A Semi-Automatic Word-Level Annotation and Transcription Tool for Spelling Error Categories

L. Linhuber[1], S. Stüker[1], R. Lavalley[2], and K. Berkling[2]

[1] Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe, Germany
sebastian.stueker@kit.edu

[2] Cooperative State University, Department of Computer Science, Karlsruhe, Germany
berkling@dhbw-karlsruhe.de

Abstract. In order to train and evaluate tools for the automatic transcription of misspelled texts and automatic annotation of over 20 spelling error categories, it is important to create training data. A very large database of children's freely written text was collected in the past and in this paper we describe the tool that we have developed in order to manually transcribe and annotate the data. The manual transcription comprises the reconstruction of the orthographically correct word sequence. Annotation is performed on a per-word basis with respect to committed (child spelling) and potential (correct word) spelling error categories. The tool supports human transcribers by suggesting automatically generated annotations. Consistent annotations are propagated and data is presented to the user in a sorted manner to minimize human effort. The tool has been implemented as a web application that makes use of PHP on the server side and a lightweight Java GUI on the client side. The annotated data is stored in a custom made XML schema.

Keywords: annotation, transcription, applications for education, language resources, orthography

1 Introduction

Proficient reading and writing skills are a prerequisites for successful citizens in today's society. Comparative studies in Germany, such as the the *Program for International Student Assessment* (PISA) and the *Progress in International Reading Literacy Study* (PIRLS) [3], have shown that around 25% of German school children do not reach the minimal competence level necessary to function effectively in society by the age of 15. Diagnostic tools on the market today offer pricey one-time spelling diagnosis on a fixed test set with high-density error-prone and unnatural text and pre-specified word field analysis. Research by Fay [4] has shown that this sort of error analysis deviates, at least in parts, significantly from the error profile derived from a child's freely written text. Thus an analysis of freely written text gives a more natural picture of the child's competence level.

The goal of this work is to support or replace manual expert effort with automated transcription of child text (achieved) into correctly written text (target) followed by automated annotation of error categories.

2 Previous Work

In [2, 5, 6] we have demonstrated on a small available data set the feasibility of creating a system for automatic error category analysis.

In order to obtain the data necessary for training, development and evaluation of our automatic tool we have collected a large amount of data at German schools [1]. The collected data contains 14,563 sentences which then needed to be transcribed and annotated.

The transcription part of this task consists of reconstructing the orthographically correct version (*target text*) of what the child has written (*achieved text*). A significant part of the work consists of creating an accurate word-level alignment from the text. This task poses some difficulty when the child's spelling errors are committed at the supra word level, adding superflous words ("Ich gehe in zu die Stadt"), splitting or connecting words ("Haustier" vs. "Haus Tier"), wrong grammatical forms ("auf den Baum" instead of "auf der Baum") or word choice ("Ich gehe in die Stadt" vs. "Ich gehe zu die Stadt"). The alignment is therefore not injective. Since we do not deal with grammatical errors, including supra word level problems listed above, the alignment is always surjective.

The purpose of this paper is to present the unified transcription/annotation tool that supports the human annotation task in several ways.

- It propagates annotations so that annotators see each error category only twice.
- It determines the order in which to annotate the data as to reduce human effort.

The tool stores the annotated data in XML-format using a custom made schema which is well suited for the processing necessary for training and evaluating our automatic spelling error categorization tools. The details of this format can be found in [1] and are beyond the scope of this paper.

3 System Overview

Our tool has a client server architecture as depicted in Figure 1. The server contains the main functionality and is programmed in PHP. A lightweight client is written in Java. The server works with the outputs of either Module 1 or Module 2. After converting the output to XML-format, these are then sorted and serve as the basis from which the GUI will select the top X files presented to the human. Not all are presented due to performance reasons. The sorting algorithm is modular and can be exchanged as necessary based on the task. The client gives the user the option of choosing module and error category to edit, thereby

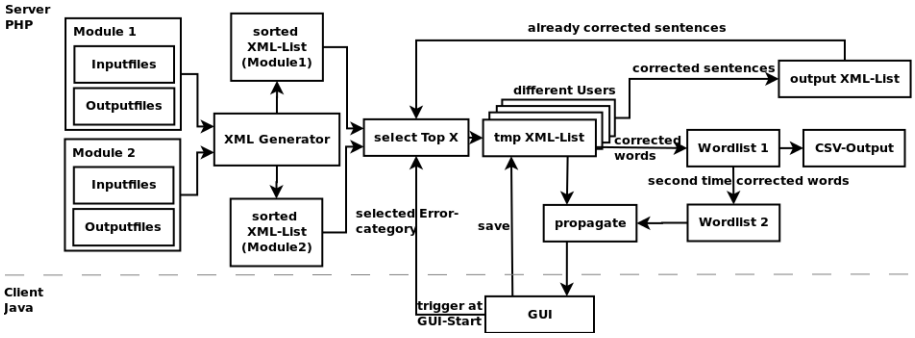

Fig. 1. System architecture of the annotation tool

triggering the server to present the relevant selected top sentences. Sentences in the list contain only those which have not been finished yet, containing at least one word that was not yet propagated or labeled for the chosen error category. The selected sentences are then written to a temporary file. A new temporary list is generated when the old list is finished or when another client triggers the process. Sentences from the client are saved into the *Temporary XML-List* as the user browses through the sentences with "next" or "prev". Saving a sentence results in removing it from *Temporary XML-List* of files and storing the completed sentence in the *Output XML-List*.

In addition, all corrected words occurring in processed sentences (after saved and moved to *Output XML-List*) are saved in *Wordlist 1*. All words in *Wordlist 1* that contain the error category that is presently annotated, are also saved in *Wordlist 2* unless they already exist in *Wordlist 2*. If there are discrepancies between the two lists regarding the annotation, an error message is sent to the user and the user has the opportunity to correct the mistake. If the user consistently annotates the word with the same error given "target" and "achieved" word annotations, this is noted by comparing *Wordlist 1* and *Wordlist 2*. As a result, the annotation is then propagated. Propagation of previous annotations is done in a modular way. The data to be worked on is not altered. Instead the alteration is done when displaying new data in the GUI. In this way, the propagation hold for all new data sets. The annotation is saved as the user moves through the sentences and saves them into *Output XML-List*.

4 Graphical User Interface

This section describes the GUI in more detail. On the start-up screen, the user chooses from Module 1 (correcting word alignment and transcription) or Module 2 (annotation of spelling errors). The error category to be worked can be chosen. (It's easier and faster to correct only one error category at a time instead of all error categories at the same time.)

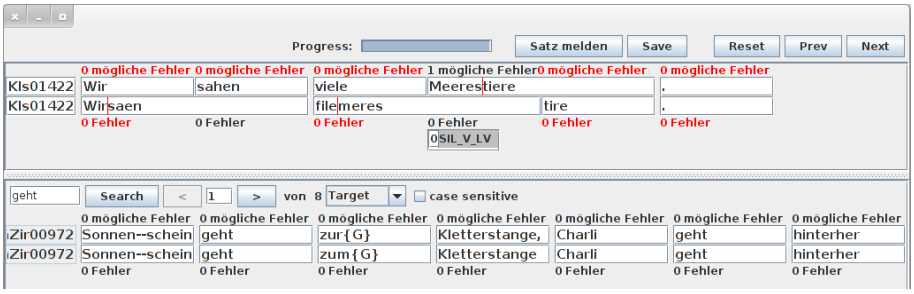

Fig. 2. Screenshot of the GUI for the transcription part of the tool

4.1 Transcription

After choosing module and error category to label, various pieces of information about the sentence are presented to the human transcriber. The upper part of the GUI shows the current target- and achieved-sentence with the word alignment and the spelling error categories. A word can consist of one or several different word-parts. A word or word-part in the target-sentence is always connected to exactly one word or word-part in the achieved-sentence and vice-versa. If a child for example wrote two words mistakenly as one (As in "Wirsahen" in Figure 2) the achieved-sentence will contain one word with two word-parts (namely "Wir", "sahen" connected to the words "Wir" und "sahen" on the Target side). Word-parts are separated by a red line. The human annotator can change the word alignment and the word-text by splitting, merging, deleting or inserting words or word-parts.

4.2 Navigation

The following buttons provide further functionality: Navigation with the buttons 'next' and 'prev'; by clicking on 'save' the word disappears and is saved on the server; The button 'reset' undoes the last operation; The button 'Satz melden' marks and removes sentences with an unexpected or unusual error. In the lower part of the GUI the user can search for a word in all already corrected sentences.

4.3 Annotation and Propagation

Error annotation is done by using the pop-up window that is presented at word level, either achieved or target word. The user then specifies the number of potential errors for target words and the number of committed errors for achieved words. As explained in the previous section the GUI has the ability to propagate error annotations. Therefore, the user has to correct the same word only twice. Propagated words are marked in red for the user. The error category occurrence of potential errors with respect to the target word are independent of the errors

committed in the achieved word. Target words can then be propagated without relation to achieved words. In contrast the error rate of an achieved word depends on the target word. E.g., the achieved word 'im' can be a misspelling of the target word 'ihm' or can be correctly spelled if the target word is im'. Achieved words are therefore propagated only in combination with their corresponding target word. An already propagated word cannot be changed or overwritten. To avoid the propagation of a wrongly annotated word, the system checks for annotator consistency. At the moment, a word has to be annotated twice in the same manner before propagation. If a second annotation of a word differs from the first annotation of the word, a message window is displayed. The user can then decide to overwrite the already saved value or to change the current annotation before the result is then propagated.

5 Preliminary User Tests

After determining the best order in which to present sentences in a simulation we have used the system for a first round of annotations. Two annotators have worked on a subset of 1,000 sentences for two different error categories. Namely, SIL_V_LV[3] (e.g., 'n**eh**men') and SIL_V_KV (e.g., 'ne**nn**en'). For error category SIL_V_LV Figure 3 plots preliminary results for the propagated word rate over the 1,000 sentences, indicating a significant reduction in labor.

The word rate is calculated as the number of not yet propagated words divided by the number of all words. As it can be seen the word rate decreased from 1 (100% of the words still need to be propagated) in the beginning to about 25% after only 1,000 sentences. This means that 25% of words have not been seen twice in these 1,000 sentences. Thus the reduction in the amount of work due to propagation is about 75%. It took two hours and 53 minutes to annotate the 1,000 sentences with an average sentence annotation time of about 10.4 seconds. For these sentences of average length of 9.0 words, annotators were able to annotate 104 words per minute and 30.0 unique words per minute.

6 Conclusion

In this paper it has been shown through preliminary usage of a newly built GUI for data annotation that annotation of word-level tags can be achieved in a robust and time-saving manner. While there are other tools in the market that work with time-aligned data, our hope is that in future work, we will be able to integrate the XML output with these tools to support reuse of data annotation tools. Furthermore, the system is built in such a way that it can support user-specific annotation schemes. The tool is therefore not hard-coded for our presently used error categories. Any word-level tag can be integrated into the GUI simply by changing the tags of the input CSV-formatted files.

[3] LV (length vowel) denotes the notation of length for vowels through the use of the letter <h>, preceding a consonant

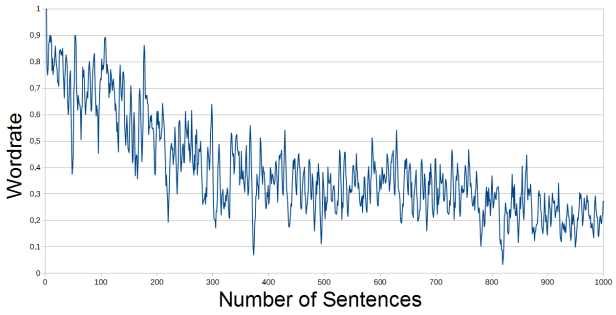

Fig. 3. propagated word rate over 1000 sentences

7 Acknowledgements

The work leading to these results was in part funded by a research grant from the German Research Foundation (DFG).

References

1. Berkling, K., Fay, J., Ghayoomi, M., Hein, K., Lavalley, R., Linhuber, L., Stüker, S.: A database of freely written texts of german school students for the purpose of automatic spelling error classification. In: The 9th edition of the Language Resources and Evaluation Conference (LREC 2014). Reykjavik, Iceland (26-31 May 2014)
2. Berkling, K., Fay, J., Stüker, S.: Speech technology-based framework for quantitative analysis of german spelling errors in freely composed childrens texts. In: The 2011 Workshop of the ISCA Special Interest Group on Speech and Language Technology in Education (SLaTE 2011). Venice, Italy (August 2011)
3. Bos, W.: IGLU: Einige Länger der BRD im nationalen und internationalen Vergleich. Münster (2004)
4. Fay, J.: Kompetenzfacetten in der Rechtschreibdiagnostik. Rechtschreibleistung im Test und im freien Text. In: Bermerich-Vos, A. (ed.) Didaktik Deutsch: Symposium Deutschdidaktik, vol. 29, pp. 15–36. Schneider Verlag (2010)
5. Fay, J., Berkling, K., Stüker, S.: Automatische Analyse von Rechtschreibfähigkeit auf Basis von Speech-Processing-Technologien. Didaktik Deutsch, Halbjahresschrift für die Didaktik der deutschen Sprache und Literatur 19(33) (2012)
6. Stüker, S., Fay, J., Berkling, K.: Towards context-dependent phonetic spelling error correction in childrens freely composed text for diagnostic and pedagogical purposes. In: Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH 2011). Florence, Italy (August 2011)