Glossing Technology in Paratext 7

Jon D. Riding and Gerrit J. van Steenbergen
jon.riding@biblesociety.org.uk; vansteenbergen@solcon.nl

Jon Riding leads the Linguistic Computing team at British & Foreign Bible Society and Gerrit van Steenbergen is a UBS Translation Consultant for the Africa Area.

1. A brief development history
During the late 1980s the first Anglicised version of the TEV, known in the U.K. as the Good News Bible (GNB), was the leading product in the BFBS range in the U.K. Many different editions were produced for different market segments. Following the success of the launch, concern grew that positioning the GNB as an accessible translation well-suited for children and non-mother-tongue English speakers was in danger of misrepresenting the scholarly basis of the translation. To counter this perception it was decided to publish two concordances to the GNB to encourage its use as a vehicle for study. One was to be a small concordance bound with the Bible and the other a large semi-exhaustive concordance which was published as a stand-alone product. The member of the BFBS editorial team given responsibility for these products was David Robinson. In addition to his skills in biblical and other ancient Eastern Mediterranean languages, Robinson was amongst the first to apply the developing discipline of mathematical linguistics to Bible translation. The concordance project provided an opportunity to experiment with computational linguistics, to improve the quality of the final product, and to shorten the time needed for the task. These early experiments identified semantic domains in the GNB by mapping them against other parallel texts and using the equivalences found to identify surface forms and their roots (technically known as lemmata) which were part of each domain.

1.1. Early models
After the concordances were made, work began on an even larger concordance to Y Beibl Cymraeg Newydd (BCN, the New Welsh Bible), giving further opportunity to develop the process. Originally named MECCA (MEchanised Closed Corpus Analysis), the processing was patented under the name MALACHI—Machine Analysis of Lemmata And Closed-Corpus Heuristic Indexing (Robinson 1991). Using this technique, the BFBS was able by the early 1990s to process most languages and accurately identify roots from the surface forms found within a particular domain. The core processing was already remarkably close to that which now ships with Paratext. At the time, however, personal computers (PCs) available in the field were simply not powerful enough to run it. At BFBS the processing was handled by a RISC-based Vax Alpha system, many times more powerful than typical PCs. At about the same time, IBM were running similar experiments in the Candide project using a Cray 2 super computer. It is testament to the robustness of the BFBS algorithm that good results were being obtained.
whilst the IBM project was closed as impractical. As the BFBS team processed more and more languages, particularly from sub-Saharan Africa, it became clear that the single biggest problem the process faced was that posed by complex morphologies. The original algorithm was, surprisingly, powerful enough to handle most of these complexities by treating different surface forms with a common root as synonyms present within that semantic domain in the target text. This approach was, however, very heavy on computing power and the process could not be transferred to PCs in the field.

1.2. First experiments in the field
Recognising the importance of Robinson’s pioneering work, the BFBS supported it by the creation of the Machine Assisted Translation (MAT) Team, now known as BFBS Linguistic Computing (LC). The team began adapting and improving the process to make it more suitable for PCs. The outcome of this work was the Augustus system, which was better able to handle complex morphologies. However, this capability came at the expense of limiting the available corpus of text with a consequent loss of precision. Nevertheless, the Augustus process was the starting point for the current Glosser. The key development introduced by Augustus was the recognition that it was possible to seek a semantic equivalent not just for complete words but for word segments. This breakthrough allowed Augustus to identify lemmata at an early stage of the process and establish semantic equivalents for lemmata rather than for surface forms from which lemmata were derived. Thus, whereas with MALACHI an attempt to gloss a set of English words such as \{love, loves, loved, loving\} against Swahili would have generated a long list of individually poorly attested “synonyms” all sharing the root *pend*, Augustus was able to identify the root *pend* and map this directly to lov* in relatively few operations. Once a root such as *pend* was identified it could be used to generate information about its associated morphology.

1.3. Standards and interoperability
Despite the progress, there remained a number of relatively trivial but hugely time-consuming problems. Texts received for processing might theoretically follow Standard Format Marker (SFm) coding standards, but whose? Each UBS region had its own set of SFMs, and variants within regions were common. Coupled with the confusions of legacy font encodings, these two problems made progress difficult. Two developments proved the key to resolving this situation:

**Unicode:** Throughout the 1990s the Unicode standard was being defined and by the year 2000 it had reached the point where most natural language was sufficiently represented. This offered real prospects of interoperability between different systems without the need to have specially designed encoding conversions.

**Unified SFM (USFM):** Second, the continuing development of Paratext resolved the problems of differing SFM codes by enforcing a single USFM standard and providing a mechanism by which other systems could access text held in Paratext. This development removed the need for text structure verification which had taken up so much time in the past.
It was now possible to write systems in the expectation that Paratext would present text as requested. The immediate outcome of this was a collaboration between Jon Riding from BFBS LC, who had developed the Augustus system, and Clayton Grassick, a member of the UBS Paratext development team. Grassick was well placed to rewrite the BFBS glossing engine, taking advantage of the facilities now offered by Paratext and Unicode and the power of modern PCs. The result of this collaboration was the Statistical Glossing Tool for Paratext and the Paratext Interlinear. These were subsequently developed into the Paratext 7 Glosser and Interlinearizer.

2. The Paratext Glosser

2.1. How it works

The Glosser is a true “black box.” A word is selected for glossing, text from the model translation* and the target text is fed into the process and, apparently by magic, results are returned. There is very little the user can do to affect the way the Glosser works. When the system was first demonstrated as the Statistical Glossing Tool (SGT) to a group of Translation Consultants they immediately named it the “Mystical Glossing Tool.” This paper hopes to dispel some of this mystery.

2.1.1. Comparison with other Machine Translation systems

Most Machine Translation (MT) systems are knowledge based. Information about languages and language pairs and examples of translations are encoded within databases which store the components of the languages and rules which govern the construction of words, clauses, and sentences. An inference engine is then written to retrieve the information and apply the rules to generate a solution to a given problem. The Glosser is quite different. A prerequisite for the system was complete language independence. This ruled out the use of supplied tables and databases, at least for target languages. Instead, the Glosser relies on its own algorithms to recognise terms in the target text which are equivalent to that selected in the model text (Riding 2008). Such an approach has only become practical as a consequence of the two important standardisations discussed above. The creation of the Unicode standard for encoding text and the development and implementation of the USFM standard for marking Scripture from Paratext 5 onward have been crucial developments. Without them it would have been much more difficult to include the glossing technology within Paratext.

2.1.2. The advantages of biblical text

To provide its analysis, the Glosser relies on the fact that biblical texts share an almost identical chapter and verse structure. Where different versifications exist between different traditions, Paratext is able to resolve these ambiguities and retrieve the verse that corresponds to the one requested. Using this alignment the Glosser is able to establish the best equivalent term in a target text for a word or lemma selected from a model text. The Glosser does this by finding all the verses in the model text containing the word or a form of the word to be glossed. A list of

---

* The term “model text” refers to the text which is going to be glossed (whether the vernacular or some standard translation such as the GNB) and “target text” refers to the text whose wording will be used to create the glosses. For a common use of the Glosser to produce a back-translation, the “model” will be the vernacular and the “target” might be the GNB.
the same verses is then read from the target text. From this a further list is made of all the words found in those verses in the target text and the number of times these words occur within that list of verses and within the text as a whole. In its simplest form all that is now required is to divide the occurrence count for a word in the target verse map by the occurrence count for the same word in the text as a whole. If the resulting value is 1.0, that is a strong indicator of equivalence. Lower scores indicate some lower probability of equivalence. In an ideal world where words in the target language map precisely to the words in the model, all glosses score 1.0 and all our questions are answered unambiguously. In reality the general fuzziness of semantics between language pairs is likely to result in a partial solution for a target word or lemma. There are a number of solutions to this particular problem. The current generation of the Glosser prefers to select the word in each verse which scores most highly as part of the overall solution.

2.2. Morphology and lemmatisation

2.2.1. Morphology

Processing languages which are close to one another is relatively straightforward. Indeed, selecting English to French in the Interlinearizer will typically generate an interlinear with a high degree of accuracy. The single biggest problem the Glosser faces in processing a language is the difficulty posed by complex morphologies. Languages with high inflection rates and complex morphologies pose far greater problems for the Glosser. It is not uncommon to find that the solution to a gloss will include many cognate surface forms which are unique in the context of the gloss. To deal with this level of confusion it is first necessary to attempt an analysis of the morphology of the target language. This initial analysis is purely automatic and does not at present require user input. The analysis enables the Glosser to recognise cognate surface forms of a lemma more easily. In effect, this analysis is analogous to a supplied database of morphology such as might be constructed using Shoebox or other similar tools. The difference is that the Shoebox database may be many weeks or months in the preparation. By contrast the Glosser’s morphological analysis is entirely automatic and is usually completed within a minute or so, depending on the speed of the processor.

2.2.2. Lemmatisation

It is tempting to conclude that the Glosser is generating a list of roots (lemmata) for the target together with the associated morpheme structures found for those roots. Whilst this may sometimes seem to be the case it is not strictly true. What the Glosser is trying to do is identify the items (we will call them tokens) present in the target text stream which most closely correspond to the token in the model it has been asked to gloss. Occasionally, both the model and target tokens will be a word which is present in a single form in both texts. It is much more likely, however, that the equivalent token is in fact a root form which appears in a number of surface forms in the target text stream. The Glosser uses its morphological analysis to identify that portion of the various surface forms which is common to them all. This portion is sometimes called the “stem” of the word. In fact both of the terms “stem” and “lemma” are often used ambiguously and are commonly interpreted as either the shortest surface form or the citation
form. Strictly speaking, the Glosser identifies the longest continuous sequence of letters common to all surface forms, called a logical root. Thus in English a set of forms {love, loved, loveless, loveliness, lovely, lover, loving, lovingly} will generate a logical root lov*. For those who consider the stem of this paradigm to be love this may be counter-intuitive. Some analyses of English morphology will prefer to identify the stem as love, and then generate rules to describe the process which rewrites forms such as love+ing as loving. The Glosser does not do this. For its purposes it does not need to know these rewrite rules. The logical root lov* is perfectly adequate to allow the Glosser to identify the various forms of love present in an English text.

2.2.3. Problem areas
At present the Glosser will not give good results for languages where there is a lot of inflection using infixes within the root, such as the Semitic language family. Such non-concatenative morphologies, where complex forms are not simply the result of adding affixes before and after the root, pose real problems for the automatic morphology analyser built into the Glosser. Work is in hand towards a solution but it is at an early stage. Complex but largely concatenative morphologies are processed rather more successfully. Highly agglutinative morphologies may not perform well depending on the nature of the agglutination. Where the agglutination is purely morphemic the Glosser will generally perform well. Where, conversely, the agglutination creates complex words with more than one root, each perhaps with its own distinct morphology, the Glosser will experience more difficulty.

2.3. Counting beans
All that computers are able to do is count things. But they do it very well and very fast. Computer scientists sometimes describe computer processing as “counting beans” and fundamentally this is all computers do. This is particularly true of the Glosser. All it does is count beans. We usually talk about tokens or perhaps lexemes rather than beans but the process is identical. For the purposes of this discussion a token will typically be a word in a stream of text.

The problem of complex morphologies is one of a number of scenarios which may cause the Glosser problems. The other common scenario is that of languages which do not use spaces between words, such as Chinese and Tibetan. In these circumstances the Glosser will not perform well. In order to map the equivalence relationships between text streams, the Glosser must be able to identify the tokens in each stream which carry the meaning it seeks to map. Token identification, whether complicated by complex morphology or by the absence of inter-word spacing (or both), remains the greatest challenge for the glossing technology, and accounts for the majority of the problems users encounter.

3. Glossing technology and translation
What value does glossing technology have for translating? This question will be considered from two perspectives which are not mutually exclusive:
1. The translation officer/consultant (TO). This paragraph will explore the benefits of the technology for more accurate methods of translation checking.
2. The translator. This section will investigate the value of glossing technology in the task of the translator.
Before discussing these points it is good to note that there is an increasing
tendency to practice what is called distance checking (Pattemore & Wong 2005).
This method of checking is defined by them as “checking of translation by a TO
in the absence of the translator(s) or the reviewer(s).”

In view of recent developments, this definition of distance checking
requires modification. As internet connections become faster and more reliable,
translation checking through Skype is an increasingly realistic option. The TO and
translation team can look at the same text and discuss translation issues related
to that text without meeting face-to-face. The fact that budgets are becoming
tighter and travel expenses more prohibitive is another stimulus towards distance
checking, not necessarily in the sense of “checking in the absence of translator(s)
or reviewer(s),” but certainly in the sense of physical distance between the TO
and the translator(s) or reviewer(s). The glossing technology that is currently
applied in a number of tools in Paratext 7 makes it easier to practice some of the
checking options that are offered by modern technology.

This discussion is not intended as a manual for running the tools, but rather
as an attempt to point out the practical advantages to TOs and translators that
are offered by glossing technology. Paratext 7 offers a fairly adequate set of
contextualised helps for running the various tools in the program.

3.1. Glossing technology and the translation officer

One of the responsibilities of the TO in connection with an ongoing translation
project is that of quality control. Translation checking is most often done through
back-translation; the translated text is re-translated into a language that is well
understood by both the TO and the translator. Back-translation may be written or
oral. Both methods present their own set of difficulties.

For a written back-translation, the first question to be asked is: Who does
the back-translation? “Where possible, the back-translation should be made by
a mother-tongue speaker, someone other than the translator, so that it reflects
what the text means to a speaker of the language” (SIL 2009). Yet the value of
this exercise is limited by other factors. To what extent is the back-translator
influenced by existing Bible translations? How literal should the back-translation
be? To what degree is the back-translator able to maintain an acceptable level
of consistency in the process? In addition, the method is time-consuming and
expensive, while the degree of efficiency is low because the TO often restricts
himself/herself to spot-checking.

Oral back-translations do not necessarily offer a better alternative. They
are often made by one of the members of the translation team. The team
have, naturally, gone through the preparatory stages of the translation and are
fully aware of the exegetical and other issues at stake in the passage they are
considering. Moreover, familiarity with the content of the Bible can easily lead
to a back-translation that demonstrates the level of Bible knowledge of the back-
translator, rather than one that conveys a reliable impression of the meaning and
nuances of the translated text.

In both cases (written and oral back-translation) further clarification can
be sought through questions. However, asking the right questions is not as
straightforward as it may seem. It is not easy for the TO to avoid the pitfall of leading questions, often resulting in answers that one wants to hear, rather than answers that are justified by the contents of the translated text. This restriction, admittedly, applies to any form of translation checking.

The Paratext Interlinearizer has the potential to avoid a number of these problems. From the preceding paragraphs about the development of the tool it is clear that it becomes more reliable as more data are available. The main advantage of the Interlinearizer is its high level of objectivity. Based on mathematical calculations the tool proposes glosses without influence or interference from any of the participants in the checking process. The fact that the results can easily (and quickly) be edited during checking meetings on the basis of information provided by the translation team enhances the usefulness of the tool considerably. The new information that is fed into the program is processed again by the Interlinearizer and increases the accuracy of the back-translation further. The editing of the data that the tool proposes can be done at two different levels:

a. Lexical items of the receptor language: A click on the lexical item provides a number of options generated by the Interlinearizer (Figure 1). This varies from a proposed morphological break-down to possible combinations of lexical items. If none of the proposed options is correct the tool also offers the option to provide a customized morphology whereby prefixes and suffixes can be marked in relationship to the stem of the lexical item to which they can be attached.

![Figure 1. Editing the morphology of a lexical item in the receptor language](image)

b. Glosses: A click on the proposed gloss provides a list with the various glossing options. It also offers the possibility to enter a gloss that is not proposed by the program. This gloss is editable and can be deleted if necessary.

By editing and/or confirming the morphological data and/or the glosses, the tool increases in reliability as the project progresses. The newly edited information is applied immediately to all data available for that particular project. However, if edited and confirmed data do not suit a particular context, adjustment can still be made at that point.
It is also possible to gloss a phrase by highlighting a number of consecutive lexical items and choosing the option Gloss Phrase.

The predominantly word-for-word back-translation that is thus generated and gradually perfected in the process provides a solid basis for relevant translation checking questions, either in preparation for or during a checking meeting. The Interlinearizer provides the TO with a wide range of options for checking. In many instances these are language and culture specific, but some are of a more general nature. For example:

- The Interlinearizer provides the TO with help and guidance in understanding the translated text. Clearly the TO will require more details and background information but the back-translation is a good starting point for more pertinent questions.
- It enables the checking of consistency, both in terms of absolute consistency (lexical items that have to be translated consistently regardless of the context) and contextual consistency (identical or highly similar contexts may demand consistency in translation). The checking of consistency is best done in combination with other tools provided by Paratext 7 and largely depends on the proficiency of the TO with the entire spectrum of checking and other tools.
- The checking of key terms: For this essential element of translation checking the Interlinearizer can provide a starting point. The Biblical Terms tool (under the Tools menu) provides further assistance for more accurate and detailed checking of this important feature. See paragraph 3.2.1 for more details.
- Issues of markedness: Linguistic structures that differ in a marked way from the norm can somewhat easily be identified so that the significance and/or relevance of the deviation can be investigated in consultation with the translator(s). Markedness applies to grammar, morphology, and syntax. It can also be extended to issues of language typology. For example, if in a VSO language there is a variation in constituent order, such a variation would typically be the subject of discussion during a checking meeting.
- Other linguistic features both in the area of grammar (concord, tense/aspect, pronominal reference, etc.) and discourse (marking of units, cohesion, etc.).
- Figures of speech are naturally hard to handle for the Interlinearizer, but a literal back-translation is still helpful for the TO as a starting point for discussions with the translator(s).

3.2. Glossing technology and the translator

Two areas where glossing technology has been applied successfully for translators in Paratext 7 are the Biblical Terms and the Guess Translations options. Although technically the application is similar in these two cases it is useful for translators in different ways. These applications are almost equally useful for TOs. A third application with relevance for translators in connection with the Interlinearizer will also be considered briefly.

3.2.1. Biblical Terms

The Biblical Terms tool (in the Paratext 7 menu, Tools > Biblical Terms) provides an excellent opportunity to check and verify the translation of various biblical key terms. It can also serve the translator as a database for translation options.
The list can be filtered on the basis of different variables (under View), while it is also possible to increase the accuracy of the list by specifying prefixes and suffixes (under Tools). The renderings can be added and/or edited manually, while there is also the possibility to Guess Renderings (under Tools > Guess Renderings; see Figure 2). This is particularly helpful where the translation process has already reached a more advanced stage. This tool enables translators as well as TOs to carry out a thorough check of the translation of biblical key terms. The renderings can be cleared and/or edited by the translators. At the same time the list can also serve translators as a reliable way to compare present decisions in the ongoing translation process with renderings that were chosen at an earlier stage.

3.2.2. Guess Translations

The Guess Translations tool is probably the least well-known among the tools that are powered by the glossing technology. It largely works as the tool that was described in the previous paragraph, but the point of departure is different. From any given translation in Paratext 7 one can right-click on a lexical item and choose the option Guess Translations. The program will automatically ask the user to select a target text. It will then provide a complete list of glosses in the target text, based on statistical
data resulting from comparison of the two versions. The list also contains the number of verses in which each of the glosses is used (Figure 3). A click on the gloss immediately produces all references in a list window, which is navigable in Paratext. The list provides the gloss in bold typeface in the context where it is used. Double-clicking the reference causes all Paratext projects to scroll to that particular reference. It is possible to use the Greek or Hebrew text as starting point for the guesses. It is also possible to take an existing translation as a basis for the guesses.

This tool provides an excellent and exhaustive overview of all glosses used to translate specific lexical items. If the tool is unable to identify a gloss it will report “No translation found.” This functionality enables the translator to quickly verify or edit all glosses. So this tool provides both translators and TOs with the option to carry out quick and thorough checks of the translation of specific lexical items. On top of that, this feature presents the TO with a good handle on the semantic range of a lexical item.

Guess Translations is also a very helpful tool for translators in the process of drafting. It enables them to quickly look up the various glosses that were used in the translation up to that particular point and decide whether one of these options is also suitable in the present context, or even conclude that in earlier stages of the translation wrong decisions have been taken. In that case the translator has a complete list of all occurrences of that particular lexical item and can make the corrections without any omission.

3.2.3. Interlinearizer and revisions or adaptations

Apart from what is mentioned under 3.1, the Interlinearizer can also be used to create new translations on the basis of adaptations of closely related languages. For several reasons (that are beyond the scope of this paper) this functionality should be used with great care. Issues of cultural appropriateness, acceptability and ownership, textual basis, exegesis, and others need to be considered very carefully.

Areas where this functionality raises fewer questions are those of revision and orthographic adaptation. The exact terms for a revision naturally have to be agreed upon and spelled out clearly by all stakeholders at the beginning of a revision project. The same applies to orthographic adaptations.

Once these parameters have been set, the Interlinearizer provides a very adequate and efficient tool, both in terms of time and consistency, to enable the translator to work through the entire process of revision or adaptation. In order to make use of the Interlinearizer, a new (parallel) project has to be created, using the same versification as the original one. Both projects (the original one as well as the newly created project which will be the revision/adaptation) have to be loaded in the Interlinearizer. The original project will be the Text to Interlinearize, while the Model Text for Interlinearizer will contain the newly created project for the revision/adaptation. In addition, the settings under the Advanced tab need to be adjusted to Model and Project Languages Are Related and to Automatically export verses when approved (see Figure 4, next page). Finally, the Destination Text has to be indicated in the drop-down menu.
Once all this has been done the translator/editor works through the text and edits or corrects the text where necessary. Every time a verse is approved it will automatically be exported to the new project, while the program continues to make the adjustments and changes automatically to the new project. In the process the Interlinearizer “learns” the rules for adapting the old project to the new revision/adaptation, so that fewer adjustments will need to be made as the project progresses. The level of accuracy and speed can obviously not be matched by any manual type of revision/adaptation.

4. Summary
The glossing technology built into Paratext 7 offers TOs and translators a new, objective method by which some aspects of translations can be assessed. The processing is language independent. The greater the amount of completed text the better the results will be, so it is likely glossing will become more useful as a project progresses. People who have adopted this technology quickly are discovering further applications for the technology, including its use in dictionary building. It is already used to power the Concordance Builder.

Thanks are due to the Paratext team, in particular to Clayton Grassick of the Canadian Bible Society, for implementing and continuing to develop glossing within Paratext 7.

References