The ICE Nigeria corpus project: Creating an open, rich and accurate corpus

Eva-Maria Wunder, University of Augsburg
Holger Voormann, Agilantis Holger Voormann
Ulrike Gut, University of Augsburg

1 Introduction

This paper reports on the corpus creation process of the ICE Nigeria, which began in October 2007 at the University of Augsburg. In the collection of the ICE Nigeria, we intend to achieve the goal of creating a richly and accurately annotated open corpus and to do this with maximum efficiency – with the least possible investment in time and cost. The corpus creation process of the ICE Nigeria is closely modelled on the theory of agile corpus creation (Voormann and Gut 2008). Like some other theories of corpus creation (e.g. Biber 1993; Atkins *et al*. 1992), the theory of agile corpus creation proposes that the corpus compilation should proceed as a cyclic process, in which repeated searches of an initially small corpus provide guidelines for further corpus annotation. In other words, it claims that the time and effort of corpus creation can be minimized by the adoption of a query-driven approach that consists of iterations of the cycle 'corpus querying – annotation schema development – corpus annotation – corpus analysis'. One of its fundamental tenets holds that corpus creation is inherently error-prone and should be open to modifications and improvements at every stage. This claim means that during the corpus creation process little upfront annotation and continuous testing of the annotation accuracy should be carried out. The process of creating the ICE Nigeria is illustrated in the following sections. Section 2 describes the cyclic corpus creation process and the annotation of the written and the spoken part. Efforts that are made to increase annotation accuracy are presented in Section 3. Section 4 shows what steps are taken to make the ICE Nigeria an openly available corpus. A summary and a description of further planned activities are given in Section 5.

2 Creating the ICE Nigeria corpus

The creation of the ICE Nigeria is carried out with Pacx (www.pacx.sf.net), a platform for annotated corpora in XML that is being developed especially for the ICE Nigeria project but is available for other corpus projects too. For example, it is currently also used in the compilation of the ICE Malta and the ICE Bahamas. The Pacx application extends the Eclipse platform (www.eclipse.org) by a set of tools, so-called plug-ins. In particular, it comprises the XML editor Vex, the image viewer QuickImage and Subversive, which is a client for the version control system Subversion that supports collaborative work (see Section 2.3). The advantage of creating a corpus with such an integrated application compared to a loose bunch of separate tools is that the different applications can be used at any time without the need to save current data and open or reload it in another application. In the compilation of the ICE Nigeria, the software ELAN (www.lat-mpi.eu/tools/elan), which is unfortunately not (yet) available as an Eclipse plug-in, is used to annotate audio and video files and has to be installed separately.

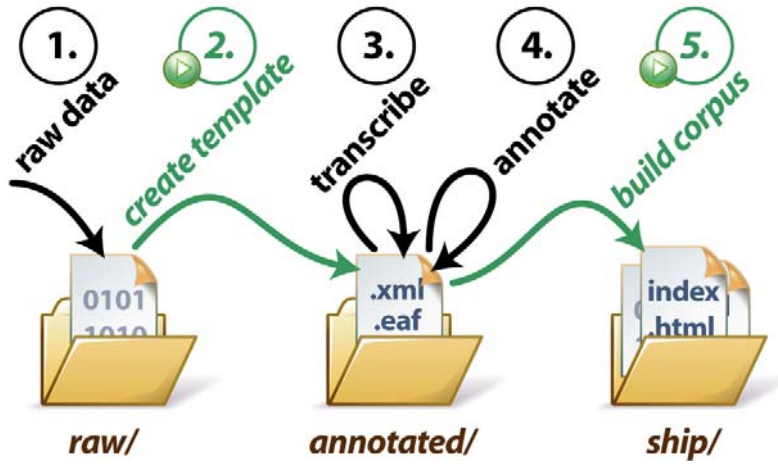

Figure 1: The five stages of corpus creation with Pacx

Figure 1 shows how the corpus creation process with Pacx proceeds. First, the raw data (the audio or video files, scanned handwritten texts, PDF files, text files and HTML files) is put into the raw data folder, which is divided into separate folders for the different text categories specified by the ICE project. By

running the Create Templates Script, a transcription template is created for each raw data file. These templates consist of Vex (.xml) files for the written and ELAN (.eaf) files (also an XML format) for the spoken data. In the third step, the data is transcribed, i.e. the raw data is transferred onto the template. This can take on different forms for the written data in the corpus (see Section 2.1): for example, the plain text of an HTML, PDF or other rich text file is copied and pasted straight onto the template, whereas the content of a scanned handwritten text is first recognized by OCR and afterwards copied onto the template. (The transcription of the spoken data is described in Section 2.2). The corpus creation process then proceeds with the annotation. For this, the transcriber marks the word that is to be annotated and selects a tag from a predefined list, as can be seen in Figure 2. In the last step, an automatic Build Script creates a summary page in HTML format, on which all the data currently contained in the corpus is listed together with the metadata, links to the raw data, the transcription in plain text and the transcription in XML. Furthermore, the total number of words of each file and the corpus as a whole are displayed there.

This process ensures that, from the transcription of the very first raw data file onward, a shippable (i.e. distributable) corpus is always available. Moreover, the Build Script includes automatic analyses, which are run every time a new corpus is being created. This allows a query-driven approach postulated by agile corpus creation (Voormann and Gut 2008). Currently, only one query is carried out for the ICE Nigeria: the analysis of all words annotated as ‘errors’. These are automatically counted and displayed next to the correction entered by the transcriber.

2.1 Annotation of the written part

The annotation of the raw data in the written part is based on the automatically created .xml template. First, the template allows the transcriber to enter the following metadata: place and time of writing of the raw file; transcriber; gender, age and ethnic group of the author/s. Furthermore, the template offers a placeholder for a heading and a placeholder for a paragraph, onto which the text from the raw data file can be copied. For the transcription of the text, the transcriber marks the word or phrase that is to be annotated and selects the appropriate tag (e.g. ‘italics’) from a predefined list, as can be seen in Figure 2:

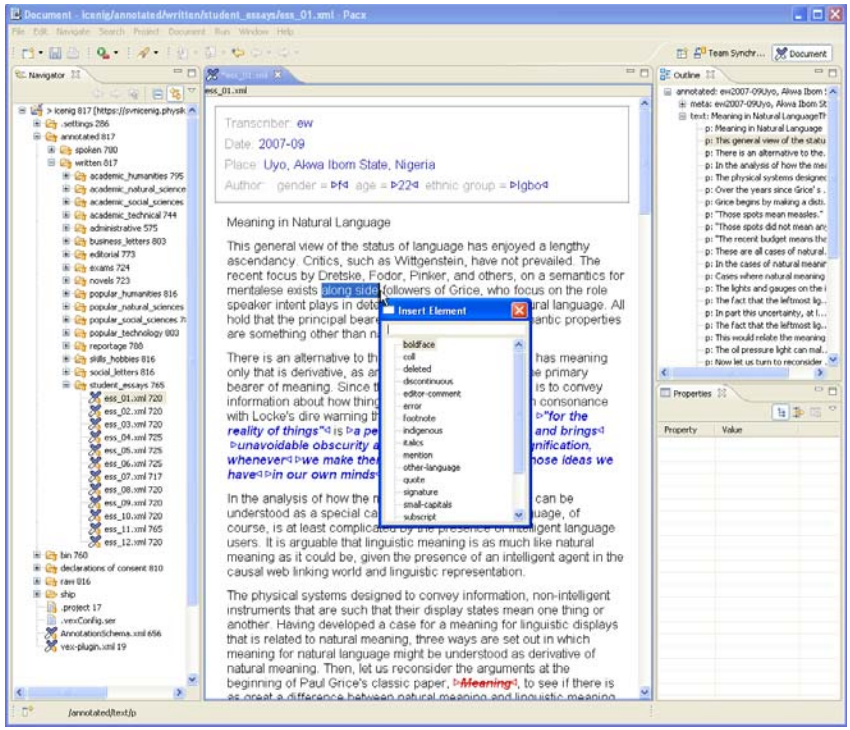

Figure 2: Annotation of the written part. The editor in the middle shows the template, in which the metadata and text have been entered. The pop-up window offers the list of available tags for annotation.

In contrast to the traditional approach with which earlier ICE corpora were created, annotation with Pacx is much simpler: it does not require the transcriber to type in any of the SGML tags that are specified in the Markup Manual for Written Texts (Nelson 2002). Instead, with Pacx (or to be precise, the integrated XML editor Vex) markup proceeds by selecting the relevant text passage and choosing the annotation label. Different markups are visualised with different text formats: for example, a word annotated as an error will appear in red and with a strike-through line (see the word ‘Meaning’ in Figure 2). Pacx saves these annotations as a well-formed XML document.

2.2 Time-aligned annotation of the spoken part

One of the key innovative features of the spoken part of the ICE Nigeria is its time-aligned transcription and annotation. Figure 3 illustrates the time-aligned transcription of a video recording in the ICE Nigeria that is carried out with the software ELAN. The video is displayed in the top left-hand corner. Beneath the speech waveform, the transcription is represented on three different levels, referred to as 'tiers'. Text-to-tone alignment links the annotation with the raw data. The boundaries of all transcribed elements are defined by time stamps, which means that information about the exact beginning and end of each element is available in the corresponding file – an XML-based file format – that is created by the programme. Automatic corpus analysis tools working on this file can thus calculate phonetic properties such as the mean length of pauses and the average number of words per interpausal unit by a speaker. Moreover, the time-aligned annotation illustrated in Figure 3 provides direct access from each transcribed element to the primary data, i.e. the original audio or video file. By clicking on any transcribed element, the corresponding part of the recording will be played back by ELAN. This is especially useful for the annotation of the corpus because items in question can be listened to repeatedly.

Annotation of spoken data in the ICE Nigeria is carried out on three tiers. On the top tier, the beginning and end of each utterance, defined in the project as an interpausal stretch of speech, is marked. ELAN supports the division of speech into interpausal units with an inbuilt automatic pause detector, which marks all stretches of speech. These units of speech are then transcribed orthographically. On the middle tier, annotations of individual words are carried out. For example, they can be marked as unclear or indigenous words, and anonymised words are indicated. On the bottom tier, transcribers can add values like 'Yoruba' to specify the type of indigenous language further. Analogous to annotating written files with Vex, the annotation of the spoken part is very simple because transcriber choices are restricted. ELAN allows the creation of a customized list of annotations in the form of a so-called Controlled Vocabulary, from which the transcriber chooses the correct annotation to insert on the relevant tier. Again, no SGML tags need to be typed in manually by the transcribers – ELAN saves the annotation as a well-formed XML file.

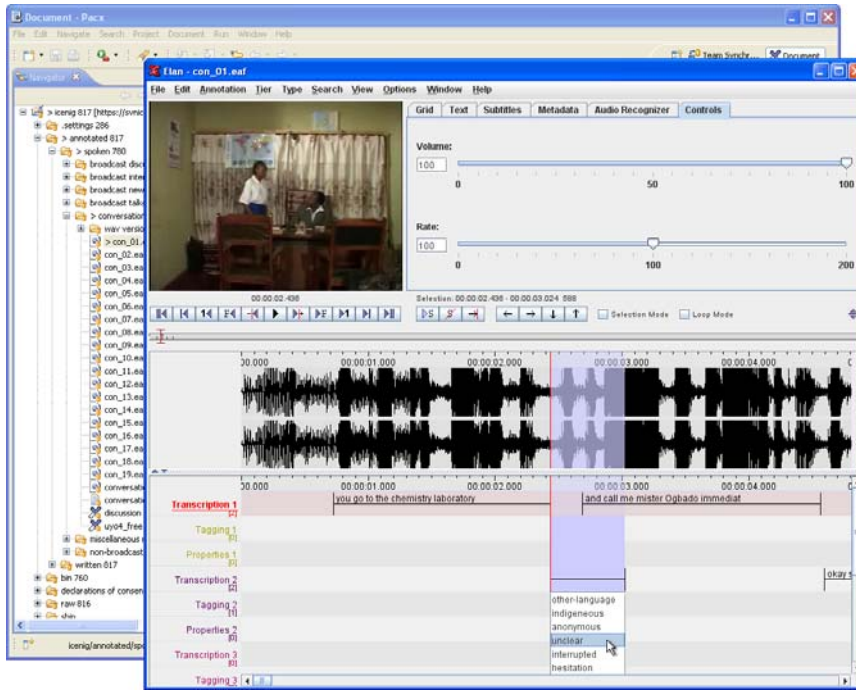

Figure 3: Annotation of a video with ELAN

2.3 Collaborative work

Pacx allows easy collaborative work with participants from all over the world, as shown in Figure 4. This is possible because all of the corpus data is stored on a central server, which provides version control. Each annotator has a local copy of the corpus data downloaded from the server on his or her computer. Changes to the corpus data are carried out and saved locally and can, in a separate step, be transferred to the central server. The annotators stay synchronized by regularly updating to the latest version of the corpus data from the server. This means that annotators can make changes to the corpus without being connected to the server and only require internet access when they want to commit changes or update to the latest version on the server.

Figure 4 illustrates this process in detail. For example, user A finishes creating a new template or editing a file and uploads these local changes to the server by choosing the *Commit* command. As a safety net, Pacx offers a list of

all locally changed files on which users can tick which files to upload. This avoids conflicts like the unintentional overwriting of alterations by another annotator in the same file. The latest version of a file is then stored on the central server. Metaphorically speaking, it is put on top of a stack of earlier versions of the same file without overwriting them. The 'change history' of each file is also stored on the central server.

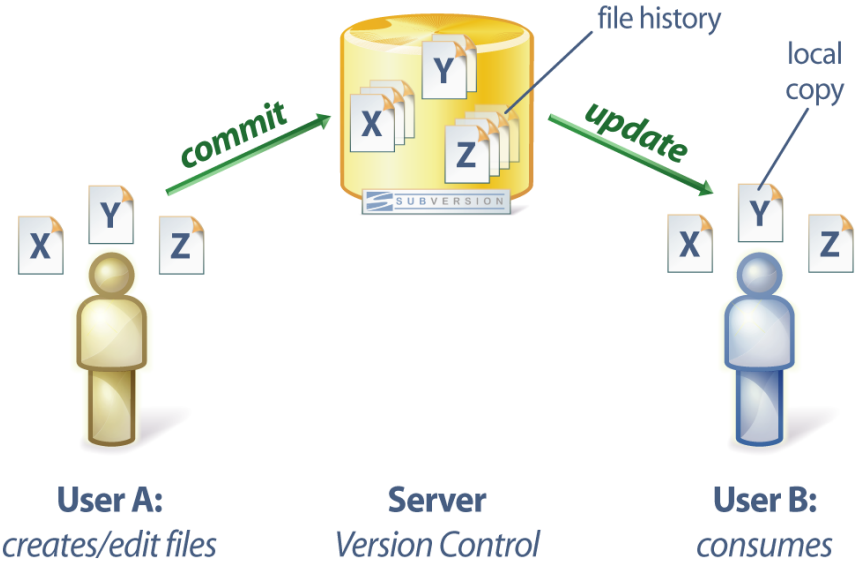

Figure 4: Collaborative work with Pacx

When user B begins a new annotation session, he or she uses the *Update* function of Pacx to get a local copy of the latest version of the corpus from the server. This version now includes the changes user A has previously committed. By the same token, changes made and committed by user B will form part of user A's local copy of the corpus after an update. In addition, a Wiki exists for the ICE Nigeria, in which all decisions regarding the annotation procedure and instructions on how to use Pacx are added. This ensures that transcribers joining the project later will have easy access to both the history and current state of the corpus creation process.

Pacx thus ensures maximally loss-free collaborative work: regular updates minimise the risk that users A and B, for instance, accidentally annotate the

same file simultaneously or work on older versions of a file. If a user is no longer certain, for example, whether he or she committed his or her last changes to the server, he or she can use the *Synchronize* function. In the *Team Synchronize* perspective, all incoming and outgoing changes that have occurred since the last update are made visible, as well as potential conflicts that might have arisen. To solve these conflicts, a user can either discard his or her local changes by overwriting the local files with the newest version of the corpus or choose to commit the local files as a newer version. Moreover, users also have the possibility to merge two conflicting files manually with the compare editor, which shows the local and the remote file side by side with the diverging sections highlighted. Even if a user commits something by mistake, the version control system offers a safety net in saving the history of each file and allowing the recovery of older versions.

3 Achieving a high level of accuracy in the annotations

One of the principal goals of creating the ICE Nigeria is to achieve a high level of accuracy of the annotations. Despite efforts to automatise some of the annotations in the ICE project (e.g. POS tagging), most of the annotations in the compilation of the ICE corpora are carried out manually. In order to ensure a high accuracy of these annotations, regular tests of their quality and the evaluation of the annotator reliability are required. Previous research has identified several factors that influence the quality of manual corpus annotations (see e.g. Gut and Bayerl 2004 for phonological annotation; Bayerl *et al.* 2003 for semantic annotation and Jovanovic *et al.* 2005 for annotation of dialogue acts and gaze direction). These include the complexity of the annotation task, lack of consensus on the annotation schema across the transcribers and transcriber characteristics such as familiarity with the material, amount of training, motivation, interest and fatigue-induced errors.

Several steps have been taken in the creation of the ICE Nigeria in order to minimize these effects. Incorrect SGML tags are the greatest source of errors in the existing ICE corpora and are immensely time-consuming to correct. As mentioned in Sections 2.2 and 2.3, with Pacx only valid – according to the specified Document Type Definition (DTD) – XML files can be created. Since transcribers do not type in the tags manually but choose annotation tags from a predefined list, syntax errors are not possible in the creation of the ICE Nigeria.

Another feature ensuring a high annotation quality that is integrated into Pacx is the function of specifying anti-patterns in the Build Script. An anti-pattern is an occurrence or sequence of annotations that is not allowed, for

example a pause followed by a pause in the spoken part of the ICE Nigeria or any other symbol than ‘f’, ‘m’ and ‘?’ for the gender of the speaker/author in the metadata. Whenever the Build Script is run in order to create the latest version of the corpus, an automatic analysis of anti-patterns is carried out and violations are marked in the summary page. Similarly, the DTD specifies and ensures the valid structure of documents, e.g. by not allowing that a phrase is annotated as a paragraph and a heading at the same time.

Moreover, any errors that inevitably occur in the corpus creation process, be it an accidental deletion of a file or the choice of a wrong annotation, are easily revertible and quickly undone with Pacx. The integrated compare editor allows the character-by-character comparison of two files and highlights all differences between them. Changes thus become visible at a glance and it can be decided which of the two is the correct file. Individual lines from one file can also easily be integrated into the other file. As Section 2.3 showed, accidental overwriting of files can be reverted with the *History* function, which allows one to enter an older version as the latest one.

4 Creating an open corpus

The ICE Nigeria is an open corpus in various ways. First, all parts of the corpus, including all raw data files in the written and spoken part, are available to the research community. This is possible because a declaration of consent allowing the distribution of the data was signed by every speaker or author who contributed data to the corpus. Having an XML-based format, the corpus is easily extensible and reusable by other researchers who might wish to add annotations or even data. Being published under a Creative Commons Plus license means that these changes have to be published under the same license.

Second, all tools used in the corpus creation process are open source tools. This includes Pacx and its components Eclipse, Vex, Subversion etc. as well as all Build Scripts and analysis scripts contained in Pacx.

Third, the creation process of the ICE Nigeria is open and documented on www.pacx.sf.net. There, for example, a video tutorial is available that shows how to create a corpus from scratch in three minutes. Collaboration in the further development of Pacx is possible and welcome there.

5 Conclusion and outlook

To sum up, with the creation of the ICE Nigeria we intend to collect a richly and accurately annotated open corpus. The development of the software Pacx allows

us to do this in a maximally efficient way. Annotation of the raw data is very quick, since the time-consuming manual typing of the specified ICE tags is replaced by an automatic and 100 per cent error-free creation of valid XML files by Pacx. Moreover, the error potential of manual annotations is further reduced by giving transcribers predefined lists of annotation tags to choose from and by integrating automatic searches for anti-patterns in the corpus Build Script. The annotation of the ICE Nigeria is the richest one in the ICE corpora family by including time-aligned transcriptions of the spoken data. This, together with the open design of the corpus, furthermore opens up new possibilities in research: so far, comparative phonological research on the world-wide varieties of English represented in the ICE corpora has not been possible. Only two of the ICE corpora collected so far (ICE GB and ICE Jamaica) provide access to the original recordings for the spoken part and neither of them have time-aligned transcriptions of the spoken data. The ICE Nigeria is the first to offer this and will thus contribute to a more in-depth exploration of the phonologies of English varieties.

Several further developments of Pacx are planned. These include the addition of automatic POS tagging of all words. Moreover, it is envisaged to run the Build Script automatically whenever a file is saved and to compute only those files that have been changed. This will make the Build Script much faster. By the same token, more anti-patterns will be included in the Build Script. Last but not least, the possibilities for querying the corpus with Pacx will be vastly improved. In particular, querying will move from text level to domain-specific searches.

The ICE Nigeria currently (10th November 2009) comprises 325,956 words. The written part will be completed in early 2010.

References

Atkins, Sue, Jeremy Clear and Nick Ostler. 1992. Corpus design criteria. *Literary and Linguistic Computing* 7: 1–16.

Bayerl, Petra Saskia, Harald Lüngen, Ulrike Gut and Karsten Paul. 2003. Methodology for reliable schema development and evaluation of manual annotations. *Proceedings of International Conference on Knowledge Capture (IK-CAP)* [no pagination], Florida.

Biber, Douglas. 1993. Representativeness in corpus design. *Literary and Linguistic Computing* 8: 243–257.

Gut, Ulrike and Petra Saskia Bayerl. 2004. Measuring the reliability of manual annotations of speech corpora. *Proceedings of Speech Prosody 2004, Nara, Japan*, 565–568.

Jovanovic, Natasa, Riek op den Akker and Anton Nijholt. 2005. A corpus for studying addressing behavior in multi-party dialogues. In L. Dybkjaer and W. Minker (eds.). *Proceedings of 6th SIGdial Workshop on Discourse and Dialogue*, 107–116. Lisbon.

Nelson, Gerald. 2002. Markup manual for written texts. <http://ice-corpora.net/ice/written.doc>

Voormann, Holger and Ulrike Gut. 2008. Agile corpus creation. *Corpus Linguistics and Linguistic Theory* 4 (2): 235–251.