BRITISH & FOREIGN BIBLE SOCIETY

Hunting the Snark

The problem posed for MT by non-concatenative morphologies

Abstract:

The breadth of languages with which the Bible Societies must work is probably greater than any other organisation. The lack of linguistic databases for most of these languages has encouraged Bible Society to begin developing systems which can analyse automatically some characteristics of natural language. A particular need is the ability identify cognate word forms in a language with the minimum of supervision. The ability to identify close cognates improves the performance of key term analysers and automatic back-translation and once texts are complete contributes to creating concordances lemmata based search routines for these texts which are increasingly being made available on the web. This paper considers the problems created for such processing by complex, non-concatenative morphologies.

1. Introduction

1.1 Bible Translation and MT

Natural languages represent a diverse and complex dataset. Even within those languages which now enjoy support from MT systems the diversity of the individual languages is such that only by employing vast example databases can much progress be made. The most effective systems employ not only large translation memories but also rely heavily on grammatical and morphological data from which translation systems are able to generate acceptable draught translations. That these systems are so successful is cause for real satisfaction and the diversity of the presentations at this conference alone is testimony to the strength of the current generation of MT solutions. Google Translate currently lists sixty five languages for which it offers translation support. By comparison to just a few years ago this is astonishing progress.

As a member of the Bible translation community I often find myself despatched to give presentations about our translation work to groups of supporters. A common question after these presentations is: why don't you just use Google Translate? From the layman's perspective this is a good question and I usually answer it with a brief demonstration:

Genesis 1:1-2 (Septuagint)[1]

ἐν ἀρχῇ ἐποίησεν ὁ θεὸς τὸν οὐρανὸν καὶ τὴν γῆν. ἡ δὲ γῆ ἦν ἀόρατος καὶ ἀκατασκεύαστος καὶ σκότος ἐπάνω τῆς ἀβύσσου καὶ πνεῦμα θεοῦ ἐπεφέρετο ἐπάνω τοῦ ὕδατος.	in the beginning epoíisen God heavens and the earth. And the earth unseen and HN akataskeúastos and Scott epáno avússou and Spirit of God epeféreto epáno waters.

There are some significant problems here but the majority of them can be addressed, at least to some extent, by the provision of better linguistic data for the ancient Greek of the Septuagint. There is, however, a rather bigger issue. For the purposes of the demonstration, the target language for the translation is English. In reality, the vast majority of Bible Translation takes place into languages which have no translation memory databases available to them, indeed, it

1 Greek text from the Septuagint (Rahlfs:1979), translation from Google Translate, Greek -> English – generated 11:30 30th October 2012.

is not unusual for the first task of a Bible translation team to be that of defining an orthography for the language and even if there is a stable orthography and grammars and lexica exist, they are unlikely to be in a form usable by MT systems.

1.2 ParaTExt – A Translator's Workbench

The United Bible Societies (UBS), in partnership with the Summers Institute of Linguistics (SIL), has developed a translation creation and editing suite called ParaTExt (PT). PT is designed to address the issues faced by a typical Bible translator in their daily work.[2] It provides access to the source texts (typically the Hebrew Masoretic (Old Testament) Text and the Ancient Greek text of the New Testament) together with the standard lexica for these texts. It also provides access to modern translations to which the translators may wish to refer. In addition to providing an editing environment and access to base texts PT offers the translators a measure of MT help with common translation and editing tasks such as key term analysis, consistency of spelling, dictionary building and an automatic interlinear back-translation facility to allow the new translation to be objectively assessed as it progresses. Much of this functionality is based upon automatic glossing technologies developed in the UK by the team at British & Foreign Bible Society (BFBS). By comparison to the majority of mainstream MT systems the help given is limited. But the key difference between the PT systems and mainstream MT is that PT systems are designed to work with any natural language[3], rather than a small set of commercial lingua franca. As such, the systems cannot be dependent upon supplied lexica and grammars but must make their own attempt to analyse the language in question as the translation progresses.

The glossing technologies which power most of this analysis are sufficiently robust to handle many of the complexities of language but they have a fundamental requirement which is vital to success. In order to make their assessments of relationships between terms in parallel corpora they have to be able to identify cognate forms of words in the text as it is created by the translators. To help them do this a morphology analyser has been built into the system which is able to identify stem lemmata and their associated morpheme structures. This data is then used to tag individual lexemes for stem and morpheme. The more complex the morphology of the language, the harder this task becomes. For the majority of natural languages which form words by concatenating morphemes with stems into *[prefix]stem[suffix]* structures the analysis is good enough to enable the glossing technologies to work well. Nevertheless, a significant minority remain[4] where the complexity of the morphology is such that the automatic systems within PT struggle to provide a coherent analysis. It is the problem posed by this set of more complex morphologies that this work seeks to address.

2. Non-concatenative morphologies

Within the set of all natural language there exist languages which construct surface forms not only by prefix and suffix agglutination to the stem but by medial modification to the stem itself. It is difficult to determine exactly how many languages behave this way, not least because the number is higher than might first be imagined. Thankfully, many languages which form words

2 Riding & van Steenbergen, 2011
3 The number of living natural languages is hard to pin down precisely but it is generally accepted that there presently in excess of 7,000 - Gordon:2005 & www.ethnologue.com
4 Probably about 75% of all natural languages. Bickel 2005:86

in this manner only use this kind of word formation in a relatively small part of their vocabulary. English is one of them. As a consequence of its Germanic origins English modifies some stems to form plurals by a medial vocalic transformation as in *man* and *men* and forms some tenses of verbs in a similar way as in *sing*, *sang* and *sung*. Such non-concatenative transformations are rare enough in English and like languages not to pose significant problems for PT systems but when the degree of non-concatenative transformation rises, the glossing technologies and their associated morphology analyser begin to struggle. The problem they face is that of not being able to identify cognate forms in the text. In computing terms, the systems are unable to find the beans to count them. Amongst the languages for which this is a particular problem are the Semitic group including, Arabic, Syriac, Amharic and Hebrew but the characteristic is shared by many other languages across the world. One example will serve to demonstrate the problem in comparison to more common concatenative morphologies:

2.1 Example word-formation

The African Bantu language group generally forms words by concatenation of stem and morphemes. Thus a stem such as *penda* (love) (strictly speaking *-pend-*) generates surface forms by adding prefixes and suffixes to the stem as in: *akipenda, anakupenda, atanipenda, mlipenda, mpende, nakupenda, nawapenda, nilipenda, ninakupenda, [-]pendana, [-]pendea, [-]pendwa, sikupendi, tulipenda, tutapenda, ulipenda, ungependa, utapenda, walipenda, wanaupenda, watapenda* etc... This is by no means an exhaustive list of forms, a Swahili verb can generate hundreds of surface forms. Turning now to Hebrew and taking again the example of a verb, in this case קטל (*qātal* - kill) valid forms include: קָטַל, קָטַלְנוּ, תִּקְטֹל, יִקְטְלוּ, קִטְלִי, קְטוּלוֹת, יִקְטְלֵהוּ, etc... Hence forward, in the interests of accessibility, I shall transliterate Hebrew forms into the their equivalents in the Michigan-Claremont encoding for Classical Hebrew. For the examples above this results in the forms: *QF+AL, QF+AL:NW., TIQ:+OL, YIQ:+:LW., QO+:L"Y, Q:+W.LOWT, YIQ:+:L"HW.*. As for the Swahili examples this is a very short list which is a small subset of possible forms.

What the two sets of examples have in common is that each cognate form is built from a stem lemma and various morphemes. The difference between the two sets is the mechanism by which these individual forms are generated. In the case of Swahili, whilst the morphology is complex in the sense that it represents many components, person, number, tense, voice, mood, object concord, negation etc... and so generates many forms, the fundamental template is simply [prefix_morphs]stem[suffix_morphs]. In contrast, a language such as Hebrew forms lexemes not only by prefix-stem and stem-suffix agglutination but also by changing elements within the stem itself. The equivalent template for the Hebrew verb is:

[prefix_morphs]**$1**[infix_morphs]**$2**[infix_morphs]**$3**[suffix_morphs]

where **$1**, **$2** & **$3** represent the Hebrew tri-literal stem. In our example the stem is *Q+L* and everything else is the consequence of changes required by Hebrew word formation. In the case of Classical Hebrew it is, in fact, worse than this as a complex system of punctuation and accentuation is also applied which generates yet more variant forms. For the purposes of this discussion, and in the interests of clarity, I shall omit references to this system[5].

5 The Maoretic cantillation system is a study in itself. The classic work is Wickes 1887, more recent contributions include Jacobson 2002 and Robinson 2002.

2.2 Analysing non-concatenative morphologies

The analysis of concatenative morphologies by MT systems is now generally well understood. A number of methods have proven to be effective including Minimum Description Length (MDL)[6] algorithms, statistical systems for identifying candidate morphemes[7] and the signature or inflection paradigm approach[8] used by the Bible Societies' ParaTExt system[9]. The performance of these systems is roughly the same with all three claiming parsing accuracies at or above 95% for concatenative languages. All of these approaches break down in the face of the complexities posed by non-concatenative systems.

Of the methods for analysing concatenative morphologies, the one best suited for extension into non-concatenative systems is paradigm based analysis. The fundamental problem, however, of how to untangle stems from their associated morpheme templates remains. Whereas in concatenative morphologies it becomes a relatively trivial task to inspect initial and final n-grams in search of statistically significant patterns, with non-concatenative morphologies it is rather less straightforward.

To address this problem the team at BFBS are further developing a system first created in the mid-1990s to help translators identify proper-names across a corpus. The method used to do this was to compare a known rendering of the name in the text to identify other words with common sequences of characters where those sequences are not necessarily concatenative. The method has also proved effective at identifying morpheme templates.

2.2.1 Identifying candidate morpheme templates

The basic premise upon which this and all other morphology analysis systems is based is the fundamental principle that any morpheme structure in a natural language will be present more frequently than any stem lemmata. Thus in English, the suffix morphemes *-ing, -ly, -liness, -ed* etc... will appear in statistically significant numbers by comparison to stems such as *lov-, rid-, wait-* etc... If this premise holds good, we may expect to find non-concatenative patterns in similarly significant numbers. As a first step, we need to generate possible morpheme templates from an analysis of words in the target language. Thus the only pre-requisite for this process is a lexicon of surface forms of words in the language. No lexica or grammars are used to aid the analysis.

The method for generating possible morpheme templates requires each entry in the lexicon to be compared to every other entry and any common sub-sequences noted. As more and more comparisons are made, sub-sequences which correspond to morpheme templates in the language are attested by more and more surface forms. To generate these templates the following method is used:

6 e.g. Goldsmith et al. 2001
7 These two processes are actually equivalent, see Snover, Jarosz and Brent: The use of probabilistic models is equivalent to minimum description length models. Searching for the most probable hypothesis is just as compelling as searching for the smallest hypothesis and a model formulated in one framework can, through some mathematical manipulation, be reformulated into the other framework.
8 Monson et al. 2004
9 Riding 2007.

Let us consider two Hebrew verbs: *Q+L* and *ML+*.[10] The equivalent structure in Hebrew for a completed action and for a continuing action are the forms *QF+AL* and *MFLA+* and *YIQ:+OL* and *YIM:LO+*.[11] To the Hebraist such transformations hold few terrors and it is the work of a moment to recognise that if we remove the stem tri-literals (*Q+L* and *ML+*) we are left with the morpheme templates which are roughly the equivalent of the English *-ed* and *-ing*, in this case *_F_A* and *YI_:_O_*. Persuading a machine to do this without the benefit of direct instruction from a lexicon or grammar is a little more involved. Not only must a way be found to identify the common elements between forms which share the same morpheme template, these elements must be ordered as they occur in the event stream and marked for relative proximity to one another. In short, we need to know what elements are common in both presence and order and also the degree by which each succeeding element is separated from its predecessor. We do this by constructing a two-dimensional matrix which allows us to mark shared elements between two words and applying a simple rule of valid succession which states that: for a shared element to be a valid successor to another shared element, its position in each word must follow the position of the previous shared element in each word. In the context of our matrix we can simply say that successors must appear to the right and below their immediate predecessor, presuming our point of origin to be set top left.

Our first example generates the matrix on the right:

		1	2	3	4	5
	Ø	*Q*	*F*	*+*	*A*	*L*
1	*M*					
2	*F*		*F*			
3	*L*					*L*
4	*A*				*A*	
5	*+*			*+*		

fig. 1 match matrix for QF+AL / MFLA+

We begin at point of origin to ensure a single entry point for the matrix. The first element match is not problematic, F is clearly present in both words. Identifying the second match is more difficult. There are three options, *+*, *A* and *L*. All obey the rule of appearing after their predecessor in both words giving three possible solutions: $S_1\{F, +\}$, $S_2\{F, A\}$, $S_3\{F, L\}$ all mutually exclusive. We now need a way of assessing the relative strengths of each solution. Happily, our matrix offers us the opportunity to assess each solution based on the proximity of successive matched elements. Each matched element can be identified by the coordinates of its position in the matrix. This allows us to rewrite our solutions thus:

$$S_1\{F_{(2,2)}, +_{(3,5)}\}, S_2\{F_{(2,2)}, A_{(4,4)}\}, S_3\{F_{(2,2)}, L_{(5,3)}\}$$

For S_1 we have two matched characters at coordinates (2,2) and (3,5) respectively. We take whichever is the greater of the distances (d) between the x and y coordinates of these two characters. In this case $d_x = 1$ and $d_y = 3$. We take the greater of these two as our distance measure and convert it into a useful value as follows: $1+\left(1-\frac{d}{f}\right)$ where f represents the maximum distance at which it is reasonable to hypothesise a relationship. For the purposes of intra-word analyses $f = 10$ is usually adequate. For S_1 we can now calculate a solution strength as follows: $1+\left(1-\frac{3}{10}\right)=1.7$. Repeating this for S_2 and S_3 we find solution strengths of 1.8 and 1.7 respectively. On the basis of order and proximity analysis we prefer S_2 although at this stage of

10 קָטַל killed, and מָלַט saved. Both of these forms are qal active perfective 3rd p. s. m..

11 יִקְטֹל killing, and יִמְלֹט saving. Both of these forms are qal active imperfective 3rd p. s. m..

the analysis drawing such a conclusion is premature. This example has served to demonstrate the basis for assessing the strength of a simple match solution with only two matching elements. It will, however, often be the case that there are many matching elements between two longer words.

Consider the forms *YIQ:+OL* and *YIM:LO+*. To find all possible match solutions for these words we construct a new matrix thus: (*fig. 2.*)

		1	2	3	4	5	6	7
	ø	Y	I	M	:	L	O	+
1	Y	Y						
2	I		I					
3	Q							
4	:				:			
5	+							+
6	O						O	
7	L					L		

fig. 2 Match matrix for YIQ:+OL / YIM:LO+

By the same process used in the previous example we construct the possible solution sets for this comparison. In this case there is a single path as far as *:* but thereafter three mutually exclusive possibilities exist:
S_1 { $Y_{(1,1)}$, $I_{(2,2)}$, $:_{(4,4)}$, $+_{(7,5)}$ }, S_2 { $Y_{(1,1)}$, $I_{(2,2)}$, $:_{(4,4)}$, $O_{(6,6)}$ }, S_3 { $Y_{(1,1)}$, $I_{(2,2)}$, $:_{(4,4)}$, $L_{(5,7)}$ } using the same method as before we can calculate the strength of the matches in each solution. For S_1 the proximity of *Y->I* gives 1.9, *I->:* also gives 1.9 and *:->+* 1.7. We repeat this for the other solutions and generate the following solution strengths: S_1(1.9, 1.8, 1.7) S_2(1.9, 1.8, 1.8) S_3(1.9, 1.8, 1.7). Taking the product of the proximity values for each pair of matched elements in each set we calculate solution strengths of: S_1=5.814, S_2=6.156, S_3=5.814 .

One more example, again of increasing complexity. This time we shall compare the forms *YIQ:+LW.* and *YIM:L:+W.*.[12]

The matched character intersections on the matrix allow us to plot five valid sequences of matched elements which score as follows (*fig 3*):

		1	2	3	4	5	6	7	8
	ø	Y	I	M	:	L	:	+	W.
1	Y	Y							
2	I		I						
3	Q								
4	:				:		:		
5	+							+	
6	:				:		:		
7	L					L			
8	W.								W.

fig 3. Match matrix for YIQ:+LW / YIM:L:+W

S_1{$Y_{(1,1)}$, $I_{(2,2)}$, $:_{(6,4)}$, $L_{(7,5)}$, $W._{(8,8)}$} scores: 1.9•1.6•1.9•1.7 = 09.8192
S_2{$Y_{(1,1)}$, $I_{(2,2)}$, $:_{(4,4)}$, $L_{(7,5)}$, $W._{(8,8)}$} scores: 1.9•1.8•1.7•1.7 = 09.8838
S_3{$Y_{(1,1)}$, $I_{(2,2)}$, $:_{(4,4)}$, $:_{(6,6)}$, $W._{(8,8)}$} scores: 1.9•1.8•1.8•1.8 = 11.0808
S_4{$Y_{(1,1)}$, $I_{(2,2)}$, $:_{(4,4)}$, $+_{(5,7)}$, $W._{(8,8)}$} scores: 1.9•1.8•1.7•1.7 = 09.8838
S_5{$Y_{(1,1)}$, $I_{(2,2)}$, $:_{(4,6)}$, $+_{(5,7)}$, $W._{(8,8)}$} scores: 1.9•1.6•1.9•1.7 = 09.8192

In this example the strongest solution is S^3. Thereafter two other solutions, S^2 & S^4 share the same score. In this case we have a clear winner but it is entirely possible that in other contexts more than one solution will share the strongest score. We cannot, therefore, expect this algorithm to generate a winning match sequence on every occasion but rather a set of possible solutions of varying strengths. It is only as we assess these solutions in the wider context of the language as a whole that it may be possible to see the strongest candidates emerge.

For each of the examples we have worked there is a set of possible solutions and in each case one

12 The forms יִקְטְלוּ and יִמְלְטוּ represent the qal 3rd p. pl. m. imperfective active of the verbs קָטַל and מָלַט .

solution has scored more highly than the others. The question remains, how do these solutions help us understand the word formation mechanism for this language? Each solution is common to both of the words from which it was derived but more importantly, the complement of each solution may also be able to help us:

Solutions	Complements	
	Q+L	*ML+*
fig 1:		
$S_1\{F_{(2,2)}, +_{(3,5)}\}$	*Q__AL*	*M_LA_*
$S_2\{F_{(2,2)}, A_{(4,4)}\}$	*Q_+_L*	*M_L_+*
$S_3\{F_{(2,2)}, L_{(5,3)}\}$	*Q_+A_*	*M_L__*
fig 2:		
$S_1\ \{ Y_{(1,1)}, I_{(2,2)}, :_{(4,4)}, +_{(7,5)} \}$	*__Q__OL*	*__M_LO_*
$S_2\ \{ Y_{(1,1)}, I_{(2,2)}, :_{(4,4)}, O_{(6,6)} \}$	*__Q_+_L*	*__M_L_+*
$S_3\ \{ Y_{(1,1)}, I_{(2,2)}, :_{(4,4)}, L_{(5,7)} \}$	*__Q_+O_*	*__M__O+*
fig 3:		
$S_1\{Y_{(1,1)}, I_{(2,2)}, :_{(6,4)}, L_{(7,5)}, W._{(8,8)}\}$	*__Q:+___*	*__M__:+_*
$S_2\{Y_{(1,1)}, I_{(2,2)}, :_{(4,4)}, L_{(7,5)}, W._{(8,8)}\}$	*__Q_+:__*	*__M__:+_*
$S_3\{Y_{(1,1)}, I_{(2,2)}, :_{(4,4)}, :_{(6,6)}, W._{(8,8)}\}$	*__Q_+_L_*	*__M_L_+_*
$S_4\{Y_{(1,1)}, I_{(2,2)}, :_{(4,4)}, +_{(5,7)}, W._{(8,8)}\}$	*__Q__:L_*	*__M_L:+_*
$S_5\{Y_{(1,1)}, I_{(2,2)}, :_{(4,6)}, +_{(5,7)}, W._{(8,8)}\}$	*__Q__:L_*	*__M:L___*

fig. 4.

Given the premise that morpheme structures and templates always occur more frequently than stem lemmata in a language we can expect valid template hypotheses to imply the existence of stem lemmata. In the examples we have worked the highest scoring morphology template for each comparison has as its complement the lemmata *__Q_+_L_* and *__M_L_+_*. If we remove the lacunae that represent morpheme structures these become *Q+L* and *ML+*. These are the corresponding Hebrew tri-literal stems for the verbs קָטַל (*QF+AL)* and מָלַט (*MFLA+)*. As we noted above, it will not always be the best scoring solution which identifies the correct stem pattern but the nature of language is such that as more and more solutions are tested those sharing common stem lemmata will become better attested. In other words, the shared patterns across the words in the language both identify and validate the morpheme templates and their stem lemmata.

2.2.2 Validating morpheme templates

Relational verification seeks to score morpheme templates not only on the basis of their length, but on the number of stems with which they are found and the size of the inflection paradigms suggested by those stems. Whereas the concatenative model calculated a solution value (*V*) for a given morpheme by $V = \log(c)^l$ where *c* represented the occurrence of the morpheme in the word list and *l* the length of the morpheme, we must construct a similar means of assessing the value of a particular template solution. Happily, we have described above a way to assess the strength of individual morpheme templates. We hope now to use these values to validate the templates across the word list as a whole:

> Let *MS* represent the sequence of matched elements, let the sequence of matched elements be considered a set of pairs such that the first pair is formed by the first and second elements, the next by the second and third and so on. Let *i* be the i^{th} pair of matched elements. As we have seen, the value of this instance of the solution is calculated as: $V = \prod_{i=1}^{i=|MS|-1} 1 + \left(1 - \frac{d}{f}\right)_i$ and we extend this across the word list as a whole

by taking the product of the value of all the occurrences *n* of this match sequence across the whole word list: $\prod_{1}^{n}\left(\prod_{i=1}^{i=|MS|-1} 1+\left(1-\frac{d}{f}\right)_i\right)_n$ as indicative of the strength of this solution within the context of the word list as a whole.

3. Example analysis – Classical Hebrew

The example data selected for the initial modelling of the process is the Classical Hebrew text-base of the Westminster/Groves Leningrad Codex of the Masoretic Text. This text offered a number of advantages. In the first instance the text of the Hebrew Bible is one of the most studied in the world and as such there are innumerable analyses of its syntax and morphology. Secondly, the text is held in Michigan-Claremont (MC) encoding, a 7-bit ASCII code for Biblical Hebrew developed by Parunak and Whittaker in 1983 for an 'electronic transcription' of the Biblica Hebraïca Stuttgartensia[13]. MC has a number of advantages in comparison to the Unicode encoding for classical Hebrew[14]. The principle reason, however, for preferring MC encoding is purely practical. Deriving morphology templates for Hebrew results in strings of characters and lacunae where many of the characters are represented in Unicode by zero width glyphs which are intended to combine with base characters. When base characters are not present attempts to render such a string of characters can easily result in a heap of overwritten glyphs which is very difficult to interpret. The second advantage to the Groves text-base is that it includes a complete morphological analysis of all forms which can used to verify the results from the parser.

Classical Hebrew has one of the most complex non-concatenative morphologies. As such it is a stern test for any morphology analyser. The text as we have it, however, includes not only the full vowel pointing but also the cantillation marks which identify the relationship between words and clauses in the text. Strictly speaking, the cantillation marks are not really part of the language's morphology. They provide an extremely detailed system of punctuation to aid comprehension. Such systems are very rare. For the purposes of this work the cantillation marks represented a level of noise which made an already difficult task unnecessarily complex and which was unlikely to be present in other languages. Prior to processing they were removed, leaving the just the word forms complete with all consonants and vowel points.

Example MC Encoding:	
UTF-8	MC
With Cantillation & Vowels:	
בְּרֵאשִׁית	B.:/R")$I73YT
Without Cantillation:	
בְּרֵאשִׁית	B.:/R")$IYT
Consonants alone:	
בראשית	BR)$YT

fig. 5.

Having prepared the text for processing the first step was to build a lexicon of all the surface forms (lexemes) in the text. Initial experiments with the whole text quickly demonstrated that the available processing power was insufficient to handle the task. Each iteration of the template discovery algorithm took approximately 25ms and the number of iterations was such that run times were measured in days rather than hours. The size of the text was reduced to just the book of Genesis in which, having discarded cantillation marks, 4,431 lexemes were found.

13 Kittel 1997

14 There are a number of minor issues in the Unicode encoding for Biblical Hebrew. The most significant are different accents sharing the glyph which are not always unambiguous in the text (at least to computers). Such confusions are really only relevant to those working in cantillation studies and include, *silluq/meteg*, *mehuppakh/yetiv* and *azla / pashta*.

Each of these lexemes was tested against every other in the set for common non-concatenative morphology templates using the method described above. The process reported 52,357 possible templates. The vast majority of these templates are random matches of characters and as such are of no interest. Experience applying similar analyses to concatenative morphologies had demonstrated that the longer of morpheme structure was the more likely it was to represent a useful analysis[15]. This combined with the occurrence count provides a helpful way to assess the value of each template. A minimum length for a template was set at 6 items (lacunae being considered as equal to 1 character for this purpose) and a minimum occurrence of the template within the lexicon being set to 30. The template set was therefore ordered on the basis of the product of template length and occurrence, highest first. The outcome of this was a list of putative templates of which the best 1% represented templates which were sufficiently well attested to be credible as initial template models. Further work is needed to demonstrate that this proportion of the result set will give helpful results across language generally. It may be that the proportion will change dependent upon the size of the lexicon and other parameters.

Initial stem hypothesis:

MC	UTF-8	Validity
_$_B_(_	שׁבע	✔
_$_B_R_	שׁבר	✔
_$_L_$_	שׁלשׁ	✔
_$_L_X_	שׁלח	✔
_$_M_(_	שׁמע	✔
_$_M_N_	שׁמן	✔
_$_M_R_	שׁמר	✔
_$_P_X_	שׁפח	✔
_$_R_C_	שׁרץ	✔
_&_M_L_	שׂמל	✔
_(_B_D_	עבד	✔
_(_B_R_	עבר	✔
_(_L_W_	עלו	✘
_)_K_L_	אכל	✔
_)_M_R_	אמר	✔
_B_R_K_	ברך	✔
_C_D_Q_	צדק	✔
_D_B_R_	דבר	✔
_G_D_L_	גדל	✔
_H_$_Q_	חשׁק	✘
_H_L_K_	הלך	✔
_L_D_Y_	לדי	✘
_L_Q_X_	לקח	✔
_L_Y_L_	ליל	✔
_M_$_M_	משׁמ	✘
_M_L_)_	מלא	✔
_M_L_K_	מלך	✔
_M_N_X_	מנח	✔
_M_Q_N_	מקנ	✘
_M_R_)_	מרא	✔
_N_$_)_	נשׂא	✔
_N_$_M_	נשׁמ	✔
_N_B_L_	נבל	✔
_N_P_L_	נפל	✔
_N_R_)_	נרא	✘
_N_S_(_	נסע	✔
_Q_B_R_	קבר	✔
_Q_L_L_	קלל	✔
_Q_R_B_	קרב	✔
_R_K_$_	רכשׁ	✔
_R_P_)_	רפא	✔
_X_P_R_	חפר	✔
_X_Y_T_	חית	✘
_Y_C_)_	יצא	✔
_Y_K_L_	יכל	✔
_Y_L_D_	ילד	✔
_Y_M_L_)	ימלא	✘
_Y_R_)_	ירא	✔
_Z_Q_N_	זקנ	✔

fig. 6.

The templates generated by the first stage of the analysis were then taken back into the lexicon and those lexemes which matched the templates identified. The complement of each template was extracted from these lexemes as putative stems. A threshold of the number of instances a stem should be attested was set. Initially this was set to 3 instances but experiment proved that, in the case of Hebrew, this could be lowered to 2 instances. This threshold generated a set of 50 putative stems (*fig. 6*). These stems were examined within the context of the BDB Hebrew lexicon[16]. 42 were found to be valid. Previous work with concatenative morphologies generated very similar results in terms of accuracy. From these stems further morpheme templates can be derived and validated by their membership of the inflection sets of other stems. This process has proved capable of parsing concatenative lexica with accuracy rates of around 97% from very similar initial stem sets. The next stage of this work is to complete the process of inflection set building with the Hebrew lexicon and then to apply the morpheme templates validated by membership of these sets to parse the lexicon as a whole. Thereafter further experiment will be required with other languages with similar morphologies.

15 Riding 2007
16 Brown et al 1968

4. Summary

Presuming that the results from the completed process mirror those achieved using a similar method of validation for concatenate languages these results hold out the prospect of automatic lemmatisation of texts for languages with complex and non-concatenative morphologies. For Bible translators this offers the prospect of significant improvements in the performance of existing MT based systems for such languages. It is hoped that the use of such analyses at an early stage in a translation project, coupled with an element of supervision from the users, might allow the construction of more sophisticated spelling checks at a much earlier stage in the translation than is presently possible.

Perhaps even more intriguing is the possibility that the template discovery process described here might be applied to clause analysis. Very preliminary results in this context hold out the enticing prospect of mapping clause structures by means of a similar technique.

Acknowledgements

Thanks are due to the British & Foreign Bible Society for supporting this work, for the encouragement of colleagues within the United Bible Societies and to the Groves Center for permission to use the Westminster Leningrad Codex and its associated morphology tables.

J D Riding,
Linguistic Computing
British & Foreign Bible Society
October 2012

Bibliography

Bickel, B. & Nickols, J. *(ed.) The World Atlas of Language Structures* OUP, **2005**

Brown, F., Driver, S. & Briggs, C. *(ed.) A Hebrew and English Lexicon of the Old Testament* Oxford Clarendon Press, **1968**

Goldsmith, J. *Unsupervised Learning of the Morphology of a Natural Language* Computational Linguistics, **2001**, Vol. 27(2), pp. 153-196

Gordon, R.G. *Ethnologue* Gordon, R. G. *(ed.)* SIL International, **2005**

Jacobson, J. *Chanting the Hebrew Bible* Jewish Publication Society, **2002**

Kittel, R. *et al (ed.) Biblia Hebraica Stuttgartensia* Deutsche Bibel Gesellschaft, **1997**

Monson, C. *A Framework for Unsupervised Natural Language Morphology Induction* Proceedings of the Student Workshop at ACL-04 **2004a**

Monson, C., Lavie, A., Carbonell, J. & Levin, L. *Unsupervised Induction of Natural Language Morphology Inflection Classes* Proceedings of the Workshop of the ACL Special Interest Group on Computational Phonology (SIGPHON) **2004b**

Parunak, H., Whitaker, R., Tov, E. & Groves, A. *Biblia Hebraica Stuttgartensia: With Westminster Hebrew Morphology, (electronic ed.)* Internet, **1996.**

Rahlfs, A. *(ed.) Septuaginta* Deutsche Bibel Gesellschaft, **1979**

Riding, J. *A relational method for the automatic analysis of highly-inflectional agglutinative morphologies* Oxford Brookes University, **2007**

Riding, J. & van Steenbergen, G. *Glossing Technology in Paratext 7* The Bible Translator, **2011**, Vol. 62(2), pp. 92-102

Robinson, D. & Levy, E. *Masoretic Hebrew Cantillation and Constituent Structure Analysis 2002* Proceedings of SBL Conference **2002**

Snover, M.G., Jarosz, G.E. & Brent, M.R. *Unsupervised Learning of Morphology Using a Novel Directed Search Algorithm: Taking the First Step* Proceedings of the 6th Workshop of the ACL Special Interest Group in Computational Phonology, Philadelphia, July **2002**, pp. 11-20

Wickes, W. *Hebrew Prose Accents* Oxford Clarendon Press, **1887**