AUTOMATIC ALIGNMENT OF HIEROGLYPHS AND TRANSLITERATION

Mark-Jan Nederhof

ABSTRACT

Automatic alignment has important applications in philology, facilitating study of texts on the basis of electronic resources produced by different scholars. A simple technique is presented to realise such alignment for Ancient Egyptian hieroglyphic texts and transliteration. Preliminary experiments with the technique are reported, and plans for future work are discussed.

1. INTRODUCTION

A convenient form to represent analysis of a manuscript is as *interlinear text*. In this form, the text is divided into fragments, each short enough to fit within the width of a page or of a computer screen. We will refer to such fragments as *phrases*, which may or may not concur with the linguistic meaning of the term. For each phrase, a number of rows present different aspects of the phrase, which may be the original text, some form of transcription, word-by-word gloss, translation, or a combination of these types of data. The data that occupies the *i*-th row of the interlinear text for each phrase is called a *tier*, or sometimes *stream*.

In the case of Ancient Egyptian, interlinear text typically offers three tiers, consisting of hieroglyphs, transliteration and translation. Additional tiers may offer glosses and lexical or syntactic analyses. The hieroglyphic text may be a facsimile, but more often, we find a normalised transcription using an electronic font, especially when the original manuscript is in

hieratic. The text direction is often mirrored with respect to the original manuscript to be left-to-right, to match the directionality of the other tiers.

By transliteration we mean the rendering of the text using the modern Egyptological alphabet, which is composed of some letters from the Latin alphabet in combination with diacritics, and two additional letters representing aleph and ayin.

Interlinear text commonly offers only one translation in one modern language, but considering that many interpretations of some of the more difficult texts are still contentious, it can be very fruitful to compare several different translations, displayed as consecutive tiers. This may also be said about transliteration, especially where the segmentation of a hieroglyphic text into words is uncertain. In general, one particular interpretation of a hieroglyphic text is best represented by the combination of transliteration and translation.

Many applications of interlinear text involve audio recordings. Such a recording in an appropriate visualisation can be one of the tiers, but it may also serve as the basis for annotations. For example, occurrences of words in a transcription as well as prosodic units can be mapped to time intervals within the recording. Alignment of such annotations can be done straightforwardly through the total ordering imposed by the time line of the recording. Several annotations can be compiled by different linguists, allowing automatic creation of interlinear text, typically restricted to a selection of the tiers, depending on the interests of the user.

A survey of tools and techniques involving such applications was presented by Bird and Liberman (2001). They pointed out that annotations can also be mapped to offsets within a particular *textual* resource, in place of anchor points within an audio recording. This requires however that the textual resource is unchanging, and that different scholars agree on the choice of this textual resource.

These constraints regrettably preclude use in many branches of philology. In the example of Ancient Egyptian texts, it would be impractical to demand that all scholars who translate or annotate a text should tag their resources with indices in some canonical representation of the text. Note that a hieroglyphic transcription as interpretation of an hieratic text cannot serve as such a canonical representation, because there may not be any such interpretation that has the approval of the entire community. Existence of lacunas would further exacerbate the problem.

If the creation of interlinear text cannot rely on anchor points in a common resource offering a total ordering, then an obvious alternative is to align different textual resources automatically, by analysing the contents of the tiers. Alignment of, for example, English, French and German translations of Egyptian texts can be done by relatively conventional techniques; see for example Gale and Church (1993). The present paper will focus on automatic alignment of hieroglyphs and transliteration, which is a form of monolingual alignment involving two very different writing systems. The implementation of this task is one significant component within a larger system to create interlinear text out of one or more hieroglyphic transcriptions, transliterations, translations, and lexical and syntactic annotations.

In passing, we would like to point out that similar techniques can also be applied to automatic alignment of different manuscripts of the same text. Examples are the four manuscripts of the *Eloquent Peasant*, the dozens of manuscripts coverings parts of *Sinuhe* and the countless manuscripts offering different versions of the *Book of the Dead*. Alignment of different manuscripts of the same text entails specific problems. For example, a phrase in one manuscript may be absent in another, or entirely different phrases may occur in the respective manuscripts. Even more difficult to handle automatically are cases where the same phrases occur, but in a different order. These issues will not be addressed in any detail here.

The task of automatic alignment of hieroglyphic text and transliteration is related to the automatic transliteration of hieroglyphs, which was investigated in a seminal paper by Rosmorduc (2001). He used finite-state transducers, achieving very high accuracy. Whereas automatic alignment seems an easier task in comparison, it is still far from trivial, especially as we have decided not to involve lexica or grammatical knowledge. The rationale is that incorporating such knowledge could bias certain genres or periods, and make the software less robust.

Another related task is word segmentation, which means dividing a sequence of signs into words. It differs from our alignment task in that the words themselves are not known. Word segmentation is relevant in general for writing systems without explicit word boundaries. It has received much attention for Chinese. Most conventional algorithms for word segmentation rely on the availability of lexica; see e.g. Sproat et al. (1994). Again, this is incompatible with our objectives.

The structure of this paper is as follows. Section 2 discusses the ongoing activities that form the context to the work reported here. The

orthographic model underlying the automatic alignment is discussed in Section 3 and initial experiments are discussed in Section 4. Section 5 outlines plans for further work.

2. CONTEXT

2.1 an XML format for alignment

The XML format AELalign allows encoding of:
- hieroglyphs,
- transliteration,
- translation, and
- lexical annotation.

For one manuscript, there may be several hieroglyphic transcriptions, several transliterations, etc., and these may be distributed over different files. Moreover, several manuscripts for the same text may be included. Constraints on alignment can be explicitly indicated by line numbers in the manuscripts, or by additional anchor points relating one tier to another. For more details, see Nederhof (2002a).

A first trial of its use involved a joint effort over the World Wide Web to translate the *Eloquent Peasant* with a group of students. Participants submitted their interpretations of parts of the text by email, in a very simple plain-text format, containing transliterations, translations and comments. This format was automatically converted to AELalign. In a next phase, the given hieroglyphic text, which was also in the AELalign format, was aligned with the respective interpretations to form an interlinear text in HTML, which could be viewed as a web page. This served as a virtual blackboard, allowing joint discussions about different interpretations.

After this successful trial, small adjustments were made to the format, and the viewing software was reimplemented to provide output in PDF and in a Java applet. The implementation in Java provides the most flexibility, allowing a selection of the tiers to be displayed. The amount of text that fits on each line depends on the width of the window, and as soon as the window size is changed, suitable line breaks are determined anew, leading to a new interlinear text.

An excerpt from the PDF output is given in Figure 1. We see that the hieroglyphic text is conveniently divided into parts that are aligned with phrases consisting of transliteration and translation. Until recently, such precise alignment could only be achieved by manually inserting suitable

42   43
B1 | |

42   43
B1 | mk wj r nḥm ꜥ≠k sḫtj   ḥr | wnm≠f šmꜥ≠j

42   43
B1 | 'Look, I shall take away your donkey, peasant,   because | it ate my barley.

                                                        9.6
R |

                                                        9.6
R   mk ḥm ꜥ≠k | ḥr   wnm jt≠j

                                                        9.6
R   'But look, your donkey | is   eating my barley!

Figure 1: Part of interlinear text showing two versions of the *Eloquent Peasant*

anchor points. From Section 3 onward, it will be explained how precise alignment can be done automatically.

2.2 HIEROGLYPHIC ENCODING

The hieroglyphic encoding we use is called the Revised Encoding Scheme (RES), and represents a significant departure from the *Manuel de Codage* (MdC) from Buurman et al. (1988). The main shortcomings of MdC encoding of hieroglyphs are:

- There is no precisely defined standard independent from any software tool.
- The syntax is chaotic and common interpretations of the official documents seem to entail ambiguities.
- The operators are not nearly expressive enough to represent a fair portion of the relative positioning of signs one finds on good monumental inscriptions.
- The Manual de Codage seems to be the product of feature creep by having it dictate not only the encoding of hieroglyphs themselves but also the layout of a document that contains hieroglyphs, as well as rudimentary grammatical annotations.

A few key properties of RES are:

- The syntax is very simple, and the meaning is rigorously defined. Given a string of characters, it can be decided with certainty whether it is or is not a valid fragment of hieroglyphic encoding, and if so, its visualisation is fully prescribed, with the font and a small number of other parameters as free variables.

- In place of absolute positioning, introduced by some dialects of MdC to make up for shortcomings in its expressivity, a number of operations are available in RES that allow the composition of signs to be described as one sees them, and a suitable appearance can be automatically computed on the basis of a given font. This means that the validity of an encoding can survive a change of font.
- RES basically only involves hieroglyphs and not other types of text. The only exceptions are footnote markers next to hieroglyphs (whose exact positions are determined automatically), and brackets for philological purposes.

An example of the enhanced expressive power is:

insert[te](G39,N5)

The meaning of this use of the 'insert' operation is that **N5** ('sun') is placed in the empty right-upper corner of **G39** ('pintail'). In particular, **N5** is scaled down as much as necessary to leave a default distance between the two signs. (This distance can be adjusted if desired. With distance 0, the two signs are touching.) The reason the validity of this construction may survive a change of font is that the positioning and scaling depend on the sizes and shapes of the individual signs. For example, in a font where the right-upper corner of **G39** leaves less empty space, the occurrence of **N5** would be scaled down more. (It should be pointed out that a similar construction exists in PLOTTEXT, developed by Stief (1985).)

This should be contrasted with the corresponding notation in most dialects of MdC, using an ampersand. The above example would be written **G39&N5**. This construction is called a 'ligature' or 'special group'. Both terms are misleading, because the individual signs are not joined together as in traditional ligatures, and there is nothing special about such groups, considering they are quite common in any hieroglyphic text.

The meaning of the so-called ligatures, in terms of the relative positioning and scaling of signs, is fixed in the font or in the software. Either way, no standardisation is achieved by the notation itself, and different tools could assign different meanings to ligatures. Attempts to exhaustively list all ligatures and prescribe standardised meanings are futile, as any newly found long text will very likely contain ligatures not included in any fixed list. A case in point is the EGPZ sign list, which contains no less than 400 ligatures.[1] While investigating an MdC encoding of Papyrus Westcar, which is

---

[1]   Version 1.0, November 2007, at http://www.egpz.com/resources/egpz.htm.

one of the most popular Middle Egyptian texts, we found two ligatures that were absent from the EGPZ list.

Our comments carry over to superimposition of signs. Instead of requiring a proliferation of combined signs as separate code points as in the case of most MdC dialects, RES offers the 'stack' operation, as for example in:

stack[on](V28,I9)

The introduction of RES by Nederhof (2002b) has not been well received by the Egyptological community. The main objections raised by members of the audience, and by others before and after the meeting, were:

1. The MdC is generally accepted as the standard, and too many existing encoded texts would become obsolete if RES were adopted.
2. The goal of preserving the validity of an encoding across different fonts, which is one of the strengths of RES, is irrelevant because Egyptologists typically throw away an encoding once they have published a text. In other words, the electronic encoding is no more than an intermediate form towards a final product on paper.
3. RES is too verbose. Instead of **C2** as in MdC, one must write **C2[mirror]**.
4. The uniform syntax of RES is irrelevant, as typical users only approach hieroglyphic encoding via a graphical interface.
5. RES is not an XML format.
6. Even the precise placement of signs relative to each other as allowed by RES would not suffice for palaeographic purposes.
7. The rendering of, for example, the 'insert' operation is too expensive and too complicated for some applications.

The first objection is in conflict with the second, and at least one of them must be invalid. The same holds for the third objection versus the fourth and the fifth objections. Apart from this, each of the above allows a number of counter-arguments.

The first objection can be rejected by pointing out that MdC is not a standard. Various tools exist today that each implement one possible interpretation of part of the features from Buurman et al. (1988), and these interpretations vary widely. All of these tools further extend MdC by new features, to make up for shortcomings in its expressivity. However, as different tools add different such features, encoded texts created with one tool become obsolete as soon as that tool becomes obsolete, and exchanging encodings across different tools is problematic.

To be able to benefit from existing texts encoded in MdC, we have implemented a tool to automatically convert various MdC dialects to RES. As very different principles underlie MdC and RES, respectively, manual post-processing is regrettably required in most cases.

As to the second objection, the reason why encodings of hieroglyphs are considered to be ephemeral may be just because the grave inadequacies of available formats such as MdC have so far hindered the development of any large electronic corpora of hieroglyphic texts. Considering the corpora available in other areas of philology, including those involving non-alphabetic writing systems, such as Akkadian and Sumerian, it is unclear why the particular qualities of Ancient Egyptian would preclude the creation of similar corpora in Egyptology, to be freely shared among different scholars.

The syntax of RES is more verbose than that of MdC, in the sense of requiring more characters to describe the same thing, but this helps to make the constructions more self-explanatory, and the main objective was to cast the enhanced expressive power into a uniform syntax. The simplicity of the syntax of RES may not be appreciated by end-users as much as by developers of hieroglyph-processing tools, which counters the fourth objection above.[2] As to the fifth objection, an XML version of RES will be created as soon as an immediate need for it arises, which has not been the case since RES was introduced.

The sixth objection is based on a misunderstanding of what RES wants to achieve. The purpose of an electronic encoding is to offer a visual appearance somewhere between a purely linear sequence of hieroglyphs on the one hand, which would be utterly unacceptable to any scholar, and a facsimile of the original manuscript on the other, which would be impractical in applications involving e.g. interlinear text. RES does not have the pretences to replace facsimiles, but it does move further away from an unacceptably rigid and unrealistic partition of the text surface into perfect squares as MdC would have it.

Furthermore, it cannot be denied that developers and users of MdC software in the past have felt a strong need for more accurate scaling and positioning of signs. In fact, after the introduction of RES, some MdC tools have adopted some of its features and added them to their dialects of MdC.

---

[2]   One striking observation illustrating the relative complexity of MdC notation is the following. The specification of the tokeniser for MdC in Serge Rosmorduc's JSesh is 188 lines long, against 34 lines for RES in our Java implementation.

Regrettably, this exacerbates some of the other problems of MdC, such as lack of standardisation and the chaotic syntax.

In response to the seventh objection above, we have introduced RESlite, in which signs receive absolute values for their positioning and scaling. In applications where the font is fixed, RES and RESlite offer the exact same visual appearance, and both allow a fragment of hieroglyphic text to be divided into smaller fragments, e.g. to allow line breaks where hieroglyphs are part of running text. In practice, RES is the format most suitable for exchange between groups of scholars, whereas RESlite can be used internally in systems to allow quick rendering using very simple software, following one-off automatic conversion from RES to RESlite.

As far as automatic alignment is concerned, the choice of RES as opposed to MdC for the hieroglyphic encoding is not essential, because the implementation as yet ignores relative positioning of signs beyond a purely linear order. Nevertheless, RES is preferable for this task, due to its emphasis on standardisation and avoidance of ad hoc signs and ligatures. Moreover, RES is ideal for interlinear text, allowing automatic line breaks and padding, and explicitly providing for applications to enforce a horizontal left-to-right text direction irrespective of the encoded directionality.

3. MODEL

Experienced Egyptologists would have little difficulty in correctly aligning hieroglyphs with corresponding transliteration. As with any other problem in the realm of artificial intelligence however, it is not so easy to capture expert knowledge in a formal representation allowing the same task to be done reliably by mechanical means.

Whereas alignment seems straightforward in the case of idealised input, many problems arise in practice. For example, some occurrences of signs may have a non-standard reading not listed in any grammar or dictionary. Further, there may be errors, made by the modern scholar in the hieroglyphic encoding or in the transliteration, or errors by the ancient scribe not reflected in the transliteration. An alignment algorithm should therefore be designed to avoid a complete failure of the task when confronted with input that is less than ideal. In particular, upon encountering problematic writings, local errors may be unavoidable, but these should not spread to cause incorrect alignments for larger parts of a text.

The ability of software to give reasonable output even when the input suffers from a limited number of inadequacies is known as *robustness*. In general, the more complicated an algorithm is, the more difficult it is to achieve robustness. We have therefore started our investigations by choosing a very simple 'orthographic' model of how hieroglyphic signs are combined to write words, in terms of their transliteration. This model assumes only two classes of signs, namely phonograms and determinatives. Ideograms will be treated as phonograms with the special property that they can only match against the start of a word. For example, sign **D56** ('leg') will be treated as a phonogram *rd* that can only match against the first two consonants of a word (or more precisely, of a morpheme; see further below).

We will refer to a mapping from signs to collections of possible readings as an 'annotated sign list'. In some cases, the mapping is from a *sequence* of signs to one or more readings; for example, three consecutive occurrences of **N35** ('ripple of water') may together have a reading as phonogram *mw*.

In the experiments, reported in Section 4, we have extracted our annotated sign list from the 'Zeichenliste' of Hannig (1995). It is relatively straightforward to map this list to a data structure that is machine readable. As we wanted to make the experiments reproducible and eliminate subjective decisions as much as possible, Hannig's list seemed preferable to the one from Gardiner (1957), which would have left much more room for interpretation.

It should be noted that Hannig's sign list is less complete than Gardiner's. In particular, many uncommon readings of signs are absent. This does not hinder our experiments however, and in fact, the existence of gaps in the sign list helps us to measure the robustness of the algorithm, in the light of the awareness that no sign list will ever cover all readings of all occurrences of signs in unseen texts.

With a fixed annotated sign list, the actual input to the alignment algorithm consists of a sequence of hieroglyphic signs and a sequence of words in transliteration. The order of the signs is roughly as they occur in the hieroglyphic encoding in RES. An exception is made however for the 'insert' operation, where the order depends on whether the inserted sign is placed before or after the main sign.

The alignment algorithm to be described below reads the hieroglyphic text from beginning to end, maintaining *positions*, which represent the boundaries between pairs of consecutive hieroglyphs, plus the position
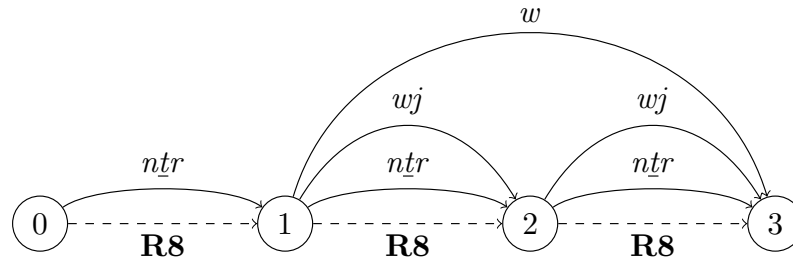
Figure 2: Edges between positions indicate possible readings of signs or sequences of signs. For example, sign **R8** ('cloth wound on a pole') can be read as phonogram *nṯr* . The second occurrence in sequence can alternatively be read as the dual ending *wj*, and the second and third occurrences can together be read as the plural ending *w.* Hence a path from position 0 to position 3 exists with the edges *nṯr* and *w,* respectively, which can be matched against a word *nṯrw* . (To simplify the figure, other readings, such as the feminine dual and plural endings, were omitted.)

before the first hieroglyph and the position after the last hieroglyph. Positions are connected by edges labelled by the possible meanings of the hieroglyphs between those positions, as determined by the annotated sign list. A meaning is either a string of consonants for a reading as phonogram (or ideogram, as explained before), or it is simply the information that a sign can serve as determinative.

Special treatment is needed for numbers and for dual and plural. For a sequence of numerals between two positions, an edge is added between those positions, labelled by the corresponding number in decimal notation, as it might occur in the transliteration. For two consecutive occurrences of the same sign, edges are added labelled by phonograms *wj* and *tj* with the extra constraint that they can only match the final two consonants of a word. Something similar holds for plural, in the case of three occurrences of the same sign, as exemplified in Figure 2.

The words of the transliteration are simply defined as strings separated by white space, consisting of consonants and punctuation signs (i.e. '.', '-', or '='). No attempt was made to do automatic morphological analysis beyond the explicit punctuation signs. For example, the feminine ending *t* is treated like any other consonant, as our transliteration conventions, which follow Hannig (1995), do not mark the boundaries between stems and feminine or plural endings.

The morphemes that are separated by punctuation signs are treated as individual entities however where it concerns our model of orthography. The basic assumption is that a morpheme is written as a sequence of phonograms, together covering all consonants in the transliteration, from left to right, followed by a sequence of zero or more determinatives. By 'left to right' we mean that the first consonant covered by a phonogram should not follow any consonants that have not yet been covered by previous phonograms.

In order to achieve robustness, orthographic analyses are allowed that violate the above basic assumption, at the cost of a 'penalty', the height of which depends on the seriousness of the violation, based on our intuitions about hieroglyphic writing. For example, a phonogram which follows rather than precedes a determinative incurs a penalty of 8. If a semi-vowel (j or w) in the transliteration is not covered by any phonogram, this incurs a penalty of 2, while this penalty is 5 for other consonants. A hieroglyphic sign that is ignored altogether incurs a penalty of 20.

The task is now to automatically determine how consecutive hieroglyphic signs corresponding to words in the transliteration. This is realised by going through the hieroglyphic signs from beginning to end, jumping from position to position following the edges, while at the same time going through the words from the transliteration from beginning to end. The labels of the edges are matched against words from the transliteration, which may incur penalties as outlined above.

One difficulty is however that the correct alignment of hieroglyphs and transliteration is not known in advance, and at each moment, it may be decided to terminate the recognition of the current word of the transliteration and move to the next. Our approach is to pursue all possibilities in parallel, and in the end the solution is returned that minimises the sum of the incurred penalties.

More precisely, we define a *configuration* as a triple consisting of the following three components:

1. A position in the sequence of hieroglyphic signs, as explained before.
2. Precisely one of the following:
- A position in the sequence of words. Positions are defined much as in the case of hieroglyphs. Each represents the boundary between a pair of consecutive words, and there is one position before the first word and one position after the last word.

- An occurrence of a word in the transliteration, together with an indication which of the consonants have been covered by phonograms encountered earlier.

3. The sum of penalties so far.

At the beginning of the alignment algorithm, we have one configuration, with penalty 0, pointing to the beginning of the hieroglyphic text and to the beginning of the transliteration. We call this the *initial* configuration. New configurations are derived from existing ones by different steps. The main steps are:

- The recognition of one word is finalised, moving to the position between the current word and the next.
- From a position between two words, the recognition of the next word is initiated.
- We follow an edge between two hieroglyphs, moving to a next position. In the case of a phonogram, the corresponding consonants in the current word are marked as having been covered.
- We ignore an hieroglyphic sign, moving to the next position.

We say a configuration is *final* if it simultaneously points to the end of the hieroglyphic text and to the end of the transliteration. Of all final configurations, the one is taken that has the smallest penalty. By tracing back how the final configuration originated, one indirectly obtains a preferred matching of sequences of hieroglyphs against words in the transliteration.

The algorithm applies two tricks that allow the task to be done within a few seconds, even for long texts. First, where two competing configurations are identical except for their penalties, the one with the highest penalty is discarded. This can be easily justified, as the configuration with the higher penalty will certainly not be part of the optimal solution when we reach a final configuration. This trick falls within a range of techniques that are known as 'dynamic programming'.

Secondly, for each position within the hieroglyphic text, we only consider the configurations with the N lowest penalties among all configurations associated with that position. Here N is a low number, for example 40. This technique is known as 'beam search'. The rationale is that partial solutions that seems less promising than many competing partial solutions will likely not be part of the optimal solution in the end. Although beam search is very effective in truncating useless computations, there is a risk that the optimal solution itself is truncated. To reduce this risk, N should be chosen sufficiently high.

Figure 3 shows an example of some configurations that match three consecutive hieroglyphic signs against an occurrence of word *ḫprt*, starting from a configuration that points to position 41 just before the corresponding hieroglyphs and to position 82 just before the word, with an overall penalty of 10, which is the sum of the penalties incurred earlier. From this configuration, another is derived that points to the word *ḫprt* between positions 82 and 83, and position 41 as before. This configuration contains information about which consonants have already been covered by phonograms. In the figure this is indicated as a hyphen ('not yet covered') or an asterisk ('covered'). At the beginning we have only hyphens. After **L1** is interpreted as phonogram *ḫpr*, a new configuration is obtained, pointing to position 42 and to the word *ḫprt* between positions 82 and 83 as before, now with three asterisks for the three covered consonants. From here, one may process **D21** as phonogram *r* and **X1** as phonogram *t*, and then finish recognition of the word, leading to the configuration with overall penalty 10 as before, pointing to positions 44 and 83.

Alternatively, the recognition of the word may be terminated just before **D21** is processed, and then the total penalty increases by 5 for the *t* in *ḫprt* that is not accounted for. The resulting configuration has overall penalty 15, and points to positions 42 and 83. More penalties seem unavoidable after that, as **D21** and **X1** may need to be skipped in order to process following words in the transliteration, and each skipped hieroglyph carries a penalty of 20. Note that the higher the overall penalty becomes, the more likely it is that the configurations will eventually be discarded in favour of competing configurations with lower penalties.

A feature was built in to deal with simple cases of honorific transposition, involving a single sign **R8** ('cloth wound on a pole'), **N5** ('sun') or **M23** ('*swt*-plant') to be moved across one or more words of the transliteration. This is realised by allowing such a sign to be skipped and stored in a 'buffer' in a configuration, to be retrieved from a later configuration derived from it. No additional mechanism was needed to deal with transposition for honorific or aesthetic purposes within single words, as the basic orthographic model is already fairly permissive with regard to the order of signs within words (although this by itself causes some errors, as we will see in the next section).

Honorific transposition in general may involve a god's name written with several signs. It is not clear how to deal with this without slowing down
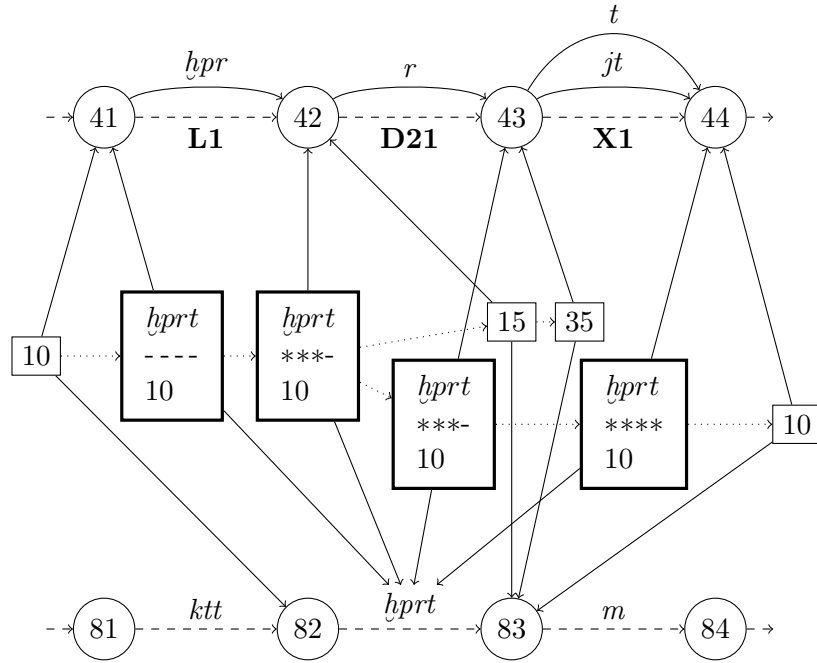
Figure 3: The circles at the top represent positions within the hieroglyphic text, those at the bottom represent positions within the transliteration. The rectangles are configurations, each containing the sum of penalties so far and a pointer to a position in the hieroglyphs. Each of the small rectangles also points to a position between two words in the transliteration. The large rectangles each point to an actual word in the transliteration, while indicating which of the consonants have been covered by phonograms so far. The dotted arrows indicate how one configuration is derived from another. By following such arrows backwards, one can find out how the final configuration with the lowest penalty was obtained from the initial configuration, through a list of steps that identifies the preferred alignment between hieroglyphs and transliteration. (Only those configurations are depicted here that are relevant to the discussion in the running text.)

the alignment algorithm considerably, and therefore we have not attempted to solve the general case in the current implementation.

4. EXPERIMENTS

The first text that was considered is the *Shipwrecked Sailor*. It was found to be very suitable for experimentation with different variants of the alignment algorithm, as it may be the least complicated of all the longer Middle Egyptian texts, having only minor lacunas and few problematic readings.

In a first phase, we segmented the hieroglyphic encoding manually, marking the first sign of the writing of each word. A simple graphical user interface was developed to help this process, allowing signs to be marked by mouse clicks, while putting the corresponding word from the transliteration under the position of each marked sign, and showing the next few words from the transliteration.

The annotated sign list was not changed after we started our experiments, but changes were made to the software at this stage. This means that the error rates cannot be taken as typical for unseen texts of the same level of difficulty, let alone unseen texts of higher levels of difficulty due to, for example, unusual writings of words.

We produced a hieroglyphic encoding of the text, and a transliteration that closely follows the conventions of Hannig (1995), the same dictionary from which the annotated sign list was extracted. By these conventions, the text is 1014 words long. A compound word consisting of two parts connected by a hyphen was counted as one word. Also suffix pronouns were not counted separately.

In a first phase, we segmented the hieroglyphic encoding manually, marking the first sign of the writing of each word. A simple graphical user interface was developed to help this process, allowing signs to be marked by mouse clicks, while putting the corresponding word from the transliteration under the position of each marked sign, and showing the next few words from the transliteration.

In a second phase, the automatic alignment was run to find the first sign corresponding to each word. This was compared to the manual alignment, and the graphical user interface then identified the differences by highlighting. Some auxiliary tools were added to provide explanations why certain mismatches between manual and automatic alignment arose. This includes a tracer, showing the steps of the alignment process for a selected part of the text.

Among the 1014 words, only 12 errors were made by the automatic alignment. These can be divided into 8 errors that are due to gaps in the annotated sign list, and only 4 that are due to inadequacies of the orthographic model. Examples of gaps in the sign list are the absence of the reading of **A50** ('man of rank seated on chair') as ideogram for *špsj,* and the

absence of the reading of **A12** ('soldier with bow and quiver') as ideogram for *mšꜥ*.

The crudeness of the orthographic model accounts for the failure to match **F20** ('tongue of ox') with reading as phonogram *ns* against a subsequence of consonants in the word *nj-sw*. One error occurs in *ḫprt·n rdjt*, where the second occurrence of **D21** ('mouth') is matched to the *r* from *ḫprt·n* rather than the *r* from *rdjt*; the problem here is that the model does not pose enough restrictions on the order of phonograms. In two occurrences of *snṯr* ('incense') the honorific transposition of **R8** ('cloth wound on a pole') misleads the model into taking the sign as determinative of the preceding word.

Whereas each of the above errors could clearly be eliminated by an ad hoc patch of the model, it seems likely that every unseen text will raise new problems, and a 100% accuracy is beyond reach. Furthermore, a frequent observation in computational linguistics is that tweaking models to correctly handle specific cases may inadvertently lead to other cases being handled incorrectly. Moreover, increasing coverage, for example, by adding possible readings to the sign list, may well lead to a decrease in accuracy.

On the positive side, for each of the cases discussed above, no trailing errors in subsequent words ensued. This means the algorithm is very robust, in the sense that local errors do not tend to spread to larger parts of the text. Moreover, for purposes of producing interlinear representations, it may not be a cause for great concern to have the start of a word misidentified by a distance of only one or two signs.

In a second experiment we investigated Papyrus Westcar, repeating the above procedures. This was done after all parameters of the model had been fixed. This means that the results can be seen as typical for unseen texts of the same level of difficulty. However, due to the many lacunas, it was often problematic to identify the sign occurrence where we would want the automatic alignment to find the beginning of a word. Mismatches between manual and automatic alignment that arose as a direct result of lacunas have therefore been ignored, leaving 81 errors, among the 2683 words of the transliteration.

Of these errors, 24 are due to gaps in the annotated sign list, and the remaining 57 must be blamed on inadequacies of the orthographic model. Among the latter, the most frequent problem is honorific transposition within a single word, accounting for 33 errors. Of these, 14 occur in the writing of *nsw-bjtj* and 6 in the writing of *snṯr*.

The other inadequacies of the orthographic model were found at pairs of consecutive words sharing one or more consonants, accounting for 24 errors. Most notably, in 14 occurrences of _ḏd·jn_ _ḏdj_ the first occurrence of **R11** ('column imitating a bundle of stacks'), with reading as phonogram _ḏd_, is incorrectly taken as part of the writing of the first word _ḏd·jn_, rather than the second word _ḏdj,_ which has two occurrences of **R11**. (See Gardiner (1957, 502) for the reading of two consecutive occurrences of **R11**.)

5. FUTURE WORK

The above reported preliminary results from work in progress. The orthographic model we have described allows a large spectrum of refinements, and the current project plans to pursue several of them. This includes evaluation on the basis of a wider range of texts.

A first priority will be the creation of a sign list that contains more detailed and accurate annotations on possible readings of signs. Although the recent Unicode proposal (Everson and Richmond, 2007) greatly contributes to the standardisation of signs used in electronic encoding of hieroglyphs, it is regrettable that no accompanying document is currently being planned that summarises and updates the information about the signs collected by Gardiner (1957; as well as several other documents). It cannot be emphasised enough that electronic resources offering such information are of the highest importance to automatic processing of hieroglyphic texts.

The creation of annotated sign lists in an electronic format also forces us to look closer at the different classes of signs and their functions in the writing of words. Whereas some Egyptian grammars distinguish between only three different classes of signs, viz. phonograms, ideograms (also called logograms) and determinatives, some publications use a finer distinction. Schenkel (1971) in addition offers a formal description of how words are composed of signs with various functions. This description cannot be readily employed for our purposes however, as no sign list exists that is annotated with corresponding functions. Furthermore, Schenkel's work does not directly link hieroglyphic writing to transliteration. For example, it does not specify how to deal with phonetic complements.

The annotated sign list that we used in the experiments was derived from the 'Zeichenliste' of Hannig (1995). The original list distinguishes between Phon, Log, Abk, Det, Phono-Det, and Log/Det. Whereas the informal meanings of these concepts may be clear, it is less obvious what

functions these classes of signs should have in a formal model of orthography.

With refinements of the orthographic model, the mechanism of penalties described in Section 3 may become harder to maintain. The finer the constraints are that one imposes on the orthography, the more frequently will constraints be violated by *valid* orthographic analyses of actually occurring hieroglyphic text, and thereby penalties may be incurred in all competing analyses. The selection of the desired analysis will therefore often depend on a suitable choice of the relative heights of different kinds of penalties. Regrettably, human intuitions tend to be quite unreliable when it comes to estimating quantitative aspects of language or, in this case, writing systems.

We therefore need to investigate stochastic approaches, to replace penalties by probabilities that are automatically estimated on the basis of annotated or unannotated hieroglyphic texts. Due to the nature of the writing system, which lacks unique standardised spellings, and due to the sparsity of the data, it would be infeasible to estimate the probability of each possible variant spelling of each word separately. A more promising approach is to compute parameters that abstract away from the actual consonants of a word in transliteration, looking at the order in which, for example, phonograms are used to represent the consonants in respective positions.

As an example, consider the writing of *nst* ('throne') as:

N35 : F20 - X1 : W11

The exact probability of this writing, given the word in transliteration, is:

$$P(\textbf{N35},\textbf{F20},\textbf{X1},\textbf{W11} \mid \textit{nst}) = P(\textbf{N35} \mid \textit{nst}) \cdot$$
$$P(\textbf{F20} \mid \textit{nst},\textbf{N35}) \cdot$$
$$P(\textbf{X1} \mid \textit{nst},\textbf{N35},\textbf{F20}) \cdot$$
$$P(\textbf{W11} \mid \textit{nst},\textbf{N35},\textbf{F20},\textbf{X1}) \cdot$$
$$P(\textbf{end} \mid \textit{nst},\textbf{N35},\textbf{F20},\textbf{X1},\textbf{W11}).$$

For example, the third factor in the right-hand side of this equation should be read as the probability that **X1** is the third sign in the writing of *nst*, following the signs **N35** and **F20** in this order. The final factor represents the probability that the word ends after the given list of four signs.

Whereas accurate estimation of each of the factors in the above is infeasible, we can approximate them by for example:

$$P(\textbf{N35},\textbf{F20},\textbf{X1},\textbf{W11} \mid \text{nst}) \approx P(\textbf{*--} \mid \text{---}) \cdot$$

$$P(**\text{-} \mid *\text{--})\cdot$$
$$P(\text{--}* \mid **\text{-})\cdot$$
$$P(\text{det} \mid ***)\cdot$$
$$P(\text{end} \mid ***),$$

given the information that **N35** can be a phonogram *n*, which concurs with the first consonant of *nst*, **W11** can be a determinative, etc. Each minus sign or asterisk in the above represents a position in the word in transliteration. The asterisks in the left-hand sides of factors of the form $P(\cdot \mid \cdot)$ are the consonants covered by the next phonogram. The asterisks in the right-hand sides indicate the consonants that have been covered by previous phonograms. For example, $P(\text{--}* \mid **\text{-})$ represents the probability that the next sign is a phonogram matching the third consonant of a three-consonant word, given that the first and second consonants have already been covered by previous phonograms.

There are many variants of such a model. For example, probabilities can be conditioned on the previous one or two signs (cf. bigrams and trigrams), or appropriate abstractions from those signs, making use of 'smoothing' of probabilities in the case of sparse training data. Very similar techniques exist for other applications in computational linguistics, such as part-of-speech tagging (Manning and Schütze, 1999).

So far we have assumed that hieroglyphic text is processed as a linear list of signs, without indication of the exact relative positioning. In particular, line breaks and the separations between quadrats are ignored. There are cases however where relative positioning is essential to the correct reading of hieroglyphs. One classical example is *m-ẖnw* written with **N35a** ('three ripples of water') below **W24** ('bowl'); see Gardiner (1957, 134). In the investigated texts, no examples were found of incorrect alignment of hieroglyphs and transliteration that could be amended if relative positioning beyond a purely linear order were to be taken into account. It cannot be excluded however that relative positioning could help to increase the accuracy of alignment.

Signs may generally be grouped together following aesthetic principles, irrespective of how a sequence of signs is to be segmented into words. For example, if the last sign of one word and the first sign of the following word are both roughly one quadrat in width and half a quadrat in height, they may be grouped together into a single quadrat, with one sign above the other. An interesting conjecture by Horst Beinlich (personal communication) is however that there was a certain tendency to let the boundaries

between consecutive words concur with boundaries between consecutive quadrats. This merits further investigation. To the extent the conjecture may be confirmed by the data, it holds the potential to enhance the accuracy of automatic alignment.

We have found that the penalties discussed in Section 3 sometimes signal errors in the hieroglyphic encoding, often due to a confusion between signs with similar appearances. Another suggestion for further research is therefore to develop tools that highlight potential errors in hieroglyphic transcriptions.

Lastly, it should be pointed out that the most useful applications of automatic alignment are at this moment hindered by the fact that many hieroglyphic transcriptions and translations are available only in printed form. It is highly desirable, for this reason and for many others, that scholars in the future will make more of their textual resources available in suitable electronic formats, either free of copyright or at least explicitly allowing use within viewing software.

6. CONCLUSIONS

Whereas the work reported here is in early stages, some conclusions can already be drawn. First, automatic alignment of hieroglyphs and transliteration is feasible with very simple techniques, without using lexica or grammatical knowledge. The accuracy may vary across texts, but experiments show that at least some texts allow a very high accuracy. In addition, there is ample room for refinements of the discussed techniques, with the potential to further reduce the error rate.

Second, our work underlines the importance of standardisation of hieroglyphic encoding. In addition, the creation of electronic resources, such as annotated sign lists documenting the possible functions of signs in the writing of words, is essential for automatic processing of texts.

7. ACKNOWLEDGEMENTS

REFERENCES

J. Buurman, N. Grimal, M. Hainsworth, J. Hallof, and D. van der Plas. *Inventaire des signes hiéroglyphiques en vue de leur saisie informatique.* Institut de France, Paris, 1988.

S. Bird and M. Liberman. A formal framework for linguistic annotation. *Speech Communication*, 33: 23–60, 2001.

M. Everson and B. Richmond. Proposal to encode Egyptian hieroglyphs in the SMP of the UCS. Working Group Document ISO/IEC JTC1/SC2/WG2 N3237, International Organization for Standardization, 2007.

A.H. Gardiner. *Egyptian Grammar.* Griffith Institute, Ashmolean Museum, Oxford, 1957.

W.A. Gale and K.W. Church. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1): 75–102, 1993.

R. Hannig. *Grosses Handwörterbuch Ägyptisch-Deutsch: die Sprache der Pharaonen (2800–950 v.Chr.).* Verlag Philipp von Zabern, Mainz, 1995.

C.D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing.* MIT Press, 1999.

M.-J. Nederhof. Alignment of resources on Egyptian texts based on XML. In *Proceedings of the 14th Table Ronde Informatique et Egyptologie*, 2002a. On CD-ROM.

M.-J. Nederhof. A revised encoding scheme for hieroglyphic. In *Proceedings of the 14th Table Ronde Informatique et Egyptologie*, 2002b. On CD-ROM.

S. Rosmorduc. Transducteurs pour la translittération des hiéroglyphes. Unpublished paper presented at TALN 2001, 2001.

W. Schenkel. Zur Struktur der Hieroglyphenschrift. *Mitteilungen des deutschen archäologischen Instituts, Abteilung Kairo*, 27: 85–98, 1971.

R. Sproat, C. Shih, W. Gale, and N. Chang. A stochastic finite-state word-segmentation algorithm for Chinese. In *32nd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 66–73, Las Cruces, New Mexico, USA, 1994.

N. Stief. Hieroglyphen, Koptisch, Umschrift, u.a. – ein Textausgabesystem. *Göttinger Miszellen*, 86: 37–44, 1985.