberg, Lars Borin and our co-workers in the SweLL project for valuable feedback.

References

[Ahrenberg et al.2002] Lars Ahrenberg, Mikael Andersson, and Magnus Merkel. 2002. A system for incremental and interactive word linking. In *LREC'02*, pages 485–490.

[Berzak et al.2016] Yevgeni Berzak, Jessica Kenney, Carolyn Spadine, Jing Xian Wang, Lucia Lam, Keiko Sophie Mori, Sebastian Garza, and Boris Katz. 2016. Universal Dependencies for Learner English. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 737–746, Berlin, Germany. Association for Computational Linguistics.

[Borin et al.2016] Lars Borin, Markus Forsberg, Martin Hammarstedt, Dan Rosén, Roland Schäfer, and Anne Schumacher. 2016. Sparv: Språkbanken's corpus annotation pipeline infrastructure. In *SLTC 2016. The Sixth Swedish Language Technology Conference, Umeå University, 17-18 November, 2016*.

[Boyd et al.2014] Adriane Boyd, Jirka Hana, Lionel Nicolas, Detmar Meurers, Katrin Wisniewski, Andrea Abel, Karin Schöne, Barbora Štindlová, and Chiara Vettori. 2014. The MERLIN corpus: Learner Language and the CEFR. In *LREC'14*, Reykjavik, Iceland. European Language Resources Association (ELRA).

[Boyd2018] Adriane Boyd. 2018. Normalization in Context: Inter-Annotator Agreement for Meaning-Based Target Hypothesis Annotation. In *Proceedings of Natural Language Processing for Computer-Assisted Language Learning (NLP4CALL)*, Stockholm, Sweden.

[Eckart de Castilho et al.2016] Richard Eckart de Castilho, Éva Mújdricza-Maydt, Seid Muhie Yimam, Silvana Hartmann, Iryna Gurevych, Anette Frank, and Chris Biemann. 2016. A web-based tool for the integrated annotation of semantic and syntactic structures. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 76–84. The COLING 2016 Organizing Committee.

[Ellis1994] Rod Ellis. 1994. *The Study of Second Language Acquisition*. Oxford University Press, Oxford.

[Granger and Lefer2018] Sylviane Granger and Marie-Aude Lefer. 2018. The Translation-oriented Annotation System: A tripartite annotation system for translation research. In *International Symposium on Parallel Corpora (ECETT — PaCor)*, pages 61–63. Instituto Universitario de Lenguas Modernas y Traductores, Facultad de Filología, Universidad Complutense de Madrid, Spain.

[Granger2008] Sylviane Granger. 2008. Learner corpora. In Anke Lüdeling and Merja Kytö, editors, *Corpus Linguistics. An International Handbook*, volume 1, chapter 15, pages 259–275. Mouton de Gruyter, Berlin.

[Graën2018] Johannes Graën. 2018. *Exploiting Alignment in Multiparallel Corpora for Applications in Linguistics and Language Learning*. Ph.D. thesis, University of Zurich.

[Hana et al.2012] Jirka Hana, Alexandr Rosen, Barbora Štindlová, and Petr Jäger. 2012. Building a learner corpus. In *LREC'12*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

[Hultin2017] Felix Hultin. 2017. Correct-Annotator: An Annotation Tool for Learner Corpora. CLARIN Annual Conference 2017 in Budapest, Hungary.

[Li and Lee2018] Keying Li and John Lee. 2018. L1–L2 Parallel Treebank of Learner Chinese: Overused and Underused Syntactic Structures. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan. European Language Resource Association.

[Lüdeling2008] Anke Lüdeling. 2008. Mehrdeutigkeiten und Kategorisierung: Probleme bei der Annotation von Lernerkorpora. In M. Walter and P. Grommes, editors, *Fortgeschrittene Lernervaritäten: Korpuslinguistik und Zweitspracherwerbsforschung*, pages 119–140. Max Niemeyer, Tübingen, Germany.

[Megyesi et al.2018] Beáta Megyesi, Lena Granstedt, Sofia Johansson, Dan Rosén, Carl-Johan Schenström, Gunlög Sundberg, Mats Wirén, and Elena Volodina. 2018. Learner Corpus Anonymization in the Age of GDPR: Insights from the Creation of a Learner Corpus of Swedish. In *Proceedings of Natural Language Processing for Computer-Assisted Language Learning (NLP4CALL)*, Stockholm, Sweden.

[Melamed1999] I. Dan Melamed. 1999. Bitext maps and alignment via pattern recognition. *Computational Linguistics*, 25(1):107–130, March.

[Mendes et al.2016] Amália Mendes, Sandra Antunes, Maarten Janssen, and Anabela Gonçalves. 2016. The COPLE2 corpus: a learner corpus for Portuguese. In *LREC'16*.

[Merkel et al.2003] Magnus Merkel, Michael Petterstedt, and Lars Ahrenberg. 2003. Interactive word alignment for corpus linguistics. In *Proc. Corpus Linguistics 2003*.

[Myers1986] Eugene W. Myers. 1986. An $O(ND)$ difference algorithm and its variations. *Algorithmica*, 1(1):251–266.

[Obeid et al.2013] Ossama Obeid, Wajdi Zaghouani, Behrang Mohit, Nizar Habash, Kemal Oflazer, and Nadi Tomeh. 2013. A Web-based Annotation Framework For Large-Scale Text Correction. In *The Companion Volume of the Proceedings of IJCNLP 2013: System Demonstrations*, pages 1–4. Asian Federation of Natural Language Processing.

[Obrusník2012] Adam Obrusník. 2012. A hybrid approach to parallel text alignment. Masaryk University, Faculty of Arts, Department of English and American Studies, Brno, Czech Republic. Bachelor's Diploma Thesis.

[Reznicek et al.2012] M. Reznicek, A. Lüdeling, C. Krummes, and F. Schwantuschke. 2012. Das Falko-Handbuch. Korpusaufbau und Annotationen Version 2.0. Humboldt-Universität zu Berlin, Berlin, Germany.

[Rosen et al.2014] Alexandr Rosen, Jirka Hana, Barbora Štindlová, and Anna Feldman. 2014. Evaluating and automating the annotation of a learner corpus. *Lang. Resour. Eval.*, 48(1):65–92, March.

[Tenfjord et al.2006] Kari Tenfjord, Paul Meurer, and Knut Hofland. 2006. The ASK corpus: A language learner corpus of Norwegian as a second language. In *LREC'06*, pages 1821–1824.

[Tiedemann2006] Jörg Tiedemann. 2006. ISA & ICA—two web interfaces for interactive alignment of bitexts. In *LREC 2006*.

[Volodina et al.2018] Elena Volodina, Lena Granstedt, Beáta Megyesi, Julia Prentice, Dan Rosén, Carl-Johan Schenström, Gunlög Sundberg, and Mats Wirén. 2018. Annotation of learner corpora: first SweLL insights. In *Abstracts of the Swedish Language Technology Conference (SLTC) 2018*, Stockholm, Sweden.

[Zipser and Romary2010] Florian Zipser and Laurent Romary. 2010. A model oriented approach to the mapping of annotation formats using standards. In *Workshop on Language Resource and Language Technology Standards, LREC 2010*, La Valette, Malta, May.