Automatic Creation of Interlinear Text for Philological Purposes

Mark-Jan Nederhof

School of Computer Science, University of St Andrews, North Haugh, St Andrews, Fife, KY16 9SX, Scotland
http://www.cs.st-andrews.ac.uk/~mjn

ABSTRACT. *Interlinear text presents a collection of interpretations of a manuscript. Whereas such a form is often compiled by a single author or a single team of scholars, we here consider automatic creation of interlinear text out of independently created linguistic resources. In terms of mathematical structures, we investigate the constraints one may want to impose on the rendering and pair-wise alignment of resources, and present a computer algorithm that solves those constraints, resulting in suitable interlinear text.*

RÉSUMÉ. *Les gloses interlinéaires permettent de présenter simultanément une collection d'interprétations d'un manuscrit. Bien qu'une telle forme soit fréquemment le produit du travail d'un seul chercheur ou d'une seule équipe de chercheurs, nous nous intéressons à la création automatique de gloses interlinéaires à partir de ressources linguistiques créées de manière indépendante. En termes de structures mathématiques, nous étudions les contraintes que l'on peut imposer pour le rendu ainsi que l'appariement des ressources, et nous exposons un algorithme qui résout ces contraintes et produit un texte interlinéaire convenable.*

KEYWORDS: *philology, interlinear text, natural language processing, electronic text formats*

MOTS-CLÉS : *philologie, texte interlinéaire, traitement automatique des langues, formats électroniques de documents*

1. Introduction

Scholars studying an ancient text may benefit from having at their disposal an extensive collection of translations, analyses and other annotations previously compiled by others. The less consensus there is on the correct interpretation of a text, the more essential it is to be able to compare different viewpoints. This holds for linguists studying a text as much as for, say, historians, as various interpretations can lead to very different interpretations of historical sources.

Lack of consensus on the interpretation of a text typically arises in the case of ancient languages and scripts which were deciphered in recent times, such as Ancient Egyptian and Mayan; but also in the case of, say, early Christian manuscripts there can be much room for debate. Hence our discussion is relevant to diverse disciplines such as philology, biblical criticism and ancient history.

Textual resources can consist of several *tiers*, each presenting one type of annotation. For example, one tier might be a transcription of a manuscript and another might be a translation or grammatical analysis.

Comparing different textual resources printed in books has traditionally been done by arranging them on a desk and going through a text phrase by phrase. Today, many textual resources are available in electronic format and can be displayed on a computer screen. Textual resources in, say, PDF format possess a fixed page layout, and therefore comparing different resources on a screen is not necessarily much easier than comparing the same resources printed on paper.

More substantial benefit is derived, however, from having several resources combined into *interlinear text*. In such a representation, the text is divided into fragments, each short enough so that the corresponding parts of the tiers each fit within the width of a printed page or of a computer screen, one printed below the other.

A number of lines of tiers corresponding to one fragment of a manuscript will be referred to as a *section*. Interlinear text is composed of a series of sections, each typically consisting of the same number of lines, provided each of the tiers covers the complete manuscript. We may also assume the tiers occur in the same order in each section.

There can be many different ways of dividing a text and its annotations into sections, and within each section there can be much freedom in how to align the different tiers. In glosses, an isolated word in translation can be vertically aligned with (a transcription of) the manuscript, whereas a word-by-word alignment with a larger phrase in translation is often not possible, because of differences in word order between the source language and the target language. Within such constraints, however, much may depend on the personal preferences of the scholar studying a text.

Many projects in computational linguistics and philology that involve interlinear text assume that the textual resources are the product of collaboration between scholars who agree on ways to link the tiers. At the very least, it is assumed that there is a fixed manuscript to which different types of annotation can be linked, through *anchor points*. For the manuscript, one may also take an audio recording, with anchor points being points in time, as discussed by Bird and Liberman (2001).

In many branches of philology however such assumptions are unwarranted. Firstly, well-known ancient texts are typically studied by many scholars around the world, who may not agree on conventions of annotation. Secondly, there may be no canonical electronic encoding of an ancient text to which to connect anchor points. This especially holds for damaged texts with lacunas and contentious readings, so that it is not even clear how many symbols a manuscript contains. In addition, financial means are often not available to edit existing resources to conform to common standards or common anchor points.

For these reasons, we depart from conventional methods of creating interlinear text, and investigate how such a representation can be automatically obtained under the following conditions:

– The input resources were independently created by different scholars.

– Minimal effort should be required to put such material into suitable electronic formats, apart from scanning of material that was previously only available in printed form.

– The tool that produces the interlinear text can be parameterised according to the personal preferences of the scholar who uses the tool, for example for font family, font size and page width.

– Alignment of resources can be incrementally improved by manually placing anchor points and linking different tiers, if this is desired.

– External components can be plugged in to automatically align tiers. For example, different translations of a manuscript can be aligned by techniques developed for modern languages, such as those from Gale and Church (1993) and Och and Ney (2003). Specialised alignment techniques can be used for particular ancient languages, such as alignment of hieroglyphic text and transliteration, as proposed by Nederhof (2008).

The structure of this paper is as follows. In Section 2 we discuss a few examples to illustrate some typical problems that arise with automatic creation of interlinear text, and motivate the material that follows. Section 3 then proposes a formal framework in which to express the various constraints on interlinear text, given a collection of textual resources and two types of links between them. A precise formulation of what constitutes an acceptable interlinear text is given in Section 4, together with an algorithm to find such a solution. Section 5 outlines an existing tool that implements most of the discussed ideas. We have not carried out any systematic investigation of interlinear text for writing systems with an inherently right-to-left text direction, but can offer some tentative remarks on this issue in Section 6. Conclusions can be found in Section 7.

2. Examples

The issues above are illustrated by means of a passage from an Ancient Egyptian text called ‘The Eloquent Peasant’. The main parts of this text are nine petitions by a provincial peasant, who impresses by using highly literary forms. Modern scholars differ however on the correct interpretation and translation of these petitions. Although many Ancient Egyptian texts may be less problematic, similar disagreements can be witnessed frequently, throughout various genres and periods.

Figure 1 presents the passage in five sections of interlinear text. Included in each of the sections are four tiers, namely a published transcription (hieroglyphs), then our transliteration in the Egyptological alphabet, and lastly two published English translations. The first two tiers include line numbers from the manuscript.

The text was divided into sections to match a partition of the text into linguistically meaningful units, such as noun phrases and verb phrases. It is difficult to give a precise characterisation of the kind of phrase that we would like to see printed whole within one section, rather than broken up between two or more sections. As an approximation, one could say that such a phrase corresponds to a sequence of words in the original text that one would translate as one phrase in another language, before turning to the next phrase, and that a change of word order between source and target languages typically takes place within one phrase.

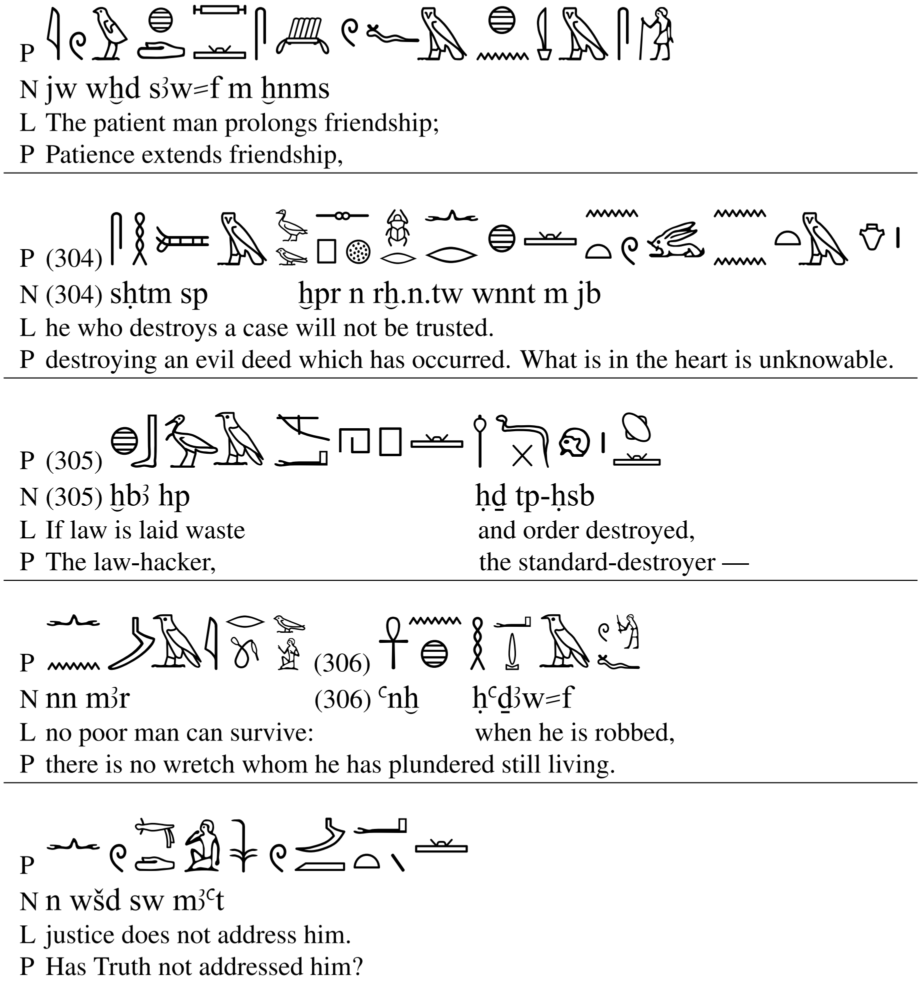
Note that line numbers, such as ‘(306)’ in the fourth section, need not occur near the left edge. This is a consequence of our desire to keep phrases wholly within one section where possible. Especially where a line in the manuscript is broken in the middle of a word, it would be unwise to demand that the line number concurs with the left edge of a section.

Of the five sections in Figure 1, the first and last are relatively straightforward. The tiers are merely aligned at the left. In the third section, the transliteration and the two translations are each broken up into two pieces, to allow alignment with the hieroglyphic, to emphasise the conjunction in the sentence structure.

Similarly, in the second section, part of the transliteration is moved to the right, to align with the two translations of the second half of the phrase. These two translations assume a totally different partition of the text into phrases. Lichtheim motivates her translation *"will not be trusted"* on the basis of *"becomes one-does-not-know-what-is-in-the-heart"* (see her footnote 27), taking ẖpr (*"become"* or *"occur"*) as the main verb. Parkinson in contrast takes ẖpr to modify the preceding noun sp (*"deed"*).

Also in the fourth section, which we will not discuss in detail, there is an interesting difference in interpretation of the text by various scholars. Once more, comparison is made easy by the interlinear text, relying on the respective tiers to be suitably aligned.

One may naively assume that the creation of interlinear text is a matter of printing chunks, representing phrases of some sort, starting at the left end of one section, until the page width is exhausted, and then continuing with the next section. The example



P

N jw wḫd sꜣw=f m ḫnms

L The patient man prolongs friendship;

P Patience extends friendship,

P (304)

N (304) sḥtm sp ḫpr n rḫ.n.tw wnnt m jb

L he who destroys a case will not be trusted.

P destroying an evil deed which has occurred. What is in the heart is unknowable.

P (305)

N (305) ḫbꜣ hp ḥḏ tp-ḥsb

L If law is laid waste and order destroyed,

P The law-hacker, the standard-destroyer —

P (306)

N nn mꜣr (306) ꜥnḫ ḥꜥḏꜣw=f

L no poor man can survive: when he is robbed,

P there is no wretch whom he has plundered still living.

P

N n wšd sw mꜣꜥt

L justice does not address him.

P Has Truth not addressed him?

Figure 1. *A sample of interlinear text, treating a passage from 'The Eloquent Peasant', manuscript version B1. The first line in each section is a transcription, following Parkinson (1991). The second line is our transliteration. The two English translations are by Lichtheim (1975) and Parkinson (1997).*

above shows that the situation is much more complicated. Firstly, the input textual resources may not contain explicit phrase boundaries, and secondly, different interpretations of a text may place phrase boundaries at distinct textual positions, so that there will be no or very few positions where a break between two sections would be consistent with all indicated phrase boundaries.

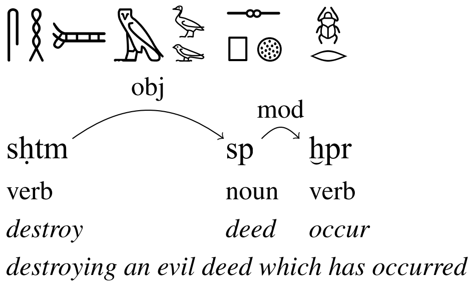


Figure 2. *Illustration of interlinear text incorporating lexical and syntactic analysis*

A second example, in Figure 2, illustrates interlinear text containing grammatical analyses, in this case part-of-speech tagging and a simple form of dependency structure.

There are a number of reasons why we have focused on Ancient Egyptian for the study of automatic creation of interlinear text. First, the directionality of a manuscript can be rendered left-to-right to allow alignment with modern translations and other annotations. Hieroglyphic text was written left-to-right or right-to-left, in rows or in columns, with the predominant direction being right-to-left and frequently in columns; it is immediately clear whether text is written left-to-right or right-to-left, as hieroglyphs that represent people and animals face the beginning of the text. A left-to-right direction in rows that is suitable for our needs can generally be obtained, where necessary by mirroring hieroglyphic text in its entirety, and by rearranging roughly square groups of hieroglyphs called *quadrats* in a horizontal sequence instead of an original vertical sequence.

For writing systems that do not allow a left-to-right text direction in rows, interlinear text must be created by different principles. We will briefly touch upon this matter in Section 6.

A second reason why we have investigated Ancient Egyptian is that its writing system poses problems that would not arise in branches of philology that only involve Latin scripts. Such problems include the existence of transcription next to transliteration as separate tiers, and specific issues involving formatting of hieroglyphic text, such as determining appropriate line breaks and padding.

Lastly, there are several texts for which multiple manuscripts have survived. Alignment of different versions of the same text poses a number of interesting problems.

3. Formalisation

In this section we investigate formal requirements that one may impose on interlinear text. The emphasis will be on operations that can be applied to the input data, in the sense of an application programming interface (API), as opposed to superficial aspects of file formats.

3.1. *Tiers*

We let T denote a collection of tiers belonging to the same manuscript. Each tier $t \in T$ is a linear structure that represents an electronic encoding of the manuscript, or an annotation of some kind, such as a translation, a gloss, a grammatical analysis, etc. Additionally, tiers may contain footnotes and row/column numbers referring to the physical representation of the manuscript. Each tier may be an independently created electronic resource, stored in a separate file, or alternatively several tiers may be bundled together in one file.

For each $t \in T$, we define $\mathit{length}(t)$ to be the length of the tier, in terms of the number of indivisible orthographic elements. Such elements may be occurrences of letters in Latin scripts, or hieroglyphs in Ancient Egyptian. We will henceforth refer to such elements as *symbols*. We will also talk about *positions* in a tier t, which are numbers between 0 to $\mathit{length}(t)$, boundaries included. It may be convenient to think of positions as locations between symbols, with position 0 denoting the location preceding the first symbol and position $\mathit{length}(t)$ denoting the location after the last symbol. Where there can be confusion over which tier t a position a belongs to, we denote the position by a pair (t, a), $0 \leq a \leq \mathit{length}(t)$.

For each position (t, a), the boolean expression $\mathit{Breakable}(t, a)$ is defined to be true if and only if there may be a break at position a. This may be either a section break or a break between two phrases in the same section of interlinear text. In the case of Latin scripts, the position preceding the first letter of each word is breakable. If we allow hyphenation, then there may also be breakable positions within words. In the case of hieroglyphs, breakable positions may be those just preceding the start of a quadrat, whereas our hieroglyphic encoding also offers provisions for breakable positions within quadrats. By convention, we assume 0 and $\mathit{length}(t)$ are breakable positions in any tier t.

For each breakable position a in tier t, we define $\mathit{penalty}(t, a)$ to be a non-negative number indicating the relative quality of having a section break at that position, that is, to have text just preceding position a in one section and the text just following position a in the next section. The higher the number $\mathit{penalty}(t, a)$, the less desirable a section break is. A penalty of 0 could typically be assigned to the beginning of each sentence (in the manuscript or in a translation). A higher number could be assigned to the beginning of a phrase that is not the first phrase in a sentence. The highest penalty could be assigned to positions within words, which would require hyphenation. In the case of hieroglyphic text, one could assign the value 0 to the beginning of each quadrat and a higher number to other positions. It may be convenient to define $penalty(t, a) = \infty$ if and only if $Breakable(t, a)$ is false.

We define a *range* to be a triple (t, b, e), where t is a tier and b and e are positions, with $0 \leq b \leq e \leq length(t)$. A range represents a substring of a tier.

For each range (t, b, e), with breakable positions b and e, the expression $width(t, b, e)$ denotes the width of the range, in terms of a fixed unit, such as distance in points or pixels on the screen. This width is used mainly at the right end of a section, to ensure that text does not run into the right margin. Note that this function assumes a fixed font family and font size.

A related quantity is $advance(t, b, e)$. It differs from $width(t, b, e)$ in that it includes the (minimal) width of whitespace that must be printed between range (t, b, e) and any following range (t, e, a). Therefore, $advance(t, b, e)$ is commonly slightly larger than $width(t, b, e)$. The whitespace between two consecutive ranges (t, b, e) and (t, e, a) that exceeds the minimal amount dictated by $advance(t, b, e)$ will be referred to as *padding*. Because of padding, positions may be thought of as being tied to the beginning of the next symbol, rather than to the end of the previous symbol.

For alignment, to be discussed in the next section, we also need a quantity $dist(t, b, e)$, which is the distance between positions b and e in tier t.

Figure 3 illustrates the concepts introduced above by two examples. The first example involves hyphenation in German. As the figure shows, orthographic anomalies such as 'ck' > 'k-k' at hyphenation (which incidentally was abandoned in the German spelling reform of 1996) can well be implemented within the confines of our formalisation. We assume the hyphen does not incur an additional position within the tier. The second example illustrates how a group of hieroglyphs can be broken up. As both examples illustrate, $width(t, b, e)$ remains invariant when tier t is broken up at a position $a > e$. We are not aware of any writing systems that would invalidate this assumption.

It should be emphasised that the above notions depend on the input linguistic resources as well as on requirements of the implementation and application. For example, if interlinear text is to be printed on paper, requiring a compact representation, one may choose to allow more breakable positions than if one displayed the interlinear text on a large computer screen. In addition, the width and advance functions depend on the font families and the font sizes used, which a reasonable implementation would allow the user to adjust according to personal preferences.

3.2. *Alignment*

Interlinear text may offer much benefit over individual linguistic resources provided the tiers are accurately aligned. That is, each section should contain no more and no less than all ranges of all tiers corresponding to one consecutive part of the

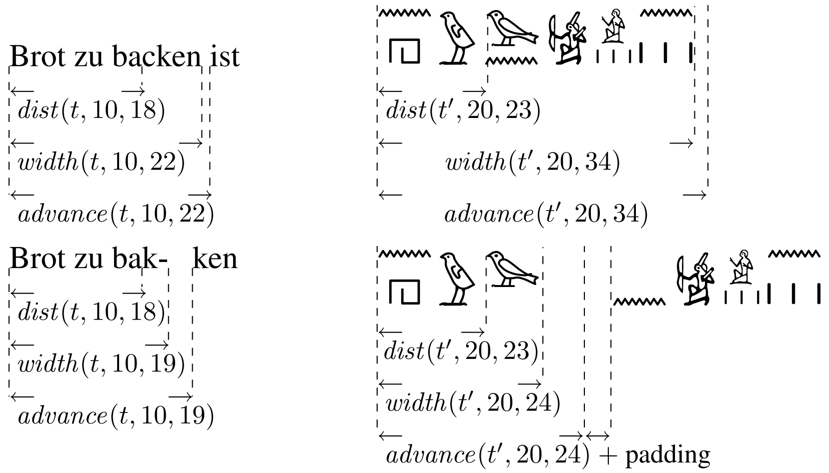


Figure 3. *Illustration of the functions $dist$, $width$ and $advance$. On the left, a German word is hyphenated at position 19, assuming that $Breakable(t, 19)$ holds, where t is a tier containing the text* 'Brot zu backen' *between positions 10 and 22. The symbol following position 18 can be either 'c' or 'k'. On the right, hieroglyphic text is broken up within a group. The distance to the symbol following position 23 (the sparrow) remains the same, as our framework demands. Note that $advance(t', 20, 24)$ includes the* minimal *distance between ranges $(t', 20, 24)$ and $(t', 24, 34)$, but the second of these may appear further to the right due to padding, as depicted.*

manuscript. A secondary requirement is that phrases within a section are accurately aligned. For example, a gloss of a word should be aligned to be exactly below the corresponding occurrence of the word in the encoded manuscript. We have chosen a formalisation that allows both types of requirements to be represented in simple ways, allowing a relatively straightforward algorithm to compute a satisfactory interlinear representation.

The requirements will be expressed in terms of two related types of links between positions in different tiers, which we will call *precedence links*. The most general of the two types we will call *diagonal precedence*, and we write $(t, a) \preceq_d (t', a')$ for a diagonal precedence link, or *d-link* for short, between positions (t, a) and (t', a'). This link expresses that position (t', a') must appear in the same section as position (t, a), or in a following section. However, there is no restriction on the exact horizontal placement. This means that if both positions occur in the same section, then (t', a') may appear to the right of (t, a), below or above (t, a), or to the left of (t, a). An imaginary line drawn between them may be slanted, which is why we refer to 'diagonal' links.

In typical usage, a d-link $(t, a) \preceq_d (t', a')$ may occur together with a reverse such link $(t', a') \preceq_d (t, a)$, which means that both positions must occur in the same section.

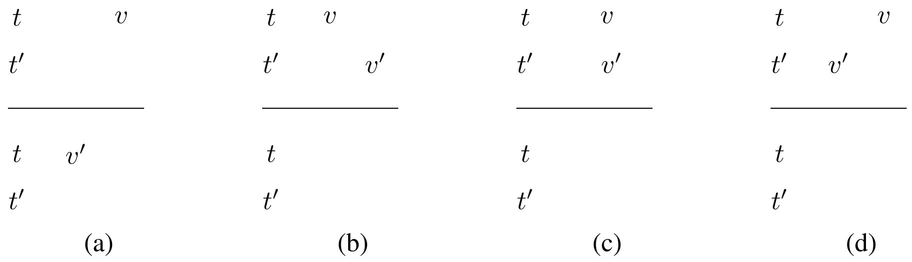


Figure 4. *Assume v and v' are symbol occurrences in tiers t and t', and a and a' are the positions just preceding v and v', respectively. The d-link $(t, a) \preceq_d (t', a')$ allows all of the renderings in interlinear text sketched in (a) through (d); in (a), v and v' occur in separate sections, the first preceding the second. If we add $(t', a') \preceq_d (t, a)$, then (a) is excluded. If the only precedence link is the v-link $(t, a) \preceq_v (t', a')$, then (a) through (c) are allowed. If we add $(t', a') \preceq_v (t, a)$, then (c) becomes the only allowable rendering.*

In order to motivate this usage, let us consider the first section of Figure 1. Depending on the application, we may want the transcription in the first line to appear without extra whitespace between hieroglyphs. Similarly, we may want the transliteration in the second line without padding. This means that words in transliteration are not aligned below corresponding hieroglyphs, and we may not wish such an alignment. However, we do want each word in transliteration to appear in the same section as the corresponding hieroglyphs. This can be expressed by having a pair of d-links $(t, a) \preceq_d (t', a')$ and $(t', a') \preceq_d (t, a)$, where (t, a) is the first letter of a word in the transliteration and (t', a') is the first corresponding hieroglyph.

The second type of link is a *vertical precedence link*, or *v-link*, which imposes a strictly stronger constraint. If there is a link $(t, a) \preceq_v (t', a')$, then we require that (t', a') appears in the same section as (t, a) or in a following section, just as in the case of a d-link; but in addition, if (t, a) and (t', a') appear in the same section, then (t', a') must appear at least as far from the left margin as (t, a), or in other words (t', a') must be printed below or above (t, a) or further to the right. Note that if both $(t, a) \preceq_v (t', a')$ and $(t', a') \preceq_v (t, a)$, then both positions must appear in the same section with one printed exactly above the other, which is why we refer to 'vertical' links.

The meanings of the two types of precedence links are rendered graphically in Figure 4, showing allowable interlinear text in the presence of one or more links of either of the two types.

It is important to realise that positions involved in links do not necessarily concur with breakable positions. This leads to an extra complication for an algorithm creating interlinear text. A typical example is presented by Figure 5, which shows an

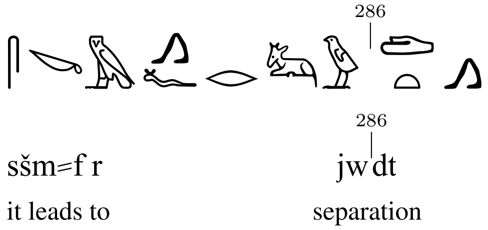


Figure 5. *Here* jwdt *is a word in transliteration, which is preferably not to be broken up, despite the line break, with line number 286, that occurs in the manuscript at the indicated position. Assuming that the application requires vertical alignment of the line numbers between hieroglyphic transcription and transliteration, the only solution is to insert padding before the word.*

Ancient Egyptian word in transliteration, with a number indicating a line break in the manuscript.

The same example also allows us to argue that d-links $(t, a) \preceq_d (t', a')$ may be used without the exact reverse $(t', a') \preceq_d (t, a)$. Assume the chosen translation of jwdt is ‘separation’. One may wish to align that word in translation roughly below the printed line number 286. However, there is no specific letter within the word ‘separation’ that corresponds to the beginning of line 286. Acceptable alignments are therefore such that the first letter ‘s’ appears below or to the left of the line number, *and* the last letter ‘n’ appears below or to the right of the line number.

Concretely, suppose the position just preceding ‘s’ in ‘separation’ is a_s and the position just preceding the ‘n’ is a_n, with the translation in tier t_t. Further suppose that a_h is the position just preceding the marker of the line number in the hieroglyphic tier t_h. The desired alignment is now expressed as the combination of $(t_t, a_s) \preceq_d (t_h, a_h)$ and $(t_h, a_h) \preceq_d (t_t, a_n)$.

As in the case of the mathematical notions introduced in the previous section, the two types of precedence links need not follow uniquely from the input linguistic resources, but are also determined by the implementation and application. Where an input file annotates consecutive phrases of the manuscript, indicating how each part of the annotation (e.g., a phrase in translation) matches to each phrase of the manuscript, then this information may well be cast in terms of v-links between the first symbol of (the electronic encoding of) the manuscript and the first symbol of the translation, for each phrase.

Pairs of d-links, one of which is the reverse of the other as discussed above, may typically be obtained through automatic alignment of tiers by auxiliary tools. Note that where such an alignment tool makes a mistake, the adverse consequences for the quality of the interlinear output are minimal. Incorrectly placed d-links may in the worst case lead to undesirable section breaks, but they would *not* lead to insertion of spurious horizontal space between symbols.

4. Solving the constraints

Before we can define an algorithm creating interlinear text based on the concepts introduced in the previous section, we first need to determine what constitutes an acceptable solution to the various constraints.

4.1. *Consistent formatting*

Given are a set $T = \{t_1, \ldots, t_{|T|}\}$ of tiers and appropriate functions $length$, $width$, $advance$, $dist$ and predicate $Breakable$. Furthermore, a page width w is given (expressed in pixels or points where appropriate).

In the formal treatment below, we will assume that each section is exactly w wide, and we imagine the sections as being arranged one immediately next to the other, rather than one below the other. As a consequence, we can use a single number to simultaneously identify a section and a horizontal location within that section. Numbers m such that $0 \leq m < w$ denote horizontal locations within the first section, with 0 representing the left edge and w representing the right edge. Numbers m such that $w \leq m < 2 \cdot w$ denote locations within the second section, etc. If there are σ sections, then locations can be between 0 and $\sigma \cdot w$, the upper boundary not included. For each such location m there is precisely one non-negative integer s such that $(s-1) \cdot w \leq m < s \cdot w$, which indicates that location m falls within the s-th section.

For σ a non-negative integer, we define a *formatting* in σ sections:

– For each tier $t_i \in T$, $1 \leq i \leq |T|$, there is a list of ρ_i ranges of the form $r_{i,j} = (t_i, b_{i,j}, e_{i,j})$, $1 \leq j \leq \rho_i$.

– Each such range $r_{i,j}$ is associated with a number $m_{i,j}$ such that $0 \leq m_{i,j} < \sigma \cdot w$. This simultaneously identifies the section and the location within that section where the range is printed, as explained above.

A formatting is called *consistent* if the following conditions hold. First, for each tier, the list of ranges should cover the complete tier, which means:

– $b_{i,1} = 0$, for each i,

– $e_{i,\rho_i} = length(i)$, for each i,

– $e_{i,j} = b_{i,j+1}$ for each i and j with $1 \leq j < \rho_i$.

Second, no range should be printed such that it runs into the right margin, or in other words if the left end falls within the s-th section, then the right end falls within the same section:

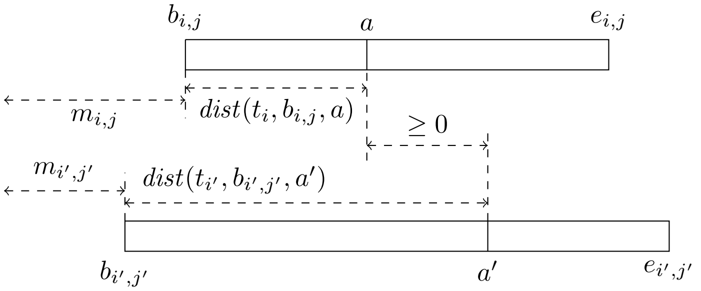


Figure 6. *The constraint imposed by* $(t_i, a) \preceq_v (t_{i'}, a')$*, with two positions within ranges* $r_{i,j} = (t_i, b_{i,j}, e_{i,j})$ *and* $r_{i',j'} = (t_{i'}, b_{i',j'}, e_{i',j'})$ *that are printed in the same section*

– $(s-1) \cdot w \leq m_{i,j} < s \cdot w$ implies $m_{i,j} + width(t_i, b_{i,j}, e_{i,j}) \leq s \cdot w$.

Third, if two consecutive ranges are printed within the same section, the second should be printed away from the first by the required minimum distance:

– $(s-1) \cdot w \leq m_{i,j} < s \cdot w$ and $(s-1) \cdot w \leq m_{i,j+1} < s \cdot w$ together imply $m_{i,j} + advance(t_i, b_{i,j}, e_{i,j}) \leq m_{i,j+1}$.

Fourth, both d-links and v-links between two positions constrain the first to appear in the same section as the second, or in an earlier section. Formally, if all of the following hold:

1) $(t_i, a) \preceq_d (t_{i'}, a')$ or $(t_i, a) \preceq_v (t_{i'}, a')$,

2) $b_{i,j} \leq a < e_{i,j}$ and $b_{i',j'} \leq a' < e_{i',j'}$,

3) $(s-1) \cdot w \leq m_{i,j} < s \cdot w$ and $(s'-1) \cdot w \leq m_{i',j'} < s' \cdot w$,

then $s \leq s'$. The first condition above states that the constraint holds both for d-links and v-links. The second condition identifies the ranges $r_{i,j}$ and $r_{i',j'}$ to which the relevant positions belong. The third condition identifies the s-th and s'-th section where the ranges are located.

Lastly, a v-link imposes an additional constraint if both positions appear in the same section. The second should then be printed at the same location as the first, or further to the right. This is illustrated in Figure 6. Formally, if all of the following hold:

1) $(t_i, a) \preceq_v (t_{i'}, a')$,

2) $b_{i,j} \leq a < e_{i,j}$ and $b_{i',j'} \leq a' < e_{i',j'}$,

3) $(s-1) \cdot w \leq m_{i,j} < s \cdot w$ and $(s-1) \cdot w \leq m_{i',j'} < s \cdot w$,

then $m_{i,j} + dist(t_i, b_{i,j}, a) \leq m_{i',j'} + dist(t_{i'}, b_{i',j'}, a')$.

4.2. *Preferences in formatting*

There may be many consistent formattings, and there are a number of criteria to choose from among them. First, we may want to minimise the penalty in a section. In the s-th section this is the sum of the penalties for each of the tiers; for tier i this is $\mathit{penalty}(t_i, e_{i,j})$ where $r_{i,j}$ is the last range in that section, or formally: $(s-1) \cdot w \leq m_{i,j} < s \cdot w$ and either $j = \rho_i$ or $m_{i,j+1} >= s \cdot w$.

For the design of our algorithm, we made three further assumptions, however:

1) We try to minimise the numbers ρ_i of ranges, for each i. In other words, breaking up ranges into smaller ranges is to be avoided unless this is necessary to satisfy the constraints imposed by precedence links.

2) All ranges are shifted to the left as much as possible.

3) In the absence of precedence links, the respective tiers are filled up with text of roughly comparable width.

The second assumption is uncontroversial, as by convention we start reading text from the left side of a page. The first constraint may be relaxed, however, for example in order to create a more spacious layout, if this is desired by the application.

The third constraint may be most relevant when few precedence links are available. In such a situation it is not unreasonable to assume that appropriate sections are obtained by taking chunks of roughly equal width from each of the tiers. An appropriate measure of width in this context may be relative to how much horizontal space a type of tier normally consumes. For example, a hieroglyphic transcription is normally wider than a transliteration of the same text. This refinement will not be used in what follows, however.

4.3. *Algorithm*

With the above assumptions, we can formulate an algorithm that computes a consistent formatting by filling up a section from left to right, ensuring the precedence links are respected at each step. When the page width is exhausted, the intermediate state with the lowest sum of penalties is chosen. The process then continues with the remaining parts of the tiers and a fresh section, until all tiers have been exhausted.

Before presenting the algorithm in more detail, we first define a range (t, b, e) to be *unbreakable* if $\mathit{Breakable}(t, b)$ and $\mathit{Breakable}(t, e)$ but not $\mathit{Breakable}(t, a)$ for any a with $b < a < e$. We say an unbreakable range (t, b, e) precedes an unbreakable range (t', b', e'), denoted by $(t, b, e) \prec (t', b', e')$, if one of the following two conditions hold:

– there is a precedence link between a position in the first range and a position in the second range, or more precisely, $(t, a) \preceq_v (t', a')$ or $(t, a) \preceq_d (t', a')$, for some a and a' with $b \leq a < e$ and $b' \leq a' < e'$; or

– the first range is just before the second in the same tier, or formally, $t = t'$ and $e = b'$.

It is self-evident that if (t, b, e) and (t', b', e') are two unbreakable ranges that have not yet been added to any section, and if $(t, b, e) \prec (t', b', e')$, then adding (t', b', e') to the current section should be accompanied by also adding (t, b, e) to the same section, otherwise one of the constraints from Section 4.1 would be broken.

In formal terms, the precedence relation $\prec$ corresponds to a directed graph, in which the strongly connected components (SCCs) are sets of unbreakable ranges that must be added in conjunction. Furthermore, for two such strongly connected components S and S' we define $S \prec S'$ if $(t, b, e) \prec (t', b', e')$ for some $(t, b, e) \in S$ and $(t', b', e') \in S'$. The relevance to our algorithm is that $S \prec S'$ means that the ranges in S must be added to a section before those in S' are added. See (Cormen *et al.*, 1990) for finding SCCs in a graph.

On a high level of abstraction, the algorithm is as follows:

1) Initiate a fresh section and an empty set of intermediate states for that section.

2) From all SCCs consisting of unused unbreakable ranges, take one SCC that is minimal with respect to the relation $\prec$. (If there are several, see below.) If no such SCC exists (i.e. the tiers are exhausted), then jump to step 4.

3) Add the ranges from the chosen SCC to the current section, resolving constraints on the locations (see below). The result is recorded as an intermediate state. If the constraints cannot be resolved within the available page width, then continue to step 4. Otherwise repeat from step 2.

4) From all intermediate states for the current section, output the one with the lowest penalty. Unless the tiers are now all exhausted, repeat from step 1.

The remaining issues will be discussed on the basis of the example in Figure 7. The unused ranges are divided into strongly connected components $\{r_6, r_7, r_9\}$, $\{r_8\}$, $\{r_{10}\}$ and $\{r_{11}, r_{12}\}$. Of these, $\{r_8\}$ and $\{r_{10}\}$ are not minimal with respect to $\prec$, as r_8 follows r_7 in the first tier, and r_{10} is preceded by r_9 via a d-link.

Of the two minimal SCCs, our implementation would now select $\{r_6, r_7, r_9\}$ to be handled in the next step, rather than $\{r_{11}, r_{12}\}$. The reason is that the right end of the range r_4 appears further to the right than either r_1 and r_2. As explained before, we prefer to fill all tiers with roughly the same amount of text, in the absence of other constraints.

The next step consists in resolving the constraints on the locations m_6, m_7 and m_9 needed to add the ranges from $\{r_6, r_7, r_9\}$ to the current section. Because r_6 follows r_1 and r_7 follows r_6 in the first tier, we naturally have:

$$\begin{array}{rcl} m_1 + \mathit{dist}(t_1, b_1, e_1) & \leq & m_6 \\ m_6 + \mathit{dist}(t_6, b_6, e_6) & \leq & m_7 \end{array}$$

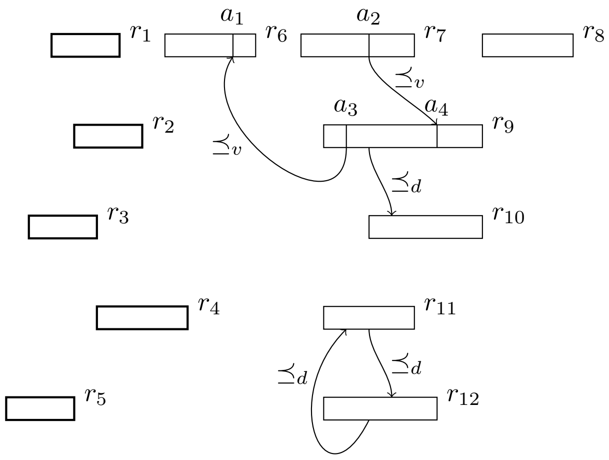


Figure 7. *Depicted are five tiers. The rectangles represent ranges $r_i = (t_i, b_i, e_i)$. Note that $t_1 = t_6 = t_7 = t_8$ and $t_2 = t_9$, etc. Furthermore, $e_1 = b_6$, $e_6 = b_7$, etc. The leftmost range r_i of each tier ($1 \leq i \leq 5$) has already been added to the section and its location m_i is fixed. The other ranges r_i ($6 \leq i \leq 12$) are still to be placed, and their locations m_i are variables constrained by the v-links between positions a_j ($1 \leq j \leq 4$) that occur within ranges. The d-links contribute to the formation of SCCs but do not impose restrictions on the locations m_i.*

In addition, the v-links connecting r_9 with r_6 and r_7 impose:

$$\begin{aligned} m_9 + \mathit{dist}(t_9, b_9, a_3) &\leq m_6 + \mathit{dist}(t_6, b_6, a_1) \\ m_7 + \mathit{dist}(t_7, b_7, a_2) &\leq m_9 + \mathit{dist}(t_9, b_9, a_4) \end{aligned}$$

It is now straightforward to find the minimal choices of m_6, m_7 and m_9 that satisfy these inequalities. Our implementation applies a heuristic to relax the constraints if no solution exists (note the circular dependency between m_6, m_7 and m_9 in the example).

There is another small complication, illustrated by the following. Suppose the obtained solution gives us m_6 such that $m_1 + \mathit{dist}(t_1, b_1, e_1) < m_6$ but $m_6 < m_1 + \mathit{advance}(t_1, b_1, e_1)$. In other words, r_6 would be too far away from r_1 to connect the two ranges into one joint range (t_1, b_1, e_6) but r_6 would not be far enough from r_1 to respect the minimal distance between the two ranges as expressed by the $\mathit{advance}$ function. Our solution in this case is to join the two ranges, creating a small error in the location of positions as dictated by v-links. We have not found any evidence that this could pose a problem in practice.

Lastly, we would like to point out that the implementation must be robust against flawed alignment algorithms that create cycles of d-links, to form SCCs stretching long parts of the text. Our implementation solves this by only considering the first three unused unbreakable ranges in each tier.

5. Implementation

The earliest experiments with automatic creation of interlinear text for Egyptology were reported by Nederhof (2002). The application was language teaching. Students could contribute translations of an Ancient Egyptian text through the internet, which were then automatically aligned with the hieroglyphic text and presented on a web page, to allow comparison of different interpretations. Although our project currently focuses on use by scholars, teaching remains an important motivation for our work.

The current implementation is an open-source Java package, which can be downloaded in its entirety including a few dozen sample texts, or accessed as an applet on a web page, via: `http://www.cs.st-andrews.ac.uk/~mjn/egyptian/texts/`.

We consider the specific data formats of the implementation to be of minor importance with respect to the preceding sections. A wide range of other textual data formats would be suitable as well, as long as certain key concepts in our formalisation, such as precedence links, can be extracted automatically or semi-automatically out of the input resources. However, a few remarks are in order.

First, breakable positions and penalties, as well as some v-links, need not be specified explicitly, but rather follow implicitly from the input data and some preferences determined by the user or developer of the software. For example, in an XML file one can specify the transliteration and translation of phrases, as in:

```
<phrase>
<transliteration>xbA hp</transliteration>
<translation>If law is laid waste</translation>
</phrase>
<phrase>
<transliteration>HD tp-Hsb</transliteration>
<translation>and order destroyed,</translation>
</phrase>
```

This may induce v-links between the beginnings of phrases in transliteration and the beginnings of the same phrases in translation, as well as specify that the beginnings of both parts of the respective tiers are breakable positions with a fixed low penalty.

Second, precedence links need not refer to absolute positions but may instead be indicated by labelling substrings of the tiers or by using anchor points that naturally arise in textual resources. For example, an input XML file may contain:

```
<transliteration>^mrw sA <line id="48"/> ^rnsj</transliteration>
<translation>
  <align id="48">Rensi, son of Meru.</align>
</translation>
```

This is a typical example of a difference in word order between two languages. In the transliteration the name Meru occurs before a line break and the name Rensi occurs after the line break, while the translation has the names in reverse order. This means that the line number in the transliteration does not correspond to a unique position in the translation. Instead, the XML tag ‘align’ induces v-links relating the position of the line number 48 to the beginning and end of the words in translation, similarly to what we have seen in our discussion of Figure 5.

6. Other text directions

In the above sections we have assumed that the direction of the encoded texts and their annotations is always left-to-right. In order to deal with writing systems that are inherently right-to-left, a few changes to the framework would be required, which can be supported by existing printing practices for philology.

The least common solution is to adapt the directionality of the annotation to that of the manuscript, to make all right-to-left. There are a few instances of this in (Daniels and Bright, 1996), where transliterations of, for example, Hebrew in the Latin script are written below a sample of Hebrew, letter by letter, starting at the right edge of a page. Outside of the study of writing systems, however, this practice seems to be rare.

A more common form of interlinear text separates the page into a left half and a right half, one presenting right-to-left text and the other presenting left-to-right text. Alternatively, one may split a section vertically into an upper part with right-to-left text and a lower part with left-to-right text, or vice versa depending on the application. Generalisation to several columns and several vertical parts is possible as well.

Our algorithm can be easily adapted to produce such forms, simply by disallowing v-links between any pair of tiers with opposite text directions. This means that the algorithm will proceed without ever comparing horizontal locations between tiers with opposite text directions. The handling of d-links remains unaltered. For representation in columns, one would need to split value w for the page width into two values w_1 and w_2 for the widths of the respective columns. The choice of w_1 and w_2 can be fixed for the text, or can be variable from one section to the next, with termination of a section when $w_1 + w_2$ exceeds a threshold.

7. Conclusions

The purpose of this paper is to investigate the creation of interlinear text, minimising the manual effort needed to bring existing textual resources into a suitable form, and maximising the usefulness of the interlinear text to the scholar who wants to study and compare the various resources. We have given a formal framework in which many constraints on interlinear text can be expressed and we have proposed an algorithm to solve those constraints.

An implementation exists that provides proof of concept, showing that the ideas can be realised and are useful. Refinements are being investigated, such as improved statistical models to obtain d-links out of automatic alignment of hieroglyphic and transliteration. We are also pursuing automatic alignment of, say, German and English translations, using off-the-shelf techniques developed in the area of machine translation. Further investigations will need to look at scripts with an inherent right-to-left text direction.

Acknowledgements

This article was written with the generous assistance of a fellowship from the Leverhulme Trust.

Anonymous reviewers have provided many helpful comments. I am indebted to Serge Rosmorduc for exchanging ideas, and for his help with the translation of the abstract.

8. References

Bird S., Liberman M., "A formal framework for linguistic annotation", *Speech Communication*, vol. 33, p. 23-60, 2001.

Cormen T., Leiserson C., Rivest R., *Introduction to Algorithms*, The MIT Press, 1990.

Daniels P., Bright W. (eds.), *The World's Writing Systems*, Oxford University Press, 1996.

Gale W., Church K., "A Program for Aligning Sentences in Bilingual Corpora", *Computational Linguistics*, vol. 19, No. 1, p. 75-102, 1993.

Lichtheim M., *Ancient Egyptian Literature — Volume I: The Old and Middle Kingdoms*, University of California Press, 1975.

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

Nederhof M.-J., "Automatic Alignment of Hieroglyphs and Transliteration", *in* N. Strudwick (ed.), *Information Technology and Egyptology in 2008, Proceedings of the meeting of the Computer Working Group of the International Association of Egyptologists*, Gorgias Press, p. 71-92, July, 2008.

Och F., Ney H., "A Systematic Comparison of Various Statistical Alignment Models", *Computational Linguistics*, vol. 29, No. 1, p. 19-51, 2003.

Parkinson R., *The Tale of the Eloquent Peasant*, Griffith Institute, Ashmolean Museum, Oxford, 1991.

Parkinson R., *The Tale of Sinuhe and Other Ancient Egyptian Poems 1940-1640 BC*, Oxford University Press, 1997.