Document Authoring the Bible for Minority Language Translation

Stephen BEALE, Sergei NIRENBURG and Marjorie MCSHANE
University of Maryland, Baltimore County
1000 Hilltop Circle
Baltimore, MD 21015
{sbeale,sergei,marge}@umbc.edu

Tod ALLMAN
University of Texas, Arlington
701 S. Nedderman Drive
Arlington, TX 76019
todallman@sbcglobal.net

Abstract

This paper describes one approach to document authoring and natural language generation being pursued by the Summer Institute of Linguistics in cooperation with the University of Maryland, Baltimore County. We will describe the tools provided for document authoring, including a glimpse at the underlying controlled language and the semantic representation of the textual meaning. We will also introduce The Bible Translator's Assistant© (TBTA), which is used to elicit and enter target language data as well as perform the actual text generation process. We conclude with a discussion of the usefulness of this paradigm from a Bible translation perspective and suggest several ways in which this work will benefit the field of computational linguistics.

1 Introduction

And you thought Moses was a prolific writer! One of the largest tasks undertaken by modern man is to take Moses' collected works (along with the rest of the Bible) and translate them into the thousands of languages for which there is a need and a desire for such translation. Since 1942, Bible translation work has been completed in 611 languages (although usually only the New Testament). Work continues in 1678 language groups, and it is estimated that a total of 3000 additional translations will be required, representing more than 380 million people.[1] The goal is to begin work in each of these languages by the year 2025 – a massive undertaking by any measure.

The typical translation project utilizes a translator with three or more years of linguistics training, usually preceded by some amount of theological training. The budding linguist/ theologian-turned-translator then goes to the country of choice and spends a year learning the national language. Then it is off to the minority language region, where the translator lives out his or her dream by learning to live in different, often difficult conditions, in a different culture, while trying to learn a different language that usually has very little or no written tradition. The translator spends the first year or two adjusting, learning, and serving, teaching the native speaking translation team translation principles, and all the while struggling to get up to speed in the language.

Then the translation process begins. Accuracy in translation is stressed above all else, resulting in a process that is slow at best. Draft translations are checked numerous times, within the translation team as well as in the community. Back translations (from the target language translation back into English or the national language) must be created so that specially trained translation checkers can come and review the work. A typical New Testament project will last anywhere from ten to twenty years. Multiply that by 3000 languages and it is obvious that the Bible translation community has taken on a big goal.

Whatever you think of the task – whether you despise it, revere it or have made it your life's ambition – it is a task. A task which, by its very size and breadth, could bring a wealth of practical results to the field of Computational Linguistics. This paper describes one approach to document authoring and natural language generation being pursued by the Summer Institute of Linguistics in cooperation with the University of Maryland, Baltimore County. We will describe the tools provided for document authoring, including a glimpse at the underlying controlled language and the semantic representation of the textual meaning. We will also introduce The Bible Translator's Assistant© (TBTA), which is used to elicit and enter target language data as well as perform the actual text generation process. We conclude with a discussion of the usefulness of this paradigm from a Bible translation perspective and suggest several ways in which this work will benefit the field of computational linguistics.

[1] Statistics provided by Wycliffe Bible Translators (http://www.wbt.org/wbt-usa/TranGoal.htm)

2 Document Authoring

2.1 The analysis environment.

The first priority of a document authoring system must be to provide a convenient interface for authors to input text (see Figure 1). This text must subsequently be analyzed in such a way as to maximize the chances for quality translation into the target languages. We increase the probability of quality target language translations by first manually converting the Biblical text into the controlled English that we will describe in section 2.2. For translating the Bible into this controlled language, various versions of the Bible, including the original Greek and Hebrew, are consulted along with other exegetical helps. We also are preparing health care texts that cover issues relevant to a minority language setting.

and syntactic analysis are performed using a simplified version of the analysis system described in (Beale & Nirenburg 2003). Word sense disambiguation is currently accomplished simply by choosing the sense most frequently used for that word (in the Bible translation texts that have been analyzed before the current text) that is compatible with the currently displayed part of speech and the current syntactic analysis (each sense of a verb will have an associated set of allowed syntactic case frames). The case role of each syntactic constituent is displayed upon a mouse-over of the corresponding colored bar of the syntactic structure diagram.

The interface provides for easy editing of the results of each of the four types of analysis. The root word, part of speech and word sense can all be changed simply by clicking on the appropriate box

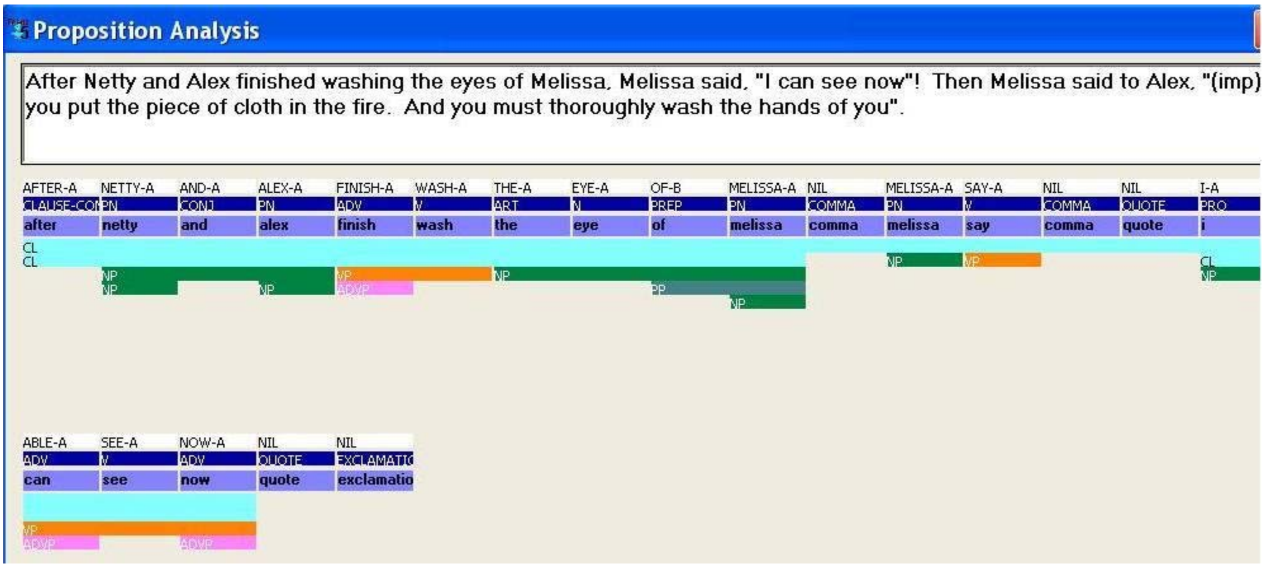

Figure 1: The machine-assisted semantic-analysis interface. This example, written in controlled English, is from the health care domain.

The next step is to create a correct and sufficient machine-tractable representation of the syntax and semantics of the text, which we accomplish using a semi-automated methodology. First an analysis program is run and the initial results are displayed. Each input word is analyzed as follows (from top-to-bottom in Figure 1):

- word sense (semantics)
- part of speech
- root/citation form
- syntactic dependencies (with case roles)

An English morphological analyzer is used to find the root form and part of speech of each input word. The syntactic analysis is visually displayed using colored bars. Part-of-speech disambiguation

and selecting a different choice. If necessary, a new root word and/or its associated part of speech or word sense can be added, although this is rare at this stage in the project. The case role of a syntactic constituent can also be easily changed with a mouse click, and a new case frame for a verbal word sense can be recorded. After any changes to the automatic selection of root word, part of speech or word sense, a new syntactic analysis is performed and redisplayed.

The syntactic analysis proposed by the system and represented by the colored bars can also be easily changed. The most common change concerns the location of attachment sites, especially for prepositional phrases, which default to the nearest possible attachment site unless the verb explicitly expects it in its case frame. Phrase attachments can be moved by clicking on the phrase and dragging it to a new attachment point. The starting and ending points of any constituent can be manually modified, and as a last measure, constituent boundaries can be deleted and new ones added.

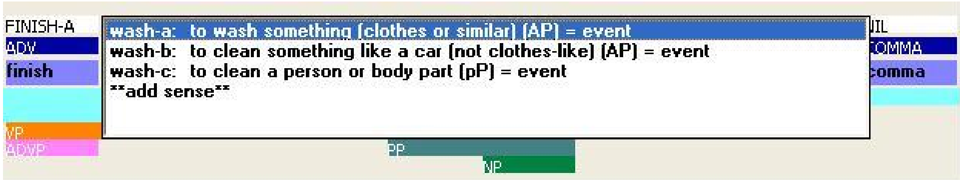

Figure 2: An example of semantic ambiguity.

Any changes or additions made by the user in this manner will be permanently stored in the analysis knowledge sources and will be used in the analysis of future texts.[2] The analysis system was created and subsequently augmented in this manner by a single user with experience both in Bible translation and computational linguistics. This user also has extensive experience with ontology and lexicon acquisition and syntactic analysis. This has eliminated many problems associated with more general document authoring systems, which are designed to be used by people who know little or nothing about the underlying knowledge and analyzer, and may only have an imperfect knowledge of the controlled language. We are able take this simplifying approach because the text corpus to be translated is known and finite, and could theoretically be analyzed by a single user. On the other hand, the corpus is very large, so we will be seeking to add one or two more analysts with similar qualifications in the next year. Currently, we have analyzed over 3000 verses and plan to complete all of the narrative sections of the Old Testament within the next two years.

In practical terms, after converting the texts to the controlled language, the user can have them automatically analyzed by the system with almost no need for post-editing of any morphological or syntactic analyses, except for PP attachments. The main task for the document author is to check that the word senses are correct. The user quickly learns which common polysemous words, such as "of" (see below), must be handled on a regular basis. Figure 2 shows the dialog box for manual sense disambiguation for the word "wash" as used in the example sentence in Figure 1.

On a global level, TBTA includes an interface that helps ensure the consistency of the semantic analyses. All occurrences of a given word sense can be examined to ensure that the meaning is uniform and that, for example, case frames are consistent. This is useful in two ways. First, while working on a particular text, the document author may need to clarify how a word sense has been used in the past. Secondly, we perform periodic reviews of the analyzed corpus as a whole. In addition, a major requirement of this project before it can be deployed for target language translation is that the resulting semantic representations must be thoroughly checked by trained specialists. We briefly discuss this issue in section 4 below.

The goal of this whole analysis process is to create Text Meaning Representations (TMR) which unambiguously encode the meaning of the Biblical text and which can be used as the input to the Text Generation process (described in section 3) to be used in each of the target languages. A TMR is made up of word senses (such as **wash-a** in Figure 2) connected by semantic relations. Various semantic attributes can also be attached to a word sense to modify its meaning. All of the word senses, relations and attributes are defined in our ontology (which was specially constructed for the Biblical domain). Although we were not able to discuss it above, a detailed analysis of time and aspect are also part of the semi-automatic analysis process.

2.2 The controlled language.

A few of the features of the controlled language we enforce can be seen in the text box in Figure 1 above:

- We do not allow possessive nouns (i.e. 's), but require the use of "of" ("the eyes of Melissa"). This is because we want to be able to specify the precise semantics of the relationship. There are 22 possible semantic relations from which the sense of "of" must be chosen for each occurrence (for example, ownership, kinship, made-of, etc.).
- The use of pronouns is allowed, but the document author is trained to use them only in cases where they are semantically unambiguous. With experience, we have learned that the target text's naturalness can be dramatically improved by specifying ahead of time which nouns can be safely referred to by pronouns. A conservative use of pronouns by the document author, with an eye trained to spot those situations that are semantically unambiguous, has proven valuable in this project. A mechanism for viewing (and changing) the analyzer's default linking of the pronoun to its antecedent is provided. In addition, the pronouns have word senses that

[2] Except that we currently do not automatically update the grammar based on changes to the syntactic bars.

distinguish them based on number (sing, dual, trial and pl) and exclusivity.

- Imperatives, yes-no questions and content questions are marked directly in the text by (imp), (yn-ques) or (ques). The actual sentence is then entered in its declarative form. For example, in the text box in Figure 1, notice that the subject “you” is included in the imperative. For content questions, an appropriate pronoun such as “who” or “where” is placed in the clause constituent that is being questioned.
- Every event is propositionalized.
- We eliminate most figurative language, except when it has theological implications. Specifically, we eliminate most instances of metonymy, synecdoche, euphemisms and idioms, and metaphors are converted to similes and the point of similarity is supplied.
- Other standard restrictions (like those described for the Kant system in Baker et al., 1994 and Mitamura, 1991) are employed, such as disallowing reduced relative clauses.

3 Target Language Text Generation

In this section we discuss the target language knowledge acquisition process along with a brief overview of the generation process. The knowledge acquisition interface and the text generator are integrated into The Bible Translator’s Assistant (TBTA). TBTA has been tested for English, Korean, Jula (spoken in West Africa) and Kewa (a clause chaining language spoken in Papua New Guinea). Korean, Jula and Kewa differ conceptually and structurally from English, yet in all cases the generated text has been well understood, grammatically perfect, and semantically equivalent to the original text, even before the post-editing process.

3.1 Target knowledge acquisition.

Each of the potentially thousands of target languages must have a target language grammar and lexicon developed. These target language knowledge sources will be used by the text generator to produce target translations from the semantic representation of the Biblical and health care texts. Below we briefly sketch some details of the knowledge acquisition process, which will take place under one of three situations:

1. A TBTA expert will work individually with a target language translation team.
2. A TBTA expert will lead a workshop for two or more translation teams.
3. A translation team will work by themselves, with consultation from a TBTA expert.

At this time, we have worked exclusively under the first situation. However, we plan to develop workshop materials so that we can quickly move into the second. Section 3.3 below describes the resources and tools that we have developed to make acquisition in situations one and two easier, and to make situation three possible.

The target language knowledge acquisition interface in TBTA was designed to be extremely flexible yet very easy to use. The knowledge sources required for generation consist of a target lexicon and grammar, both of which are used to map input semantic structures to target text. The target grammar contains two main sections: the first is used to restructure the semantic representations, or TMR (Text Meaning Representation) into appropriate target language structures, the second is then used to synthesize the proper surface forms. These processes will be briefly described next.

The first section of the target grammar performs **restructuring operations** on the TMR in order to change it into a new representation that is appropriate for the target language’s descriptive grammar (the second section). These operations include inserting new constituents, deleting constituents, moving constituents, copying constituents, and setting or copying features. This section of the grammar is responsible for all of the case frame adjustments, and it is used to generate grammatical relations from semantic roles, build clause chains with medial and final verbs, etc. After this first section of the grammar is executed, the input TMR will be transformed so that it contains a mix of purely semantic elements along with target language features, structures and some target words.

An example of a TMR restructuring rule for Korean is shown in Figure 3. Korean does not have a lexical equivalent appropriate for the concept PREVENT. However, by restructuring the proposition, the semantic equivalent can be formed. Consider the sentence *Mary prevented John from reading the book*. The Korean equivalent is *Because of Mary, John was unable to read the book*. A restructuring rule can perform the case frame adjustment for the event PREVENT to generate a new underlying proposition that is semantically equivalent to the original but is more suitable for Korean. The rule in Figure 3 shows the input propositional structure on the top. This input structure is purely semantic in nature. We use the tags NP and VP instead of something like OBJECT-PHRASE and EVENT-PHRASE for simplicity. In this case, the input semantic structure (which is unordered) expects an NP, a VP headed by the PREVENT concept (please note carefully that this is a concept, not an English word) and a C (a clause, or more accurately, a semantic proposition). Note that much of the internal structure of the constituent C is not specified; however, the VP and its head V is included because they are involved in the transformation. In general, any constituent or sub-constituent involved in the transformation should be listed in the input in its appropriate semantic relationship to the rest of the input; all other constituents and sub-constituents should be omitted[3]. The output structure of the rule is shown below. Several things happen in this output structure. The whole VP, along with the PREVENT concept, is deleted. A target language adposition (which basically means "because of") is added to the NP. The constituent C boundary is deleted and a verb particle which negates the embedded V is added. Note that the output of the restructuring rules can contain a mix of semantics and target language words or features. Both the input and output structures are unordered; only the semantic (and for the output structure, grammatical) relationships are specified. When the user first decides to create a rule related to the PREVENT concept, the visual grammar interface shown in Figure 3 will automatically present the standard case frame for PREVENT in the input structure and copy it to the output structure. The grammar writer can then make any necessary modification to the input structure, such as in this case, adding the sub-constituent VP and V in the C constituent. These changes to the input structure will be automatically copied to the output structure of the rule. At that point, the user makes the modifications to the output structure using the visual interface.

Several examples of restructuring rules for Jula (J), Kewa (K) and Korean (KO) are presented next. For simplicity, we give a rough English translation.

- J: X becomes sick -> illness happens to X
- X wears Y -> X is on Y's neck
- X leads Y -> X seizes Y's face
- K: X loves Y -> X sits happily with Y
- KO: obeys Y -> X hears Y's talk
- X is thirsty -> the throat of X is dry

[3] Unless the constituent or sub constituent must be present for the transformation to take place, even though it is not affected by the transformation.

The second section of the grammar is devoted to a more traditional **descriptive grammar** that will be used to produce and order the actual target language surface forms. A brief listing of the most important types of descriptive rules follows, with a short example or description for each.

- Feature copying rules - copying the number of the grammatical subject to the verb in English.
- "Spellout" rules.
  - Simple - add suffix for possessive noun.
  - Table - all the forms of "be" in English.
  - Morphophonemic - 'y' + 's' -> 'ies'
  - Form Selection - choose past tense of English verb under specified conditions.
- Phrase Structure rules - specify the correct surface ordering.

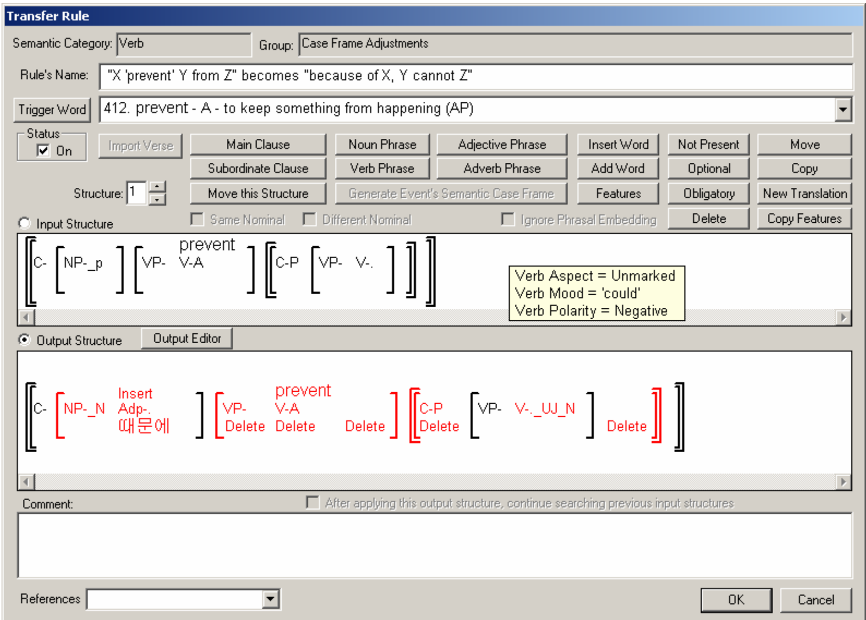

Figure 3: Restructuring Rule for Korean

The **target lexicon** is where all the target words will be listed, along with their basic mappings to word senses. For example, the Kewa word *tá* is mapped to the concept HIT. Unlike Mikrosmos (Beale et al, 1995) and OntoSem (Nirenburg & Raskin, 2004), where the semantic to syntax correspondence is recorded for every word sense in the lexicon, only the basic semantic mapping is listed in the TBTA lexicon. The standard mappings between case frames and surface structures are accomplished through descriptive grammar rules; exceptions to the standard mappings are handled by restructuring rules (which are linked to and accessible from the lexicon entry interface). The lexicon also contains a convenient interface for defining different word forms associated with the different target language parts of speech. For example, in Kewa, verbs can have gerund (*ti* for the root *tá*), habitual (*t*) and modified habitual (*tu*) surface forms. Rules for automatically generating each form from the root word are entered with a visual interface similar to the "spellout" rules of the descriptive grammar. Irregular forms can be entered when necessary. Features of words that are important in the target language can also be defined for each part of speech. For example, in English the count vs. non-count distinction is important for nouns.

3.2 The text generation environment.

The text generation environment is designed for easy debugging. In fact, target grammar and lexicon development is expected to be developed in an "acquire-debug" cycle. During the target language generation from an input semantic representation, the system keeps track of all the rules that participate in the generation of each particular constituent. After a short passage has been generated, the user can rest the cursor on each constituent to see which rules were involved in the synthesis of that particular constituent. If any of these rules were not functioning as expected, the user would right click and a new dialog box listing these rules would appear. The user could then edit the appropriate rule.

TBTA also contains a "grammar debugger." The user may set a breakpoint in any rule in the grammar. After the user clicks the Generate button, the system executes all of the rules that precede the rule with the breakpoint. The system then stops the execution and lets the user step through the rule with a visual interface that initially shows the input for the rule, how it matches the current state of the text representation, and then, assuming all input conditions are satisfied, it shows how the output of the rule is produced.

By integrating the grammar editor, debugger and execution modules, the user is able to quickly and easily develop the grammar and lexicon so that a clear target text is generated.

3.3 The quick ramp-up grammar acquisition process.

TBTA has several additional features which help users[4] build their grammars very quickly. By far the most common task performed by the restructuring grammar is case frame adjustments. In order to help users build their case-frame-adjustment rules quickly, the system creates the semantic case frames for all the events defined in the TBTA ontology. These case frames are created by examining the corpus analyzed texts. General rules for case frame surface realizations are typically included in the target descriptive grammar, but when a particular event has a non-standard realization, the user only needs to enter the necessary adjustments into the output structure of each rule. Other common tasks that must be performed by the restructuring grammar have been loaded into pre-written rules stored in a library. Users can access these rules and modify them when necessary. A major goal of this project is to produce a set of restructuring rules relevant for specific language families. A new target language user would then simply check which language family applies, and a whole set of rules will automatically be added to the grammar. This is a valuable capability, for example, for the hundreds of Bantu languages in Africa vs. the hundreds of languages in Papua New Guinea; two families which will have widely varying characteristics.

In order to further facilitate the development of the target grammars, a Grammar Development module has been developed. This consists of approximately 300 basic propositions and culminates in a short narrative discourse. Each of the propositions illustrates a particular feature, concept or construction that is found in the TMRs. These propositions illustrate a variety of verbal aspects and moods, relative clauses formed on a variety of semantic roles, patient propositions (object complements) formed with a variety of matrix events, different types of adverbial clauses, different types of questions, etc.. After developing the grammar rules for these basic propositions, the user will have built a solid foundation for his grammar. To emphasize the utility of this module, Figure 4 below shows the number of rules that were required for the Grammar Development module, and how many additional rules had to be entered to translate chapters of text. As can be seen, the number of new rules per chapter drops off dramatically after the module has been completed.

Future development of this project will include the addition of a semi-automatic grammar acquisition module. This module will prompt users to enter responses to very specific questions. The module will then analyze the answers and propose rules that the user will be able to edit and save in the grammar. See Probst et al, 2003, McShane & Nirenburg, 2003 and McShane et al, 2002 for related literature.

Our long-term vision for this project, therefore, is as follows. We will finish the process of authoring semantic representations of the Biblical text (we plan on finishing the narrative portions of the Old Testament by the end of 2007). These representations will subsequently be thoroughly

[4] Again, currently the "users" are TBTA experts, but we expect to expand to the situations described earlier, in large part by upgrading and expanding the features in this section.

checked for consistency and accuracy by trained translation consultants. Each new target language user will first identify the language family being worked in, at which point the corresponding rules for that family will be added. The user will then proceed through the Grammar Development module, which will consist centrally of eliciting target language sentences; the corresponding rules will be automatically created. Any of the automatically created rules can easily be edited, and new rules can be created with the visual rule interface. Text generation of the Bible can then begin, with lexicon development being the main remaining task, guided by the needs of the current text being translated. The amount of additions or edits to the target knowledge sources will approach zero as the number of verses processed increases. The output of the system will then be checked and revised for naturalness by native speakers.

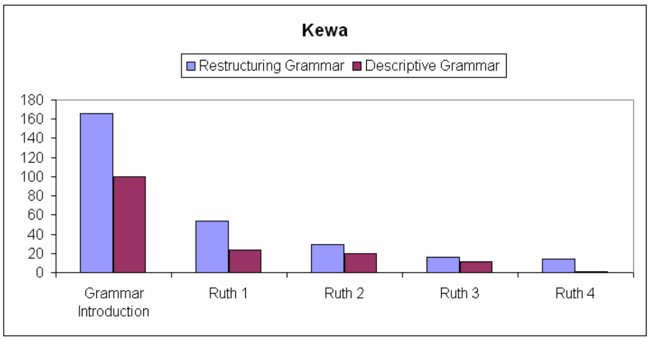

Figure 4: Utility of Grammar Development.

4 Benefits for Bible Translation

The following benefits of TBTA for Bible translation have been identified.

4.1 Clear, accurate and reliable translations.

It might be argued that the controlled English used as input to the document authoring stage would result in translations that miscommunicate as compared to manual translations. In practice, we expect the opposite to occur. Traditional Bible translation (as a gross over-characterization) seeks to produce texts that mirror the spoken language. In the best of all circumstances, translators imagine a fluent native reader reading a perfectly fluent translation to a group of listeners. Unfortunately, this situation rarely occurs. In the context of minority languages, literacy rates are typically low. Long, flowing sentences are often misread. Thus, the apparent paradox: "good" translations can be misunderstood. TBTA tends to produce sentences that are short, with relatively simple syntax (although long sentences can certainly be created; for example, in Kewa, a clause-chaining language, sentences with multiple clauses are preferred). Beginning readers benefit from the simplified syntax and the underlying straightforward semantics, and they will, on the whole, understand the meaning of the text at least as well as a manual translation, a fact that has been borne out by our evaluations. TBTA has been used to generate several books of the Old Testament in English, Korean, Jula and Kewa. In every case, readers have said that the texts are easily understandable, grammatically perfect, and have the same semantic content as the TMRs.

The translations produced by TBTA will also be reliable, in the sense that the underlying semantic representation will have been thoroughly checked. A traditional New Testament translation is checked by a trained consultant. Unfortunately, each consultant comes to the task with a particular set of linguistic and theological strengths, weaknesses and theoretical biases. A centralized representation of meaning can be more thoroughly and uniformly checked, while still maintaining a degree of freedom, as discussed next.

4.2 Standardized treatment of difficult passages while allowing for individuality.

The document authoring tools allow the author to produce alternate analyses and to include optional information such as implicit information. Not every translator will agree on the meaning of a particular passage; thus, we have found it helpful to be able to present these alternative meanings. Much more frequent than theological differences, though, are differences in translation theory. Some translators prefer to be more literal, others are much freer. And in particular, there is a fairly wide spectrum of thought on the inclusion of implicit information. All of these can be addressed with alternative semantic analyses. But our main point here is that these difficult passages and related decisions will be presented to the target language translation team, systematically forcing them to deal with issues that can sometimes be passed over in a traditional translation. The result will be a higher quality translation.

4.3 An aid in language learning and description.

TBTA can help focus and guide the language learning process by presenting the semantic "vocabulary" and constructions that need to be learned. TBTA includes a language learning module. A nice feature of TBTA is that a printed target language grammar and lexicon description can be produced simply by clicking a button.

4.4 Speed! Feasibility!

In field tests conducted so far, TBTA has been used to produce target language texts in one third the time[5] of a parallel manual translation. We expect this savings to increase as more texts are translated within a language, taking advantage of the leverage TBTA provides as the knowledge acquisition curve approaches zero. In practical terms, this makes the translation of the Old Testament feasible. Until now, only small sections of the Old Testament (which is much larger than the New Testament) are typically translated.

Another related time-saver will be the changing requirements and goals of the consultant checking process. Currently, trained consultants must check every verse of a manual translation to ensure accuracy and theological correctness. This is understandably a long process; in fact, it can lengthen the translation project by 50% or even more as compared to the time taken to produce the translation itself. Because the TBTA semantic representations will already have been checked, the consultant checking process will, in large part, fundamentally change. Instead of focusing on accuracy, the consultant can concentrate on making sure readers are understanding. Not only will this be faster, but it will provide an emphasis on target reader understanding that can only improve the translation quality.

5 Benefits for Computational Linguistics

The following benefits of this project are clear:

- A large corpus of semantically analyzed texts (high quality, thoroughly checked, deep semantic representation)
- an abundance of target language knowledge (up to 3000 languages!)
- study in the area of computer-aided language acquisition
- a tool for linguists for describing (and teaching in a field methods class) morphology, syntax and semantics.

6 Conclusion

TBTA is a useful tool for a large task. Presently, we have completed the semantic analysis of the following texts:

- an eye care text from World Vision
- Exodus 1-21
- Esther 1-10
- Luke 1-10
- Mark 1-16
- Matthew 1-28
- Nahum 1-3
- Philippians 1-4
- Ruth 1-4

These 96 chapters contain nearly 3000 verses.[6] We plan on spending the next two years completing the semantic analysis of the narrative sections of the Old Testament. We will continue developing the Grammar Development module, preparing training materials, developing materials for its use in field methods courses for linguistic training and in translation workshops.

7 Acknowledgements

All Korean data courtesy of Dr. Beak Sung Choi. All Kewa data courtesy of Dr. Karl Franklin.

References

Baker, Kathryn L., Alexander M. Franz, Pamela W. Jordan, Teruko Mitamura, Eric H. Nyberg. 1994. "Coping with Ambiguity in a Large-Scale Machine Translation System," In *Proceedings of COLING-94*, Kyoto, Japan.

Stephen Beale, Sergei Nirenburg and Kavi Mahesh. 1995. Semantic analysis in the Mikrokosmos machine translation project. *Proceedings of the 2nd Symposium on Natural Language Processing*. Bangkok, Thailand.

Beale, Stephen and Sergei Nirenburg. 2003. "Just in time grammar." In *Proceedings 2003 International Multiconference in Computer Science and Computer Engineering*, Las Vegas, Nevada.

McShane, M., and Nirenburg, S. 2003. "Parameterizing and Eliciting Text Elements Across Languages for Use in Natural Language Processing Systems," In Machine Translation 18(2): 129-165.

McShane, M., Nirenburg, S., Cowie, J. and Zacharski, R. 2002. "Embedding Knowledge Elicitation and MT Systems within a Single Architecture." In Machine Translation 17(4): 271-305.

Mitamura, T., Nyberg, E. and Carbonell, J. 1991. "An Efficient Interlingua Translation System for Multilingual Document Production," In *Proceedings of MT Summit III*, Washington, DC.

Probst, Katharina, Lori Levin, Erik Petersen, Alon Lavie and Jamie Carbonell. 2003. "MT for Minority Languages Using Elicitation-Based Learning of Syntactic Transfer Rules." In Machine Translation 17(4): 245-270.

Sergei Nirenburg and Victor Raskin. 2004. *Ontological Semantics*. The MIT Press.

[5] Not including the time needed to complete the Grammar Development module.

[6] Only a small portion of this has been checked by consultants.