Apples – Journal of Applied Language Studies
Vol. 8, 3, 2014, 5-20

Establishing a Standardised Procedure for Building Learner Corpora

Aivars Glaznieks, Lionel Nicolas, Egon Stemle, Andrea Abel & Verena Lyding
European Academy of Bozen/Bolzano

Decisions at the outset of preparing a learner corpus are of crucial importance for how the corpus can be built and how it can be analysed later on. This paper presents a generic workflow to build learner corpora while taking into account the needs of the users. The workflow results from an extensive collaboration between linguists that annotate and use the corpus and computer linguists that are responsible for providing technical support. The paper addresses the linguists' research needs as well as the availability and usability of language technology tools necessary to meet them. We demonstrate and illustrate the relevance of the workflow using results and examples from our L1 learner corpus of German ("KoKo").

Keywords: L1 learner corpus, corpus building workflow, German as a first language

1 Introduction

The field of learner corpus linguistics refers to *learner corpora* as "systematic computerized collections of texts produced by language learners" (Nesselhauf 2004: 40). Likewise, a generally accepted definition of *language learners* is given by Granger (2008), who says that language learners are "speakers who learn a language which is neither their first language nor an institutionalised additional language in the country where they live" (Granger 2008: 260). This narrow definition covers only foreign language (FL) learners and excludes L2 learners as well as L1 learners. However, research in need of learner corpora is concerned with learners' interlanguage, i.e. "transitional language" in general, and with finding factors that influence it (Granger 2008: 259). This is why researchers started to use the term *language learners* also for L2 and L1 learners (e.g. Hana et al. 2010; Abel & Glaznieks in press). Learner corpora are usually *error-tagged*, that is, orthographic, lexical, and grammatical errors in the corpus have been annotated with the help of a standardised system of error tags (Granger 2003); in addition, the corpora provide meta-information, for example, on the authors' L1, age, gender, etc. Other valuable information, from all levels of linguistic description can be annotated as well.

Corresponding author's email: aivars.glaznieks@eurac.edu
ISSN: 1457-9863
Publisher: Centre for Applied Language Studies, University of Jyväskylä
© 2014: The authors
http://apples.jyu.fi

Annotations can be done automatically, which is often the case for lemma and part-of-speech (POS) information, or manually which usually involves orthographic, lexical, and grammatical errors. Technically, the annotations are done either inline (cf. Granger 2003) or in a multi-layered way using a stand-off format (cf. Lüdeling et al. 2005; Reznicek et al. 2013; Zinsmeister & Breckle 2012; Hana et al. 2010, 2012).

Decisions at the outset of preparing a learner corpus, such as the choice of software tools, data formats, and annotation procedures, may have substantial implications on the way the linguistic data can be retrieved and analysed. The selection of appropriate software tools and data formats for the transcription and annotation of the original data is a challenge for corpus linguists, as the software needs to be flexible (to facilitate intuitive and speedy transcription) and powerful (to meet annotation demands). Besides this, the software also needs to be adaptable (to enable easy transfer from one project to another), and it needs to meet certain formal criteria (to ensure data persistence and congruency). At present, there are no stringent guidelines or standard approaches to building learner corpora.

In this paper, we present an abstract and generic workflow and propose to use it as a standardised way to build corpora for written language. This workflow is suitable for the processing of analogue data (e.g. hand-written text) as well as digitised data and applicable to the creation of learner corpora. It has been established with users' needs and technical feasibility in mind. The paper is structured in the following way: In section 2, we characterise the users' requirements before the workflow is presented and explained in section 3. In Section 4, we describe how the workflow has been applied to our learner corpus of German ("KoKo") that originally motivated its development. In Section 5, we evaluate the impact of our workflow.

2 Research on a Learner Corpus - the Linguists' Needs

Learner corpora are mainly created and employed in linguistic research, with their main user group being linguists. In the following, we describe the users' requirements for building a learner corpus from two perspectives: the corpus and the procedure for building the corpus.

2.1 Requirements related to the Corpus

2.1.1 Extensible Corpus Annotations

When building a learner corpus, different annotation levels can be added in separate processing phases. The corpus might initially be annotated with information about the visual appearance of the documents, such as graphical arrangement (header, paragraphs, emphasis, etc.) and self-corrections (insertions, deletions). Then, information about deviations from the standard written variety of the language (e.g. orthographical and morphosyntactic errors) can be added. Also, non-contiguous annotations might be added, for example anaphoric relations. To ensure extensibility of the corpus annotations, different processing phases need to add annotation levels in a well-structured and systematic fashion.

2.1.2 High-Quality Corpus Annotations

When using learner corpora to investigate learner language, high-quality corpus annotations as a basis for precise analyses are of paramount importance. This is already true for each annotation level given the subtle differences that may set apart individuals and groups, but it is even more important for subsequent annotation levels because errors may escalate and thereby artificially augment differences between individuals and groups. Thus, it is important to minimise the number of annotation errors on each individual level.

2.1.3 Searchable Corpus

In order to perform research on learner corpora, linguists need to perform practical searches and analyses on the corpora. With larger corpora, performing non-automated analyses becomes a time-consuming, labour-intensive and error-prone activity. Additionally, with many research questions at hand, researchers need to be able to test their ideas and compute statistics in an easy and dynamic way. Thus, it is important that the corpus can be searched via sophisticated queries and that statistics can be computed on the result sets, taking into account the different annotation levels.

2.2 Requirements Concerning the Corpus Building Procedure

2.2.1 Efficient Procedure for Manual Work

Among the necessary resources for building a learner corpus, human effort is both the most important and usually the scarcest one. As such, any means to enhance a manual task or to avoid that a manual task needs to be performed repeatedly has a noticeable impact on the size of the final corpus and its annotation levels, and thus, on the validity of the arguments derived from it. Therefore, procedures for building corpora have to integrate components, including manual work, in an efficient way.

2.2.2 Dynamically Evaluable and Adaptable Procedure

While building corpora, it is helpful to monitor the quality and quantity of transcriptions and annotations in order to identify problems early on; for example, if inter-annotator agreement is low on a specific annotation level, the procedure should allow to identify the situation and amend it, or in case of early ample annotations a premature termination of the annotation task should be possible. Since such issues are difficult to predict in advance, the procedure should facilitate the regular evaluation of the corpus as well as the correction and extension of all types of annotations if necessary.

2.2.3 Formalised and Reproducible Procedure

An abstract procedure should yield a blueprint for building learner corpora, i.e. the procedure should be formalised in a way that (major) decisions are highlighted, so that identical objectives and design decisions for a corpus ensure identical results. In addition, the result of the building process should be

reproducible by others, provided that they have knowledge of the objectives and design decisions, and access to the intermediate data.

3 Workflow for Building Learner Corpora

3.1 Abstract Workflow – Blueprint

In Figure 1, the abstract workflow is designed as comprising seven components, of which each can rely on one or several tools.

Component (1) covers the process of converting analogue data into a digital representation. Component (2) addresses manual annotation tasks, and component (3) refers to annotation with support from human language technology (HLT) tools. Component (4) enables the linguist to explore the corpus and search for specific elements in the cotext while component (5) relates to the computation of general numerical values in the corpus. Component (6) handles conversions of the corpus (or parts of the corpus) between different source and target formats. Component (7) describes the encoding of the corpus in an exchangeable format that accommodates any type of annotation provided by the other components.

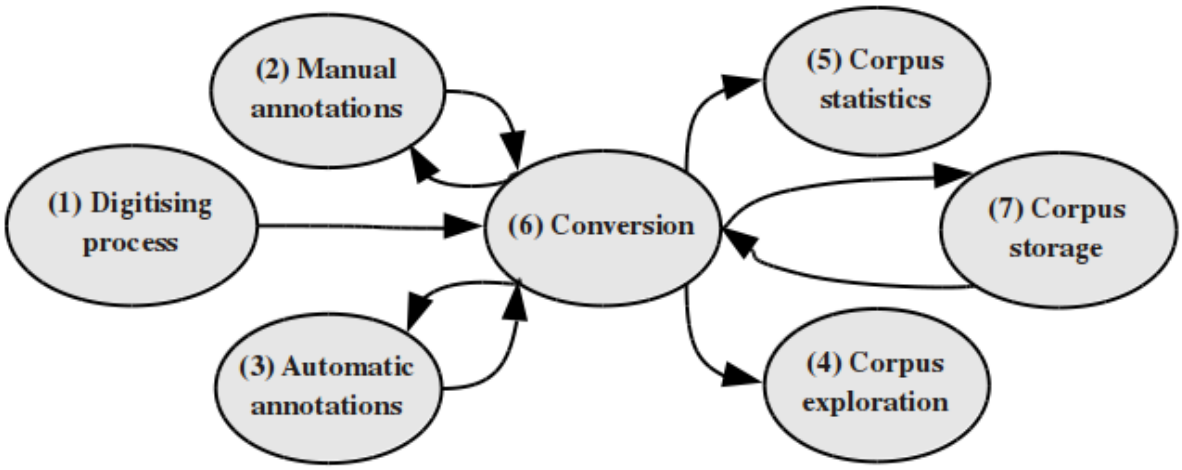

Figure 1. Abstract workflow for corpus building

Finally, we added an all-encompassing, optional tracking system that is of practical relevance: the change-log system. Its purpose is to track all relevant changes - the evolution - of both the corpus and the tools implementing the workflow.

3.2 How does the abstract workflow relate to user requirements?

3.2.1 Extensible Corpus Annotations

The annotation components (2, 3) should dynamically integrate additional annotation levels. Then, the extensibility of corpus annotations is supported by two further components: the conversion component (6) and the data storage component (7); both components should be able to deal with all data from the digitising process (1) and the annotation components (2, 3).

3.2.2 High-Quality Corpus Annotations

The workflow lays the foundations for high-quality corpus annotations by means of a well-defined data flow between components, and the possibility to repeat data processing steps to improve overall data quality. In combination with the dynamically evaluable and adaptable procedure (see section 3.2.5) this ensures that pre-defined quality criteria for the annotations can be met.

3.2.3 Searchable Corpus

The searchability of the corpus is covered by the components for corpus exploration (4) and corpus statistics (5). Again, component (6) ensures interoperability between these two components and the data from the digitising process (1) and the annotation components (2, 3).

3.2.4 Efficient Procedure for Manual Work

The efficiency of the entire procedure for manual work is related to the implementation of two components: the manual annotations component (2) and the digitising component (1) in case it includes human interaction. Furthermore, in order to employ manual work efficiently, a workflow should avoid unnecessary manual work by promptly detecting any issue in the performed annotations. To this effect, both the corpus exploration (4) and the corpus statistics component (5) should give users the possibility to browse the corpus for transcription and annotation errors and to implement methods and tests to detect such errors.

Semi-automatic annotations efficiently combine computational (3) with human resources (1, 2) and reduce human effort. Finally, the optional change-log system can recover earlier versions of annotations and thus helps to resume work quickly.

3.2.5 Dynamically Evaluable and Adaptable Procedure

The components for corpus exploration (4) and corpus statistics (5) are used to perform quantitative and qualitative analyses. Then, in case of failing to adhere to predefined quality and quantity criteria, annotations can be adapted with the help of the annotation components (2, 3). The conversion (6) and data storage component (7) facilitate this dynamic exchange process.

3.2.6 Formalised and Reproducible Procedure

The very existence of the abstract workflow is a formalisation of the procedure; adhering to the workflow increases comprehensibility and reproducibility of the work and the obtained results. Additionally, the change-log system realises reproducibility on the data level, i.e. versioning control.

4 Application of the Corpus Building Workflow in the KoKo Project

The KoKo project[1] is part of 'Korpus Südtirol' (cf. Abel & Anstein 2011; Anstein et al. 2011) - a corpus linguistic initiative to collect, file and process South Tyrolean texts in order to make them available to the public and to document the use of written German in South Tyrol.[2]

4.1 Research Focus

As part of the 'Korpus Südtirol' initiative, KoKo focuses on L1 learner texts. The declared aim of the project is to investigate and describe the writing skills of German-speaking secondary-school pupils at the end of their school career by analysing authentic texts produced in classrooms. The corpus building process was guided by two linguistic goals, namely (1) to describe writing skills at the transition from secondary school to university, and (2) to determine external factors that influence the distribution of writing skills, such as sociolinguistic (gender, age), socio-economic, and language-related biographical factors (L1, preferred variety of German, reading and writing habits, etc.). In addition, KoKo aims at employing a corpus-based methodology in order to facilitate the analysis of the learner texts.[3]

4.2 Design and Method of Data Collection

1511 pupils from 85 classes and 66 schools participated in the project by writing a text and providing information about their background. The pupils are from three different German-speaking areas: North Tyrol (Austria), South Tyrol (Italy), and Thuringia (Germany). At the time of data production, all writers attended secondary schools one year before their school-leaving examinations. In detail, 89% of them were between 17 and 19 years old, 7% were older than 19 years, and 4% did not specify their age. Classes were sampled randomly using the size of the cities in which the schools were located (small *vs.* medium *vs.* big) and the type of school (providing general education *vs.* education specific to a particular profession) as strata for the sampling. All texts were collected in May 2011.

The text production was integrated in the regular course work of the participating classes and consisted of a written in-class assignment. The pupils' task was to write an argumentative essay on the same predetermined topic. In addition, all participants completed a written survey with sociolinguistic, socio-economic and language-related biographic questions. Identities of the participants were anonymised but remained associable to each essay with its corresponding survey data.

4.3 Linguistic and Technical Differences between L1, L2/FL Learner Corpora

We follow Abel and Glaznieks (in press) and refer to people as L1 learners, when they are still in the process of learning their L1 or at least substantial parts of it, such as writing and text production. Prototypically, instances of L1 learner language can be found in the educational and academic context. From a linguistic point of view, the texts of language learners writing in their L1 are likely to have many features of non-standard writing in common with those

written by L2/FL learners. However, these situations still relate to separate learner varieties due to the fact that some features are specific to either L1 or L2/FL learners. From the perspective of computational processing, L1 or L2/FL learner corpora are equivalent as they are both compilations of textual data that deviate from a standard variety. Indeed, the technical requirements regarding all components are all identical. We therefore believe that the results we obtained from our L1 corpus are of interest for evaluating the relevance of the workflow of learner corpora in general.

4.4 Implementation of the Abstract Workflow for KoKo

In the following section, we describe the way in which we have implemented the abstract workflow for KoKo along with the way in which we intend to improve this implementation.

4.4.1 Digitising Process and On-the-Fly Annotation (Inline) - XMLmind XML Editor

The digitising process of handwritten documents, i.e. the transcription, uses XMLmind, a strictly validating, near WYSIWYG, XML editor, which can be used to create documents conforming to a custom schema.[4] During the transcription, the corpus was manually annotated with surface features of the text, such as graphical arrangement (header, paragraphs, emphasis, etc.) and self-corrections (insertions, deletions). Specific deviations (orthographical errors, uncommon abbreviations) from the standard written variety of German were given a target hypothesis (cf. Lüdeling et al. 2005) on a separate level; for example, for the token "bereuhen" with an orthographic error, an annotation providing the corrected token "bereuen" (engl. 'to regret') has been added. This ensures an annotation level that provides an error-free version of the corpus, which can be used to search for canonical word forms, and it improves the accuracy of the POS-tagging (see section 5.2); the latter can be reduced in a remarkable way by a high number of misspelled and non-recognisable words (cf. Schmid 1995: 8). Also, emoticons and symbols were annotated. All annotations were done on-the-fly, in an environment very similar to a word processor. Using a known environment significantly facilitated the transcribers' tasks.

The main shortcomings of XMLmind are general limitations of XML inline-annotations relating to the cumbersome or impossible annotation of crossing hierarchies and of discontinued constituents, as well as problematic handling of multiple annotations of the same layer.

4.4.2 Manual Annotation (Stand-Off) - MMAX2

To be able to perform linguistic analyses, elaborated annotations are being added in sequentially dependent and independent phases. They concern new lexical and grammatical annotations as well as annotations for phenomena on the text level. These types of annotations demand a stand-off annotation tool. MMAX2 is a tool for annotating text in a stand-off format that allows for multi-layered annotations. It is well suited for annotating linguistic elements at the level of the text and allows for the definition of customised annotation schemes. It also provides useful means to customise displays and the user interaction (cf. Müller & Strube 2006).

4.4.3 Automatic Annotation – TreeTagger

We are interested in tokenisation, sentence splitting, POS-tagging and lemmatisation. The TreeTagger is a tool for annotating text with POS and lemma information, and includes tokenisation and sentence splitting as pre-processing steps (cf. Schmidt 1994); it supports German and many other languages.

4.4.4 Corpus Exploration and Corpus Statistics - CQP and ANNIS2

In an earlier stage, we relied on the IMS Open Corpus Workbench's flexible and efficient query processor CQP (cf. Christ 1994), its application programming interface (API) and its front end to perform both corpus exploration and compute corpus statistics. However, we have recently decided to migrate our previous work, and use ANNIS2 (cf. Zeldes et al. 2009) from now on. As explained on its website[5], "ANNIS2 is an open source, versatile web browser-based search and visualization architecture for complex multi-level linguistic corpora with diverse types of annotation". Since information structure interacts with linguistic phenomena on many levels, ANNIS2 addresses the need to visualise annotations covering various linguistic levels such as syntax, semantics, morphology, prosody, referentiality, lexis and more. It also provides means to build highly elaborated queries.

4.4.5 Conversion – SaltNPepper

As explained on its website[6], SaltNPepper (cf. Zipser & Romary 2010) is an Open Source project developed to tackle an important issue in HLT research: there is a range of formats and no unified way of processing these. This issue derives from the fact that many expert tools for annotating and interpreting linguistic data have been developed for very specific purposes. In order to fill that gap, a metamodel called Salt, which abstracts over linguistic data and a pluggable universal converter framework called Pepper have been designed and implemented. It currently handles PAULA, MMAX2 and a large variety of other formats[7], where the Pepper module for MMAX2 was developed in the course of this project.

4.4.6 Data Storage - MMAX2 and Paula

The KoKo corpus is currently stored in MMAX2 format. However, our data storage component will be migrated to Paula XML format[8] (cf. Dipper et al. 2007) which, just like for MMAX2, is a stand-off XML format. Nevertheless, Paula takes into account more recent technical developments and has originally been designed to be an exchange format for linguistic content. As such, it is able to represent a wider range of annotations more efficiently.

4.4.7 Change-Log - Subversion (SVN)

Theoretically, many versioning systems could be used to implement the change-log system. Being widely adopted, SVN[9] still[10] appears to be a good means for managing changes to documents and tools. Indeed, several SVN clients are available on major operating systems (Windows, OS X, Linux), and the HTTP transport layer can use well-established proxies and thus be integrated into

corporate security configurations. Last but not least, some of the clients enable point-and-click interaction with the data, which is an important feature for computer laymen.

4.4.8 Dynamic Evaluation and Adaptation

Although manually checking each annotation is very time-consuming and labour-intensive, it yields high data quality. Voormann and Gut (2008) noted that errors in the corpus creation process can occur throughout all processing stages. Inspired by their Agile Corpus Creation approach we iteratively performed quality checks during the digitising process and within the annotation components. The correction phases dealt with transcription errors and those errors within the on-the-fly annotations that affect the automatic annotation of the data. We focused on words missing from the HLT processing tool, i.e. those that were assigned the placeholder 'unknown' on the lemma level. Mostly, the transcription of the words was erroneous (human error during transcription) or the target layer contained a 'new' word. At the end of the correction phase, a high-quality corpus (i.e. ideally, with no transcription errors and all orthographic errors annotated) with satisfying automatic processing results (i.e. for tokenisation and sentence splitting, no unknown lemmas, and acceptable accuracy of the POS tagger) was ready to use (see section 5).

4.5 The KoKo Corpus

From 1511 pupils, 1503 essays along with the corresponding written surveys (see section 4.2) were used for the corpus; the essays were manually transcribed and on-the-fly annotated (see section 4.4.1), and automatically processed (see section 4.4.3).

The corpus (version KoKo 2, Dec. 2012, cf. Abel et al. 2014) consists of 1503 texts, with a total amount of 46,734 sentences and 811,330 tokens (with punctuation: 930,241 tokens) with POS and lemma information. Almost 90% (1,319 texts, 716,405 tokens) of all texts were written by writers with L1 German (cf. Table 1). Several metadata information is available in the KoKo corpus coming from the questionnaire survey. Apart from the writers' L1, their type of school, their grade at school, their region of origin, and their gender information can all be flexibly used to create sub-corpora. Therefore, the corpus can be used to analyse and compare simple text features such as text length, sentence length, lexical variation, etc. In addition, the corpus can also be used for more sophisticated statistical analyses of data from different annotation levels. The results of these analyses can be related to the metadata and can be analysed for significant correlations. We intend to make the corpus publicly available at the end of 2014, and make it accessible via ANNIS3[11].

Table 1. Example of sub-corpora within the KoKo corpus using 'region' and 'L1' as a filter.

	total		L1 German	
Sub-corpus (region)	*tokens*	*texts*	*tokens*	*texts*
North Tyrol	233,098	457	206,439	404
South Tyrol	222,209	520	192,891	451
Thuringia	353,674	521	317,075	464
not defined for region	2,349	5	---	---
total	**811,330**	**1503**	**716,405**	**1,319**

5 Evaluation of the Corpus Quality

In the following, we present an evaluation of the quality of the corpus. The quality of the current corpus is evaluated first; then, the impact of the target hypothesis and the manual adaptation on the automatic annotation component is evaluated. This is done on the basis of changes from one processing stage to another.

5.1 Evaluation of the Quality of the KoKo Corpus

To evaluate the quality of the corpus, we measured the number of errors on different annotation levels. We randomly selected a sample of 255 sentences (0.54%) from the entire corpus (46,734 sentences). All 255 sentences of the sample were evaluated with respect to (1) transcription errors, (2) errors in the annotation of orthographic errors, (3) tokenisation errors and (4) sentence splitting errors. Twenty-eight sentences were excluded from the subsequent evaluation because they influence the POS-tagging and thereby the evaluation result. They either showed transcription errors (seventeen sentences), flaws in manual annotation of the orthographic level (nine sentences), or errors in tokenisation (one sentence) and sentence splitting (eight sentences). The remaining 227 sentences were evaluated with respect to (5) POS-tagging errors. The mentioned aspects (1–5) influence the usability of the corpus and were considered indicative of the overall quality of the corpus.

In order to evaluate the POS-tagging results, we created a data set of the desired output, our gold standard, and calculated the percentage of identically assigned tags between the output of the POS tagger and the gold standard. The percentage indicates the accuracy of the POS tagger. We executed the following several successive steps to produce the gold standard: (a) The automatic POS-tagging output was independently checked by two annotators, but as expected, their POS tags were still containing disagreeing information (*silver standard 1* and 2)[12]. (b) A third person annotated the first 20 sentences of the sample for POS from scratch (*silver standard 3*). (c) Results of step (a) and (b) were compared on the overlapping parts, and differences were discussed until a consensus was reached. This result was then taken as the gold standard for the first 20 sentences of the sample. Analysis of the individual silver standards revealed that the annotators in (a) were biased towards the result of the automatic pre-annotation whereas the third annotator in (b) did not perform significantly better but made different errors. Both results are in line with Fort and Sagot

(2010). (d) For the remaining 207 sentences, a new silver standard was created using an ensemble of three POS taggers. In addition to the TreeTagger, we used the Berkley Parser (Klein & Manning 2001) and the Stanford POS Tagger (Toutanova et al. 2003) for the ensemble. Evaluation on the first twenty sentences revealed a very high accuracy (>97%) in case of complete agreement. In addition, the evaluation revealed systematic errors for some well-known deficiencies of POS taggers for German (e.g. interchanging a finite and a non-finite verb, a noun and a proper noun, cf. Schmid 1995:7–8 and end of this chapter). Except for these deficiencies, the POS tags of the ensemble were used for *silver standard 4* in case of a complete agreement of the ensemble (~86%). In all other cases, the third annotator tagged from scratch. This procedure reduced the workload while still leading to high quality results. (e) Results of the *silver standards 1*, *2* and *4* for the 207 sentences were compared and all differences were discussed. The consensus on the silver standards together with the agreements on the discussions, then, constituted the gold standard for the rest of the sample. This process also revealed errors in tokenisation and sentence splitting. Table 2 shows the result of this evaluation process.

Table 2. Evaluation of the quality of the corpus (KoKo 2).

level	total size		correct		accuracy in %	
	token	sentence	token	sentence	token	sentence
(1) transcription	4,842	255	4,825	238	99.65	93.33
(2) orthographic errors	61	49	49	40	80.33	86.96
(3) tokenisation	4,842	255	4,841	254	99.98	99.60
(4) sentence splitting	---	255	---	247	---	96.86
(5) POS-tagging	4,191	227	3,969	96	94.70	42.29

Accuracy in the evaluated dimensions (1–5) varies: transcription accuracy (1) is fairly high and has reached an accuracy rate of more than 99.6%. With respect to orthographic error annotation (2), the accuracy rate of around 80% was lower than expected. Still, we are not aware of any numbers for comparison. However, annotation accuracy of orthographic errors cannot be proved, and this aspect should be considered when a new version of the corpus is prepared. Regarding the automatic processing, only one tokenisation error (3) remained in the sample, five sentences were wrongly split and three did not get split at all (4). The POS-tagging accuracy of 94.70% (5) is on the lower end of the state-of-the-art POS-tagging performance for German (up to 97%, cf. Schmid 1995: 6–7). Considering the remaining grammatical errors, the token level accuracy is still excellent, but it should be noted that only 43% of the sentences are free of tagging errors. A reason for this result may be related to inherent deficits of the tagger.

The TreeTagger has several well-documented language specific shortcomings (Schmid 1995:7–8). For example, the German version of the TreeTagger produces erroneous tags for homographic finite and infinite verb forms (e.g. *sagen*: infinitive, 1st/3rd person plural, 'to say' vs. '(we/you) say') that are assigned with different tags (VVINF vs. VVFIN). In addition, some errors occur also for sentence adverbs (ADV) that are homographic with adjectives (ADJD). Finally, relative pronouns (PRELS) are sometimes conflated with definite articles (ART)

when the obligatory comma that introduces subordinate clauses is missing. All these and other phenomena that are vulnerable to shortcomings of the tagger on average appear once per sentence. This causes the errors to be equally spread over the sample. The shortcomings of the tagger obviously influence the quality of the automatic annotations. Unfortunately, they cannot easily be fixed by automatic intervention or adjustment of the annotation tool.

5.2 Evaluation of the Corpus during Adaptation Phases

The individual corpus adaptation phases are evaluated on the basis of the number of errors on the lemma and POS level. As mentioned in section 4.4.1, orthographic errors as well as uncommon abbreviations were annotated, and target words in standard spelling were added. This led to a revised version (KoKo 1.2) of the original corpus (KoKo 1.1). All 7774 out-of-vocabulary (OOV)[13] tokens in KoKo 1.2 were checked, the correct lemma information was added to the HLT tool's lexicon and in case of a POS error the correct information was also added; this improved lexicon was used to create a revised version of the corpus (KoKo 2) with 892 OOV tokens.

All 227 sentences (see section 5.1) were evaluated in the three corpus versions: the POS information with respect to our gold standard and the lemma information with respect to a reduction in the number of OOV tokens. The 227 sentences (4191tokens) were further subdivided into two sets (1) comprising only those 34 sentences with target level annotations (718 tokens) and (2) the 193 remaining ones (3473 tokens).

Table 3 shows the corresponding sample of KoKo 1.1, i.e. the corpus without added target words which contained 1.3% OOV tokens (54 out of 4191 tokens). Adding target words improved this number for KoKo 1.2 to 0.6% of unknown tokens (25 out of 4191 tokens); as expected, this only affects subdivision 1 of the sample. KoKo 2 then, with the improved lexicon for the processing, contained no OOV tokens in our sample.

Table 3. Evaluation of the corpus during adaptation phases regarding lemmatisation.

corpus	size		OOV lemmas in % (absolute numbers in brackets)					
			KoKo 1.1		KoKo 1.2		KoKo 2	
	token	sentence	token	sentence	token	sentence	token	sentence
sample	4,191	227	1.29 (54)	17.62 (40)	0.60 (25)	8.81 (20)	0	0
subdivision 1	718	34	5.15 (37)	70.59 (24)	1.39 (10)	17.65 (6)	0	0
subdivision 2	3,473	193	0.49 (17)	8.29 (16)	0.43 (15)	7.25 (14)	0	0

Table 4 shows that the POS-tagging accuracy (token level) in the corresponding sample of KoKo 1.1 was almost 94%. The improvement of the whole sample is moderate but particularly noticeable for the subdivision 1: from 90.81% to 94.43%. McNemar's Chi-squared tests with Yates's continuity correction (R Development Core Team 2011) demonstrated that there were significant (df = 1, $p < 0.01$) improvements regarding the whole sample (χ^2(KoKo 1.1 vs. KoKo 1.2) = 22.32; χ^2(KoKo 1.1 vs. KoKo 2) = 26.28) as well as subdivision 1 (χ^2(KoKo 1.1 vs.

KoKo 1.2/ KoKo 2) = 22.32). The improved accuracy on the sentence level in subdivision 1 is even more noticeable, from 11.76% to 35.29%, and this illustrates the necessity of providing target words for the tagger. Again, subdivision 2 is not affected by this processing step. KoKo 2 did not improve significantly on the POS level, i.e. the POS tagger was able to assign correct tags even without lemma information.

Table 4. Evaluation of the corpus during adaptation phases regarding POS-tagging.

corpus	size		accuracy in % (absolute numbers in brackets)					
			KoKo 1.1		KoKo 1.2		KoKo 2	
	token	sentence	token	sentence	token	sentence	token	sentence
sample	4,191	227	93.99 (3,939)	37.89 (86)	94.61 (3,965)	41.41 (94)	94.70 (3,969)	42.29 (96)
subdivision 1	718	34	90.81 (652)	11.76 (4)	94.43 (678)	35.29 (12)	94.43 (678)	35.29 (12)
subdivision 2	3,473	193	94.64 (3,287)	42.49 (82)	94.64 (3,287)	42.49 (82)	94.76 (3,291)	43.52 (84)

5.3 Summary of the Evaluation

We have shown that the L1 learner corpus KoKo is of high quality with respect to transcription, tokenisation and sentence splitting, and of satisfactory quality regarding POS-tagging accuracy. The quality of the corpus results from our dynamic workflow in which we first add target words during the transcription phase and then use the automatic annotations to highlight errors of the HLT tools. After that, we adapt the data, re-iterate the processing, and evaluate the results. The data also shows quite clearly that accuracy of manually annotating orthographic errors has not reached such a high standard. A consequence of missing error annotations is the lack of a corresponding target word that can be used by the POS tagger as the correct basis for annotation. Therefore, the more accurate the manual annotations are performed, the better the results on POS-tagging will be.

The evaluation of the corpus during the individual adaptation phases revealed that providing target words indeed improved the accuracy of the POS tagger and the lemmatiser. The accuracy of the POS tagger improved considerably on the sentence level, and, in sentences containing orthographic errors and uncommon abbreviations, it almost reached the same accuracy throughout the rest of the corpus. In addition, the target layer reduced the number of cases to be considered for the lexicon by about a half and, therefore, the workload for further manual intervention. However, adding OOV lemmas to the lexicon has only a minor effect on POS-tagging accuracy. Planned annotations on the grammatical level are expected to further increase the POS-tagging accuracy (Rehbein et al. 2012: 7).

6 Conclusion

In this paper, we have introduced a standardised procedure for building learner corpora. The procedure is based on actual needs as identified by linguists. For explaining the procedure, we sketched an abstract workflow that can be used as a blueprint for building new learner corpora. We illustrated each component of the workflow using examples from an on-going project, in which we compiled an L1 learner corpus. In this project, the workflow has proven useful and led to a high-quality corpus. The dynamically evaluable and adaptable procedure was successfully employed, and the corpus annotations could be extended in different phases. However, during the creation of the workflow, we encountered some detours and hit a few dead-ends; hence, for some components the improvement in efficiency is still to be shown. Also, only some analyses have been done: the corpus exploration and statistics component could easily deliver the necessary data but larger and more complex analyses still have to be carried out. Our experience with setting up a project of corpus creation together with external partners has shown that employing the workflow presented in this paper helps to use the available human resources efficiently, increases the transparency of the creation process, and thus supports the generation of a high-quality corpus.

Endnotes

[1] The full title of the project is: "Bildungssprache im Vergleich: *ko*rpusunterstützte Analyse der Sprach*ko*mpetenzen bei Lernenden im deutschen Sprachraum (unter besonderer Berücksichtigung des Deutschen in Südtirol) / Comparing 'Bildungssprache': analysis of the language competence of - especially South Tyrolean - German L1 learners on the basis of corpora"; the acronym "KoKo" derives from the German words for corpus and competence.

[2] www.korpus-suedtirol.it

[3] Besides the creation of a learner corpus, a further project goal is to develop tools that assist the linguists' research work (*cf.* Anstein 2013).

[4] http://www.xmlmind.com/xmleditor/ (last accessed on 18 April 2013).

[5] http://www.sfb632.uni-potsdam.de/annis/ (last accessed on 18 April 2013).

[6] https://korpling.german.hu-berlin.de/p/projects/saltnpepper/wiki/ (last accessed on 18 April 2013).

[7] CoNLL, DOT, EXMARaLDA, FALKO, GrAF, RelANNIS, RST, Tiger, TreeTagger, TueBaDZ, *etc.*

[8] http://www.sfb632.uni-potsdam.de/en/paula.html (last accessed on 18 April 2013).

[9] http://subversion.tigris.org/ (last accessed on 18 April 2013).

[10] https://www.youtube.com/watch?feature=player_detailpage&v=4XpnKHJAok8#t=189s (last accessed on 18 April 2013).

[11] http://korpling.german.hu-berlin.de/falko-suche

[12] Given our awareness of a persisting imperfection of the data, we call it "silver" rather than "gold".

[13] With respect to the automatic lemmatisation by the TreeTagger.

References

Andersen Abel, A. & S. Anstein 2011. Korpus Südtirol - Varietätenlinguistische Untersuchungen. In A. Abel & R. Zanin (eds.), *Korpora in Lehre und Forschung.* Bozen-Bolzano: University Press, 29–54.

Abel, A. & A. Glaznieks in press. Wo Sprachkompetenzforschung auf Varietätenlinguistik trifft: Empirische Befunde aus dem Varietäten-Lernerkorpus "KoKo". In A. Lenz & M. Glauninger (eds.), *Variation und Varietäten des Deutschen in Österreich - Theoretische und Empirische Perspektiven.* Frankfurt: Peter Lang.

Abel, A., A. Glaznieks, L. Nicolas & E. Stemle 2014. KoKo: An L1 Learner Corpus for German. *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014),* Reykjavik, 26–31 May, 2014, 2414-2421.

Anstein, S. 2013. *Computational Approaches to the Comparison of Regional Variety Corpora - Prototyping a Semi-automatic System for German.* Doctoral dissertation, University of Stuttgart, Stuttgart.

Anstein, S., M. Oberhammer & S. Petrakis 2011. Korpus Südtirol - Aufbau und Abfrage. In A. Abel & R. Zanin (eds.), *Korpora in Lehre und Forschung.* Bozen-Bolzano: University Press, 15-28.

Christ, O. 1994. A Modular and Flexible Architecture for an Integrated Corpus Query System. *Proceedings of COMPLEX 1994,* Budapest, 7–10 July, 1994, 23–32.

Dipper, S., M. Götze, U. Küssner & M. Stede 2007. Representing and Querying Standoff XML. In G. Rehm, A. Witt & L. Lemnitzer (eds.), *Data Structures for Linguistic Resources and Applications. Proceedings of the Biennial GLDV Conference 2007*, Tübingen, 11–13 April, 2007, 337–346.

Fort, K. & B. Sagot 2010. Influence of Pre-Annotation on POS-Tagged Corpus Development. *Proceedings of the Fourth Linguistic Annotation Workshop*, 56–63. Uppsala, Sweden: Association for Computational Linguistics. [Retrieved November 27, 2013]. Available at http://dl.acm.org/citation.cfm?id=1868720.1868727.

Granger, S. 2003. Error-tagged Learner Corpora and CALL: A Promising Synergy. *CALICO Journal* 20 (3), 465–480.

Granger, S. 2008. Learner corpora. In A. Lüdeling & M. Kytö (eds.), *Corpus Linguistics. An International Handbook,* Vol. 1. Berlin: de Gruyter, 259-275.

Hana, J., A. Rosen, S. Škodová & B. Štindlová 2010. Error-tagged Learner Corpus of Czech. *Proceedings of the Fourth Linguistic Annotation Workshop (LAW IV),* Uppsala, 15–16 July, 2010, 11–19. [Retrieved April 2, 2013] Available at http://dl.acm.org/citation.cfm?id=1868722&CFID=304754737&CFTOKEN=66453585.

Hana, J., A. Rosen, B. Štindlová & P. Jäger 2012. Building a learner corpus. *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012),* Istanbul, 21–27 May, 2012, 3228–3232.

Klein, D. & C. Manning 2001. An O (n^3) Agenda-Based Chart Parser for Arbitrary Probabilistic Context-Free Grammars. Technical Report. Stanford. [Retrieved November 29, 2013]. Available at http://ilpubs.stanford.edu:8090/491/1/2001-16.pdf

Lüdeling, A., M., Walter, E. Kroymann & P. Adolphs 2005. Multi-level Error Annotation in Learner Corpora. *Proceedings from the Corpus Linguistics Conference Series* 1 (1). [Retrieved April 2, 2013]. Available at http://www.birmingham.ac.uk/research/activity/corpus/publications/conference-archives/2005-conf-e-journal.aspx

Müller, C. & M. Strube 2006. Multi-Level Annotation of Linguistic Data with MMAX2. In S. Braun, K. Kohn & J. Mukherjee (eds.), *Corpus Technology and Language Pedagogy. New Resources, New Tools, New Methods*. Frankfurt: Peter Lang, 197–214.

Nesselhauf, N. 2004. *Collocations in a Learner Corpus.* Amsterdam: John Benjamins.

Rehbein, I., H. Hirschmann, A. Lüdeling & M. Reznicek 2012. Better tags give better trees – or do they? *Linguistic Issues in Language Technology – LiLT* 7 (10), 1–20.

R Development Core Team 2011. *R: A language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing.

Reznicek, M., A. Lüdeling & H. Hirschmann 2013. Competing Target Hypotheses in the Falko Corpus: A Flexible Multi-Layer Corpus Architecture. In A. Díaz-Negrillo, N. Ballier & P. Thompson, Paul (eds.), *Automatic Treatment and Analysis of Learner Corpus Data.* Amsterdam: John Benjamins, 101–124.

Schmid, H. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, 14–16 September, 1994. [Retrieved October 14, 2013]. Available at http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tree-tagger1.pdf

Schmid, H. 1995. Improvements In Part-of-Speech Tagging With an Application To German. *Proceedings of the ACL SIGDAT-Workshop*. Dublin, 1995. [Retrieved October 14, 2013]. Available at http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/tree-tagger2.pdf

Toutanova, K., D. Klein, C. Manning & Y. Singer 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. *Proceedings of HLT-NAACL 2003*, 252–259. [Retrieved November 29, 2013]. Available at http://nlp.stanford.edu/downloads/tagger.shtml

Voormann, H. & U. Gut, 2008. Agile Corpus Creation. *Corpus Linguistics and Linguistic Theory* 4 (2), 235–251.

Zeldes, A., A. Lüdeling, J. Ritz & C. Chiarcos 2009. ANNIS: A search tool for multi-layer annotated corpora. *Proceedings of Corpus Linguistics 2009,* Liverpool, 20-23 July, 2009. [Retrieved October 14, 2013]. Available at http://ucrel.lancs.ac.uk/publications/cl2009/358_FullPaper.doc

Zinsmeister, H. & M. Breckle 2012. The ALeSKo learner corpus: design – annotation – quantitative analyses. In T. Schmidt & K. Wörner (eds.), *Multilingual Corpora and Multilingual Corpus Analysis*. Amsterdam: John Benjamins, 71–96.

Zipser, F. & L. Romary 2010. A model oriented approach to the mapping of annotation formats using standards. *Proceedings of the Workshop on Language Resource and Language Technology Standards (LREC 2010)*, Valetta, 17–23 May, 2010, 7–18.

Received November 30, 2013
Revision received July 14, 2014
Accepted November 25, 2014