

A probabilistic model of Ancient Egyptian writing

Mark-Jan Nederhof and Fahrurrozi Rahman
School of Computer Science
University of St Andrews
United Kingdom

ABSTRACT

This article offers a formalization of how signs form words in Ancient Egyptian writing, for either hieroglyphic or hieratic texts. The formalization is in terms of a sequence of sign functions, which concurrently produce a sequence of signs and a sequence of phonemes. By involving a class of probabilistic automata, we can define the most likely sequence of sign functions that relates a given sequence of signs to a given sequence of phonemes. Experiments with two texts are discussed.

Keywords:
Ancient Egyptian,
writing systems,
language models

1

INTRODUCTION

Ancient Egyptian writing, used in Pharaonic Egypt, existed in the form of *hieroglyphs*, often carved in stone or painted on walls, and sometimes written on papyrus (Allen 2000). Hieroglyphs depict people, animals, plants and various kinds of objects and geographical features. A cursive form of Ancient Egyptian writing, called *hieratic*, was predominantly written on papyrus. Most hieratic symbols can be seen as simplified hieroglyphs, to such an extent that it is difficult for the modern untrained eye to tell what is depicted. Because hieratic handwriting varied considerably over time, with notable differences between regions and scribes, the creation of computer fonts for hieratic is problematic, and consequently scholars commonly resort to publishing hieratic texts in a normalized hieroglyphic font. Since Version 5.2,

Unicode contains a selection of 1071 hieroglyphs. Henceforth we will use the term *sign* to refer to a hieroglyph or a hieratic symbol.

The Ancient Egyptian language is in the family of Afro-Asiatic languages, which includes the Semitic languages (Loprieno 1995). As in writing systems of several Semitic languages (e.g. Hebrew, Arabic, Phoenician), only consonants are written. Modern scholars use 24 or 25 letters to transliterate Ancient Egyptian texts in terms of these consonants. Most are written as Latin characters, some with diacritical marks, plus aleph ʾ and ayin ʿ. An equal sign is commonly used to precede suffix pronouns; thus *sdm* means “to hear” and *sdm=f* “he hears”. A dot can be used to separate other morphemes; for example, in *sdm.tw=f*, “he is heard”, the morpheme *tw* indicates passive.


The Ancient Egyptian writing system itself is a mixture of phonetic and semantic elements. The most important are *phonograms*, *logograms* and *determinatives*. A phonogram is a sign that represents a sequence of one, two or three letters, without any semantic association. A logogram represents one particular word, or more generally the lemma of a word or a group of etymologically related words. A determinative is commonly written at the end of a word, following phonograms, to clarify the meaning of a word; in their most obvious use, determinatives disambiguate between homophones, or more precisely, different words consisting of the same consonants. In addition, there are *typographical* signs, for example, three strokes that indicate the plural form of a noun (also used for collective nouns). These and more classes of signs are discussed in detail in Section 2.

What makes automatic analysis of Ancient Egyptian writing so challenging is that there was no fixed way of writing a word, so that table-lookup is largely ineffective. Even within a single text, the same word can often be found written in several different ways. Moreover, one sign can often be used in different functions, e.g. as phonogram or as determinative. Some signs can be used as different phonograms with different sound values. Together with the absence of word boundary markers, this makes it even hard to segment a text into words.

Generalizing statements can be made about writings of words. Typically, either a word starts with a number of phonograms, covering all the letters of the stem, possibly some covered more than once, followed by one or more determinatives, or a word starts with a logogram, possibly followed by one or more phonograms, possibly fol-

lowed by one or more determinatives. More phonograms can follow the determinatives for certain suffixes. This coarse description is inadequate however to model the wide spectrum of writings of words, nor would it be sufficient to disambiguate between alternative analyses of one sequence of signs.

These factors motivate the search for an accurate and robust model that can be trained on data, and that becomes more accurate as more data becomes available. Ideally, the model should be amenable to unsupervised training. Whereas linguistic models should generally avoid unwarranted preconceptions, we see it as inevitable that our model has some knowledge about the writing system already built in, for two reasons. First, little training material is currently available, and second, the number of signs is quite large, so that the little training material is spread out over many parameters. The *a priori* knowledge in our model consists of a sign list that enumerates possible functions of signs and a formalization of how these functions produce words. This knowledge sufficiently reduces the search space, so that probabilistic parameters can be relatively easily estimated.

In our framework, a *sign function* is formally identified by the combination of (a) the one or more signs of its writing, (b) its class, which could be ‘phonogram’, ‘logogram’, ‘determinative’, etc., and (c) a sequence of letters or a description of a semantic value, depending on the class. One example is the phonogram function for sign  with sound value *r*. There is a logogram function for the same sign, with as value the transliteration of the lemma *r3*, which means “mouth”. A typographical function for the three strokes may have a semantic value ‘plurality or collectivity’.

The first attempt to systematically classify functions of signs in context may have been Schenkel (1984). The proposed system used a notation that is close to traditional transliteration, but with additional elements, capturing *some* functional aspects of *some* used signs. For example, for each determinative in the writing of a word, a superscript giving the name of the sign is added to the transliteration. Use of logograms was indicated by capitalizing letters of the stem in the transliteration. It is not possible however to reconstruct a complete hieroglyphic writing from an instance of this notation, and moreover this system does not seem to lend itself to formalization.

The problem we will address in the experiments is guessing the sign functions given the signs and the letters. This is related to the problem of automatically obtaining transliteration from hieroglyphic text. As far as we are aware, the earliest work to attempt this was Billet-Coat and Hérin-Aime (1994), which focussed on a multi-agent architecture to combine expert knowledge about signs, words and clauses. Another approach to automatic transliteration, by Tsukamoto (1997), used Unix applications such as ‘grep’ and ‘sed’. The approach by Rosmorduc (2008) used manually produced rewrite rules. Further work along these lines by Barthélemy and Rosmorduc (2011) used two approaches, namely cascades of binary transducers and intersections of multitape transducers, with the objective to compare the sizes of the resulting automata.

A more modest task is to automatically align given hieroglyphic text and transliteration, as considered by Nederhof (2008), who used an automaton-based approach with configurations, similar to that in Section 5, except that manually determined penalties were used instead of probabilities. As we will demonstrate, the use of probabilities allows training of parameters of the model.

Relating hieroglyphic texts and their Egyptological transliteration is an instance of relating two alternative orthographic representations of the same language. The problem of mechanizing this task is known as machine transliteration. For example, Knight and Graehl (1998) consider translation of names and technical terms between English and katakana, and Malik *et al.* (2008) consider transliteration between Hindi and Urdu. Another very related problem is conversion between graphemes and phonemes, considered for example by Galescu and Allen (2002).

Typical approaches to solve these tasks involve finite-state transducers. This can be justified by the local dependencies between input and output, that is, ultimately the transliteration can be broken down into mappings from at most n to at most m symbols, for some small n and m . For Ancient Egyptian however, it is unclear what those bounds on n and m would be. We therefore depart from finite-state methods, and propose a model that involves a tape, with a tape head that can jump left as well as right. This idea is reminiscent of alignment models of machine translation (Brown *et al.* 1993) and of the Operation Sequence Model (Durrani *et al.* 2015).




Sproat (2000) formulates the *Regularity* hypothesis, stating that orthographic processes can be realized in terms of finite-state methods. For Ancient Egyptian, he singles out two isolated phenomena, namely a particular writing of plurality (cf. Section 2.6) and honorific transposition (cf. Section 4). He argues that whereas their realization requires extra care, they can be realized in terms of finite-state methods nonetheless. He ignores more problematic phenomena however, such as phonetic complements (cf. Section 2.2) and phonetic determinatives (cf. Section 2.4), which are core elements of the writing system and form the main motivation for our non-finite-state automaton model. Thereby, Ancient Egyptian remains a significant challenge to the Regularity hypothesis.


In the sequel, we let ‘Egyptian’ refer to ‘Ancient Egyptian’. The structure of this paper is as follows. Section 2 explains in more detail the sign functions that are distinguished in our model of Egyptian writing. An annotated sign list couples sign functions to signs, as explained in Section 3. The annotated texts themselves, which were used for training and testing, are presented in Section 4. A formal model of Egyptian writing is the subject of Section 5, extended with probabilities in Section 6. Experiments are discussed in Section 7.

2 SIGN FUNCTIONS

In our formal framework, we distinguish the sign functions that are explained in the following sections. Except for ‘spurious’ functions, each function has exactly one value, specified at the end of each section.

2.1 *Logograms*


A *logogram* is a sign that represents a word, or more accurately, the lemma of a word, or possibly a group of etymologically related words with closely related meanings. Often a logogram depicts the word it represents. For example, the aforementioned sign  can be a logogram for *r3*, “mouth”. In other cases, a logogram may represent an idea that can be associated with the thing that is depicted, rather than the thing itself. For example,  depicts a (standing) leg, while its meaning is the word *bw*, “place”. A related example is the sign  depicting (walking) legs, with meaning *jw*, “to come”.



An example where we would include etymologically related words is the following. The sign  can literally mean the thing that is depicted, namely *bjt*, “bee”, but the same sign is used in much the same way for the etymologically related word *bjt*, “honey”.

The value of a logogram is the transliteration of the lemma that it represents.



2.2


Phonograms


Much of the Ancient Egyptian writing system evolved via the principle of *rebus writing* (Daniels and Bright 1996), that is, the use of a sign solely for its sound value, derived from one or more sounds that occur in the word expressing what the sign depicts. For example, from the logographic use of sign  for *bw*, “place”, the use as *phonogram* evolved, allowing it to represent the letter *b* in the writing of any word.

For each letter, there is at least one phonogram that represents that letter in isolation. We call such a phonogram *uniliteral*. There are also several dozens of phonograms for sequences of two or three letters. For example,  is a (biliteral) phonogram with sound value *wn* and  is a (triliteral) phonogram with sound value *tjw*.

A word is often written using several phonograms, which together cover some letters more than once. A uniliteral phonogram representing a letter that is also represented by a neighboring biliteral or triliteral phonogram is known as a *phonetic complement*; there are examples in Figure 3 that will be discussed later.

As pointed out by e.g. Schenkel (1984), it can be very hard to distinguish between logograms and phonograms, especially in the case of triliteral phonograms that can by themselves write a whole word. For example,  can stand for the word *whmt*, “hoof”, and in this use it is obviously a logogram, but it can also stand for the word *whm*, “to repeat”. (The *t* in *whmt* is the feminine ending.) It is plausible that the two words are etymologically related, as the depicted cloven hoof ‘repeats’ a toe. However, traditionally the use of  in “to repeat” is analyzed as phonogram, as if its use was motivated by accidental similarity of the pronunciations of the two words. We have adopted that view.

One more example is the sign , which is primarily used as logogram for *ntr*, “god”. It is also used in the writing of the word *sntr*,







“incense”, and one may naively interpret it as phonogram there. However, it is very likely that the sign is not merely chosen for its sound value, but for its semantic relationship to *ntr*, “god”, in combination with the causative prefix *s-*. An alternative etymology suggested by de Vartavan (2010) involves the verb *sn*, “to smell”, in combination with *ntr*, but either way, the sign  in *sntr* is best analyzed as logogram.





In later stages of Egyptian, some pairs of sounds from earlier stages merged together. As a result, the corresponding signs were sometimes confused. One example is the use of a sign representing the sound *t* for writing a word whose historical pronunciation had a different sound *t̄*. In our framework, we let the value of a phonogram be its historical sound value, regardless of how it is used. However, we follow Hannig (1995) in not distinguishing *s* from *z*. In Middle Egyptian, these two sounds had merged together to such an extent that even the (conservative) writing system treated both as largely exchangeable. Both sounds are therefore transliterated as *s*.

2.3

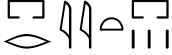


Determinatives


A *determinative* is a sign that derives a semantic value from what is depicted, much like many logograms. However, determinatives are not used in isolation to form writings of words. Instead they must be combined with logograms and phonograms together covering all the letters. Typically, determinatives occur at the end of a writing, following the logograms and phonograms.




Most determinatives do not pertain to any particular word. For example, the “tree” determinative  is used with various nouns related to trees, plants and wood. Another example is , depicting a papyrus scroll with ties, which is used as determinative for words that express abstract notions. Thus we have    , *jqr*, “excellent”, where the first three signs are each uniliteral phonograms for the three letters in that word.


The sign  can be used as determinative with the general meaning “man and his occupations”. For example, it is used in   , *šmsw*, “follower” (someone accompanying the king). Here the first sign is a logogram for the verb *šms*, “to follow” and the second sign is a

unilateral phonogram with value *w*, a suffix which turns the verb into a masculine noun.

The distinction between determinatives and logograms is illustrated by the word *pryt*, “settlement”, written as . The first sign (reading left-to-right and top-to-bottom for stacked signs) is a logogram depicting the plan of a house, with meaning *pr*, “house”, and derivatives. The next four signs are phonograms together covering the letters *r*, *y* and *t*. Note the *r* in *pryt* is covered by both the logogram and the first phonogram, which makes  here a phonetic complement. The second occurrence of  has a different function from the first. Here it is a determinative, clarifying that the written word has something to do with buildings.

A determinative may also be specific to one lemma. For example,  is generally used only for the noun *mnjt*, “mooring post”, and its derivatives. One may ask what distinguishes such a determinative from a logogram, which is by definition also specific to one lemma. The answer lies in the different roles that logograms and determinatives fulfil in the writing of words, as illustrated above for *pryt*, “settlement”.

When a determinative is specific to one lemma, the same sign can often be used as logogram as well, that is, the sign can be used to write a word without accompanying phonograms. For example,  can as logogram stand on its own for *hr*, “to fall”, but it is determinative in the alternative writing  , *hr*, where it is preceded by two unilateral phonograms.

The value of a determinative specific to a word is the transliteration of that word, such as *mnjt* for . The value of other determinatives is a general description of the kinds of concepts that are covered, such as “building, seat, place” for .

2.4

Phonetic determinatives

A *phonetic determinative* is similar to a determinative in that it tends to be placed near the end of a word, next to normal determinatives. However, its value is phonetic, repeating letters already written by logograms and phonograms.

An example is given by the writing of the word *mnḥt*, “splendid (fem.)” in Figure 1. The phonetic determinative 𓆎 here has phonetic reading *mnḥ*. Note that unlike the phonograms, it occurs near the end of the word, even following the feminine *t* ending.

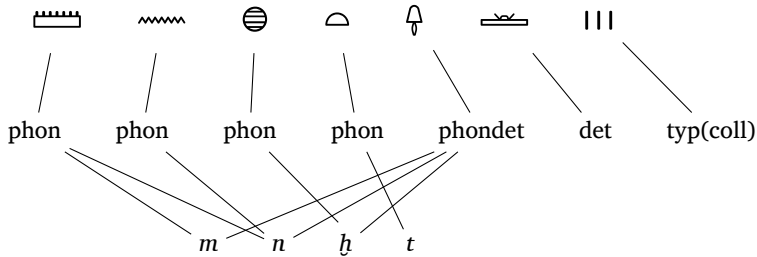


Figure 1:
Use of a phonetic
determinative

Many signs can be both phonograms and phonetic determinatives, even with the same sound value. We then classify an occurrence as the latter only if the corresponding letters have already been accounted for by earlier signs.


The value of a phonetic determinative is its (earliest historical) sound value.

2.5 *Typographical signs*

Signs that fall outside any of the classes above will be called *typographical*. One example is the single stroke written under, or next to, another sign, most often a logogram. Often its function is to indicate that the meaning of that other sign is what is depicted, rather than, say, the sound value of that sign. We then call the single stroke a *semogram marker*. For example, 𓆎 might mean *r3*, “mouth”, while 𓆎 , without semogram marker, might stand for the preposition *r*, “to”. The sign 𓆎 here is logogram or phonogram, respectively.

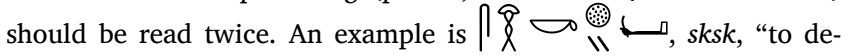
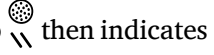
The function of the single stroke is not always clear however. More often than not, it acts as *space filler*; at this point we should explain that Egyptian writing is often influenced by aesthetical considerations, in particular the desire to fill up empty spaces between signs. As a consequence, 𓆎 can either mean *ḥr*, “face”, or *ḥr*, “on”. In the first case, the single stroke is clearly a semogram marker, but in the second it is merely a space filler.

Further typographical symbols consist of a combination of two or three strokes. These are typically written at the end of a noun as

marker of duality or plurality. (Egyptian had a dual form next to a plural.) The three strokes were however also used for singular nouns with collective meanings, such as *mnt*, “people” and *jmnt*, “what is hidden”. The three strokes are also written behind plural personal pronouns. Similarly, two strokes can be used for singular words whose meaning involves the idea of pairing two things or two people. An example is , *snw=f*, “his fellow”.

There are also *false dual* and *false plural* writings, with two or three strokes for words that happen to end on *-wj*, *-tj* or *-w*, the masculine and feminine dual and masculine plural endings, while these words are not grammatically dual or plural. In these cases the group of two or three strokes is analyzed as phonogram with sound value *wj*, *j* (without the feminine ending *t*) or *w*. It is not always easy however to determine whether words ending on *-wj/-tj/-w* are (historically) dual or plural.



Further typographical symbols include the numerals. We analyze a number written using a sequence of numerals as one sign function. Egyptian numerals are a topic by themselves (Ifrah 1981) and further discussion here would not be productive.

A peculiar typographical function exists in a combination of signs that indicates the preceding (phrase, word or sequence of letters) should be read twice. An example is , *sksk*, “to destroy”. Here the first three signs are phonograms together accounting for the first two letters *sk* of *sksk*. The following group  then indicates the letters *sk* should be read a second time.

As value of a typographical function we take a description, which can be, for example, “semogram marker”, “space filler”, “duality”, “plurality or collectivity”, “replaces human figure, or sign difficult to draw” and “number”.

2.6

Multiplication of signs

We discussed above that duality and plurality (and collectivity) can be expressed by two or three strokes. There is an alternative way to express the same, by repeating a sign once or twice. For example, the logogram  stands for *ntr*, “god”. By repeating it twice, we obtain , *ntrw*, “gods”. We recall *-w* is the masculine plural ending.

Typically only the last sign of a singular writing is repeated to obtain a dual or plural writing, but sometimes larger groups of signs are repeated. For example, $\overline{\text{r}} \overline{\text{n}}$ stands for *n*, “name”, written with two uniliteral phonograms for *r* and *n*, respectively. The plural can be written $\overline{\text{r}} \overline{\text{r}} \overline{\text{n}} \overline{\text{n}} \overline{\text{n}}$, *rnw*, “names”.

Also determinatives may be repeated. An example is the writing of “the two lands (Upper and Lower Egypt)” as $\overline{\text{t}} \overline{\text{w}} \overline{\text{j}}$. We recall $\overline{\text{w}} \overline{\text{j}}$ is the masculine dual ending. A typical writing for the singular is $\overline{\text{t}} \overline{\text{w}}$, “land”, written with a logogram for *t*, depicting a strip of land with three grains of sand, a semogram marker, and a determinative depicting irrigated land.







We have chosen a modeling of such writings that allows straightforward automatic processing. This consists in taking all repeated signs together to correspond to a single function indicating plurality. In the example of “names”, the first occurrences of $\overline{\text{r}}$ and $\overline{\text{n}}$ are analyzed as phonograms *r* and *n*, respectively, as they would be in the singular writing of the word. The two remaining occurrences each of $\overline{\text{r}}$ and $\overline{\text{n}}$ together indicate plurality. An example for duality, as illustrated in Figure 2a, will be discussed later.

As in the case of the dual and plural strokes (Section 2.5), there are false dual and false plural writings using duplication of signs. Common examples concern the nisbe form. A *nisbe* is an adjective derived from a noun by adding the ending *-j*. For example, $\otimes \overline{\text{n}} \overline{\text{j}} \overline{\text{w}} \overline{\text{t}}$ means “town” while $\overline{\text{n}} \overline{\text{j}} \overline{\text{w}} \overline{\text{t}} \overline{\text{j}}$ means “concerning the town; local”. The latter word is typically written as $\otimes \otimes$, which should be read “local” and not “the two towns”.

There are cases of plural and collective nouns that are written using three similar but *distinct* signs. For example, the word that means “cattle” can be accompanied by three determinatives depicting different kinds of cattle, and the word that means “birds” can be accompanied by three determinatives depicting different species of birds. These cases are rare enough to be ignored for the purposes of our model in Section 5. At this point we should emphasize that playfulness and creativity are important features of Egyptian writing, and this precludes existence of an exhaustive list of orthographic phenomena.


The value of a multiplicative function is a number, which can be 2 for dual and 3 for plural. In rare cases, we also find multiplicative functions for the numbers 4 and 9.




2.7 *The spurious functions*

Occasionally we find signs that do not have a clear function. Some can be plausibly attributed to scribal errors. There are also cases however for which a historical explanation can be given. For example, the two signs , representing crossing streets, and , the phonogram for *t*, are often written as one ‘frozen’ group. This makes sense in the writing of the word *njw*t, “town”, which has the (feminine) ending *-t*, with  being a logogram. However, where  is used as determinative with meaning “inhabited area” at the end of a masculine word (not ending on *-t*), we sometimes also find . We then classify  as spurious.

The spurious functions also contribute to creating a robust model. By interpreting some signs as ‘spurious’, the model can complete the analysis of a problematic writing as fall-back option if nothing else works. We return to this matter in Section 5.

2.8 *Combinations of signs having a function*

In the above, we have seen a few instances of a group of signs together having one function, in the case of multiplications of signs and in the case of typographical signs. Another example is , which together represents the logogram *twj*, “the two lands”. The signs in isolation represent two different plants, lily and papyrus, symbolizing Upper and Lower Egypt, respectively.

The group  has a single function as phonogram with sound value *nn*. An isolated  can only be a phonogram *nḥbt*. Similarly, the combination of signs  has a single function as a determinative for a “group of people”.

3 SIGN LIST

Essential to the application of our model is an annotated sign list. We have created such a list in the form of a collection of XML files.¹

¹ <http://mjn.host.cs.st-andrews.ac.uk/egyptian/unicode/>

Apart from being machine-readable, these files can also be converted to human-readable web pages. Among other things, the files contain knowledge about the various functions of the 1071 signs from the Unicode repertoire, gathered from a number of sources, the foremost of which is Gardiner (1957). The annotated sign list is necessarily imperfect and incomplete, which is due to inadequacies of the Unicode set itself (Rosmorduc 2002/3; Polis and Rosmorduc 2013), as well as to the nature of Ancient Egyptian writing, which gave scribes considerable freedom to use existing signs in new ways and to invent new signs where existing signs seemed inadequate. We have furthermore ignored the origins of signs, and distinguish fewer nuances of sign use than e.g. Schenkel (1971). See Polis and Rosmorduc (2015) for a revised taxonomy of hieroglyphic sign functions, motivated by the goal of compiling sign lists.

The items in our annotated sign list most relevant to this article each consist of:



- a sequence of signs (sometimes multiple sequences of alternative writings),
- a sign function class of that sequence,
- a sequence of letters or a semantic value, depending on the class.

As discussed in Section 2, a sign can often be both a logogram or a determinative specific to a lemma. Similarly, sometimes a sign can be both a phonogram or a phonetic determinative. To avoid duplication, we have created two combined classes. Thus, the sign list distinguishes the following:

- logogram, with the transliteration of a lemma,
- determinative, with a description of meaning,
- logogram / determinative, with the transliteration of a lemma,
- phonogram, with a phonetic value,
- phonetic determinative, with a phonetic value,
- phonogram / phonetic determinative, with a phonetic value,
- typographical, with a description of meaning.

Note that multiplication of signs and spurious signs are not included in the sign list, as these are not properties of the signs themselves but consequences of particular use.

Some signs can be used instead of other signs. This happens in particular where one sign is a graphical variant of another. In order to avoid redundancy, the sign list then only contains a listing of the sign functions for the most representative of two or more graphical variants, plus references from less representative to more representative variants. Such a reference can be automatically expanded into the relevant functions of the most representative sign. Also the two combined classes (logogram / determinative, and phonogram / phonetic determinative) can be split into the individual classes for the purposes of the model of Section 5.

The sign list contains (very rudimentary) information about the morphological structure of the lemmas written by logograms, in particular the stem and the gender (of nouns). The motivation is that this is necessary in order to match sign occurrences to transliterations. For example, the information that the word *nmtt*, “step”, denoted by the logogram , is feminine can be used to infer that uses of the logogram in plural writings should be matched to *nmtwt*, “steps”, with the feminine plural ending *-wt* in place of the feminine singular ending *-t*. Similarly, the logogram , for *hnj*, “to row”, is accompanied by information that its stem is *hn*, so we can identify the use in the writing of *hn=f*, “he rows”, without the weak consonant *j*, which disappears in most inflections.

4

CORPUS



There is currently only one comprehensive corpus of Late Egyptian, which is still under development (Polis *et al.* 2013). Corpora of Middle Egyptian, the object of our study, are scarce however. Moreover, we are not aware of any available corpora of hieroglyphic texts in which each sign is annotated with its function. One attempt in that direction was reported by Hannig (1995, p. XXXV), with the objective to determine the ratios of frequencies of four main classes of signs, using the first 40 lines of the text of Sinuhe.

It follows that in order to train and test our model, we had to create our own annotated corpus.² As yet, it is of modest size, including

²as part of the St Andrews corpus: <http://mjn.host.cs.st-andrews.ac.uk/egyptian/texts/>

just two classical texts, known as The Shipwrecked Sailor (Blackman 1932) and Papyrus Westcar (Blackman 1988). Disregarding damaged parts of the manuscripts, the segmented texts comprise 1004 and 2669 words, respectively.

For the convenience of annotation with sign functions, the texts were linearized, that is, information about horizontal or vertical arrangement of signs was discarded. Whereas the positioning of signs relative to one another can be meaningful, our current models do not make use of this; if necessary in the future, the exact sign positions can be extracted from another tier of annotation.

We normalized the texts by replacing graphical variants, such as  and , by a canonical representative, using machine-readable tables that are part of our sign list (Section 3). We also replaced composite signs by smallest graphemic units. For example, we replaced a single sign consisting of three strokes (typographical sign for plurality or collectivity) by three signs of one stroke each. Motivations for this include convenience and uniformity: in typeset hieroglyphic texts one may prefer to use three separate strokes and fine-tune the distance between them to obtain a suitable appearance.

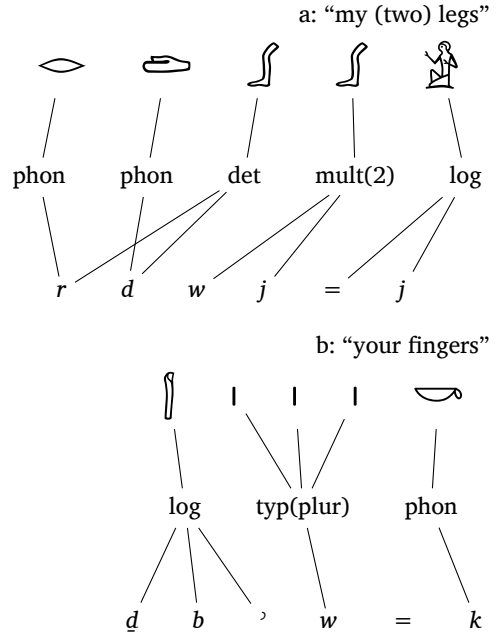
The texts were annotated with functions, using a customized, graphical tool. In this tool one can select known functions for signs, as present in the XML files mentioned in Section 3, but the tool also gives the option to create new functions that are not covered by the sign list. Many such functions were found during annotation.

A peculiar phenomenon in Egyptian writing is *honorific transposition*, which means that a sign or word is written first, even though its linguistic position is further to the end of a word or phrase. This applies in particular to gods and kings. For example, The Shipwrecked Sailor has $dw\textsubscript{3}.n=f n=j ntr$, “he thanked the god for me”, with the sign for ntr , “god”, written before the signs for $dw\textsubscript{3}.n=f n=j$. Where there is honorific transposition in the corpus spanning more than one word, all these words are put together in one *segment*. Apart from honorific transposition, a segment in the annotated corpus is simply one word.

For each word (or segment), the annotated corpus has:

- the sequence of functions, and
- the sequence of letters of the transliteration.

Figure 2:
Annotations in the corpus
(Shipwrecked Sailor)



The allowable functions are those listed in Section 2. Each function represents one or more signs, which are assumed to occur consecutively. Thereby the sequence of functions specifies the sequence of signs in the hieroglyphic writing. This was made possible by, among other things, our representation of the multiplicative functions (Section 2.6). An example is given in Figure 2a for *rdwj*, “pair of legs”. Whereas the first ‘leg’ sign of the writing is represented by a determinative function, the second such sign is represented by a multiplicative function with value ‘2’, that is, indicating duality.

Depending on their classes, functions may also represent letters, but due to such phenomena as phonetic complements, the sequence of letters of the transliteration is not determined uniquely by the sequence of functions. For this reason, the transliteration is present as separate tier, and functions are linked to the relevant letters of the transliteration, where applicable. In particular, phonograms and phonetic determinatives are linked in this way, and so are logograms and determinatives specific to words.

Also multiplicative functions may be linked to the letters of the dual/plural endings, as exemplified in Figure 2a. The same holds for

a: “he is seen”

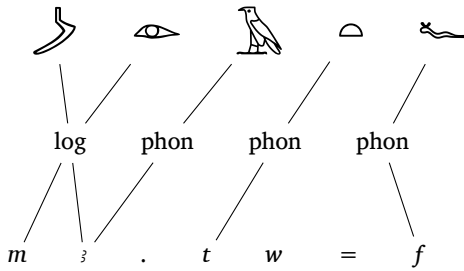
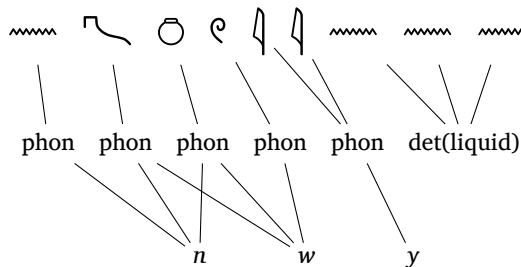


Figure 3:
Further annotations
(Shipwrecked Sailor)

b: “water”



the two or three strokes that indicate duality/plurality. An example is found in Figure 2b, for the plural of *ḏb*, “finger”. Recall that the masculine plural ending is *-w*. Not linked to letters are determinatives that are not specific to any word, as exemplified in Figure 3b.

In the diagrams, the values of the functions are abbreviated or omitted altogether to avoid clutter. For example, we do not explicitly indicate the sound values of phonograms, which usually follow from the links between functions and letters. Also the lemmas of logograms and determinatives specific to words are not shown in the diagrams. Note that these may not be equal to the relevant letters from the transliteration. For example, the lemma of the first function in Figure 3a is in fact *m33*, “to see”; the second *3* disappears in some verb forms. Recall that the morpheme *tw* indicates passive; in this writing the *w* is not written out.

Figure 3b is interesting in that it shows two phonetic complements: both the first and the fourth signs are uniliteral phonograms that cover the letters *n* and *w*, which are also covered by the second and third signs, which are both biliteral phonograms.

An essential document while annotating the corpus was the annotation manual, which helped to disambiguate contentious cases, of

which there were many. Examples of such cases were discussed in Section 2.2. We have as far as possible relied on conventional wisdom, but on several occasions we had to resort to informed guesses, making additions to the annotation manual to ensure consistency.

5

MODEL

In order to motivate our model, we investigate Figure 3a. If we string together the letters coming from the respective functions we obtain *m:ʒtʃ* rather than the correct transliteration *m:tw=f*. Similarly, for Figure 3b we would obtain *nnwnwwy*. There are two causes for this mismatch. The first is that letters can be written more than once, by several functions. In most cases this is done with phonetic complements, that is uniliteral phonograms, but we also find biliteral and triliteral phonograms as well as phonetic determinatives that cover letters already covered before. The second cause is that some letters in the transliteration, often weak consonants, are not represented by any signs at all. For pragmatic reasons, we will treat the Egyptological punctuation symbols, such as the period and the equal sign, on a par with weak consonants not written by signs.

For the second issue, our solution is to introduce an additional type of function, which we call *epsilon-phonogram*. Such a function acts much like a normal phonogram in the sense that a letter is produced in the transliteration, but it does not correspond to any sign (in other words, it corresponds to the empty, or epsilon string of signs).

For the first issue, that of letters covered several times, we conceive of the transliteration as being produced incrementally, in terms of a tape with a head that can move in both directions. In the simplest case, a function appends letters at the end of the tape, and moves the head a corresponding number of places to the right. This suffices for Figure 2b, as shown in Figure 4. The left column indicates the kind of function that is applied, omitting the associated signs, and the right column indicates the tape content, with the arrow marking the position of the head. Initially the tape is empty, and the tape head is at position 0. The logogram function then puts *ḏb* on the tape, moving the tape head to position 3. Subsequently, the typographical function appends a *w*, moving the head to position 4. After application of an epsilon-phonogram and a phono-

	↓	Figure 4:
log($\underline{d}b^{\prime}$)	$\underline{d} b^{\prime} \downarrow$	Computation for $\underline{d}b^{\prime}w = k$
typ(plur)	$\underline{d} b^{\prime} w \downarrow$	
eps-phon(=)	$\underline{d} b^{\prime} w = \downarrow$	
phon(k)	$\underline{d} b^{\prime} w = k \downarrow$	

gram function, = and k will have been appended and the head is at position 6.

The situation is only slightly more involved for Figure 2a. Here the determinative specific to rd should only be allowed if rd occurs at the beginning of the tape. This ‘lookback’ amounts to a check of validity of the computation, but it does not alter the fact that the tape is written strictly from left to right, and the tape head always moves rightward.

However, a different approach is needed for cases such as those in Figure 3, which involve phonograms that cover letters more than once, some appending more letters to the tape at the same time. Our solution is to add one more type of function, which we call *jump*. This decrements (or increments) the position of the head, so a string of letters can be written starting from a position other than the end of the tape. The computation for $m_{\mathfrak{z}}.tw = f$ is given by Figure 5. Here a jump one position back allows another occurrence of \mathfrak{z} corresponding to a phonogram, after \mathfrak{z} was already seen as part of the logogram. Recall that the second \mathfrak{z} of the lemma $m_{\mathfrak{z}}$, “to see”, is omitted in many verb forms. The second feature of ‘log($m_{\mathfrak{z}\mathfrak{z}}, m_{\mathfrak{z}}$)’ in our ad hoc notation attempts to convey that we are dealing with a particular use of this logogram that produces only the letters $m_{\mathfrak{z}}$ in the transliteration. For writing of other words, in which the full, geminated form is present,

	↓	Figure 5:
log($m_{\mathfrak{z}\mathfrak{z}}, m_{\mathfrak{z}}$)	$m_{\mathfrak{z}} \downarrow$	Computation for $m_{\mathfrak{z}}.tw = f$
jump(-1)	$m \downarrow \mathfrak{z}$	
phon(\mathfrak{z})	$m_{\mathfrak{z}} \downarrow$	
eps-phon(.)	$m_{\mathfrak{z}} . \downarrow$	
phon(t)	$m_{\mathfrak{z}} . t \downarrow$	
eps-phon(w)	$m_{\mathfrak{z}} . t w \downarrow$	
eps-phon(=)	$m_{\mathfrak{z}} . t w = \downarrow$	
phon(f)	$m_{\mathfrak{z}} . t w = f \downarrow$	

Figure 6:
Computation for *nwy*

	↓
phon(<i>n</i>)	<i>n</i> ↓
jump(-1)	↓ <i>n</i>
phon(<i>nw</i>)	<i>n w</i> ↓
jump(-2)	↓ <i>n w</i>
phon(<i>nw</i>)	<i>n w</i> ↓
jump(-1)	<i>n</i> ↓ <i>w</i>
phon(<i>w</i>)	<i>n w</i> ↓
phon(<i>y</i>)	<i>n w y</i> ↓
det(liquid)	<i>n w y</i> ↓

Figure 7:
Computation for *mnht*

	↓
phon(<i>mn</i>)	<i>m n</i> ↓
jump(-1)	<i>m</i> ↓ <i>n</i>
phon(<i>n</i>)	<i>m n</i> ↓
phon(<i>h</i>)	<i>m n h</i> ↓
phon(<i>t</i>)	<i>m n h t</i> ↓
phondet(<i>mnht</i>)	<i>m n h t</i> ↓
det(abstract)	<i>m n h t</i> ↓
typ(coll)	<i>m n h t</i> ↓

we could use alternatively 'log(m_{33} , m_{33})'. We will see more examples of functions having additional features later.

The computation for *nwy* is given by Figure 6. Here several jumps are needed to model that *n* and *w* are each covered by three different signs. Note that the determinative has a general description 'liquid' and so does not correspond to any letters.

The computation for *mnht* is given by Figure 7. As shown, application of a phonetic determinative does not require a jump. This is motivated by the observation that phonetic determinatives behave similarly to determinatives in that they tend to appear at the end of a word, even after phonograms for subsequent letters (cf. Figure 1). A phonetic determinative with a certain sound value is applicable if that value is a substring of the current content of the tape. Application of the function leaves the tape content and position of the head unchanged.

We impose two constraints on the use of jumps. The first is that jumps with positive values, moving the tape head rightward, should

not bring it beyond the end of the (written) tape. This is because the transliteration should be a sequence of letters without any gaps.

The second constraint is that no tape square that already contains a letter can be overwritten with a different letter. This is consistent with the application we are aiming to model, viz. Egyptian writing. This means for example that the first ‘phon(nw)’ in Figure 6 is applicable because the tape content to the right of the head, which is n , is a prefix of nw . Application of the function leaves that existing n unaffected and in addition appends the remaining suffix w at the end of the tape and moves the head to be after that suffix. In general, if the tape content to the right of the head is β , then we can apply a phonogram with value γ if:

- β is a prefix of γ (as in the case discussed above) or
- γ is a prefix of β (cf. phonogram for f in Figure 8a below).

Our aim is to complete the above framework to allow a sequence of functions to uniquely determine a sequence of signs and a sequence of letters. The sequence of signs is straightforwardly obtained as we already assumed from Section 1 onward that each function determines one or more consecutive signs. After having added epsilon-phonograms and jumps, we can now also account for letters not represented by signs, and for letters represented by several signs.

At least one more refinement remains to be explained. A phonogram for t or d is sometimes used for letters in a word that historically should have \underline{t} or \underline{d} , and vice versa; cf. the discussion in Section 2.2 about historical sound changes in Egyptian. Hence we sometimes need to give a phonogram an additional feature, so that for example ‘phon(t,\underline{t})’ indicates that t is the historical sound value of the sign, say \ominus , but the sign is used in the writing of a word whose transliteration has \underline{t} instead.

After this and other minor refinements, any sequence of functions corresponds to at most one analysis of a word, in terms of a sequence of signs, a sequence of letters, and the links between them, as exemplified in Figures 1–3, or in other words, in terms of the kinds of annotations that exist in our corpus. We also aim to achieve the converse, namely to translate an annotation of a word to a unique sequence of functions. Part of this is straightforward, as most of the functions and the order

in which they occur in a sequence are determined by the order of the signs. However, if there are no further restrictions, jumps may be inserted anywhere, even when they are not useful. In particular, they may be applied just before applying a determinative, even though a determinative does not depend on the input positions. In principle we could even apply a number of jumps in sequence, moving the head back and forth.

We solve this by demanding that jumps only occur just before application of a phonogram, or a related function whose application relies on the input position. The concrete realization is by a flag \mathbf{fl}_{jump} , which is set to **true** after a jump. As long as the flag is **true**, no determinative, phonetic determinative, or another jump is applicable. A phonogram and a few other functions reset the flag to **false**. For similar reasons, we use a flag \mathbf{fl}_{eps} that is set to **true** after application of an epsilon-phonogram. As long as this flag is **true**, no jump is allowed. The effect is that epsilon-phonograms are applied as late as possible. Two more flags, \mathbf{fl}_{fp} and \mathbf{fl}_{end} , will be discussed later.

To make the preceding more precise, we introduce the concept of *configuration*, which contains:

- the tape content preceding the head position, denoted by α ,
- the tape content from the head position onwards, denoted by β ,
- the values of the four flags.

Initially, the tape is empty, so $\alpha = \beta = \varepsilon$, where ε denotes the empty string, and all flags are **false**.

In a given configuration, only a subset of functions is applicable. For example, if $\alpha = \varepsilon$ and $\beta = n$, then a function $\text{phon}(nw)$ would be applicable, but not say a function $\text{phon}(t)$. The flags also restrict the applicable functions, as explained above. In general, every function has a *precondition*, that is, a set of constraints that determines whether it is applicable in a certain configuration, and a *postcondition*, which specifies how its application changes the configuration. The most important functions are characterized in this manner in Table 1, with tape content and position of the head as specified by α and β .

The precondition of a logogram for lemma γ says that γ must occur from the position of the head onward, possibly after a prefix of γ was written already, e.g. using phonograms. Furthermore, the position of the head must be either 0 or 1, and in the latter case,

Table 1: Preconditions and postconditions

Logogram for γ	
Pre	$\alpha = \varepsilon$ or (for causative; see main text) $\alpha = s$, β is prefix of γ , $\mathbf{fl}_{eps} = \mathbf{false}$.
Post	$\alpha := \alpha\gamma$, $\beta := \varepsilon$, $\mathbf{fl}_{jump} := \mathbf{false}$.
Phonogram with sound value γ	
Pre	γ is prefix of β or β is prefix of γ .
Post	$\alpha := \alpha\gamma$, if β was of the form $\gamma\delta$ then $\beta := \delta$ else $\beta := \varepsilon$, $\mathbf{fl}_{jump} := \mathbf{false}$, $\mathbf{fl}_{eps} := \mathbf{false}$.
Determinative not specific to any word	
Pre	$\mathbf{fl}_{jump} = \mathbf{false}$, $\mathbf{fl}_{eps} = \mathbf{false}$.
Post	-
Determinative specific to γ	
Pre	$\alpha\beta = \gamma\delta$ or (for causative) $\alpha\beta = s\gamma\delta$ for some δ , $\mathbf{fl}_{jump} = \mathbf{false}$, if $\mathbf{fl}_{eps} = \mathbf{true}$ then $\delta = \varepsilon$.
Post	$\mathbf{fl}_{eps} := \mathbf{false}$.
Phonetic determinative with sound value γ	
Pre	$\alpha\beta = \delta_1\gamma\delta_2$ for some δ_1 and δ_2 , $\mathbf{fl}_{jump} = \mathbf{false}$, if $\mathbf{fl}_{eps} = \mathbf{true}$ then $\delta_2 = \varepsilon$.
Post	$\mathbf{fl}_{eps} := \mathbf{false}$.
Spurious	
Pre	$\mathbf{fl}_{jump} = \mathbf{false}$, $\mathbf{fl}_{eps} = \mathbf{false}$
Post	-
Jump with value j	
Pre	$\mathbf{fl}_{jump} = \mathbf{false}$, $\mathbf{fl}_{eps} = \mathbf{false}$, $\delta = \alpha\beta$, $i = \alpha $, $0 \leq i + j \leq \delta $.
Post	$\alpha := \alpha'$, $\beta := \beta'$ for some α' and β' such that $\alpha'\beta' = \delta$ and $ \alpha' = i + j$, $\mathbf{fl}_{jump} := \mathbf{true}$.
Epsilon-phonogram for letter ℓ	
Pre	$\beta = \varepsilon$.
Post	$\alpha := \alpha\ell$, $\mathbf{fl}_{eps} := \mathbf{true}$.

the first letter on the tape must be *s*. This is because in Egyptian, the prefix *s-* can be used to derive causative verbs from other verbs. The writing may then consist of a phonogram for *s* followed by a logogram for the original verb. The postcondition for logograms says simply that γ is written to the tape and the head moves rightward by $|\gamma|$ positions.

The precondition of a phonogram with value γ was discussed before. The postcondition is slightly complicated by the need to distinguish between two cases, where γ is a prefix of β or where β is a prefix of γ (if $\gamma = \beta$, the two cases collapse).

The preconditions and postconditions of determinatives not specific to any words are straightforward. For a determinative specific to word γ , we merely need to check whether γ is present near the beginning of the tape, possibly after the causative prefix *s-*; if the previous function was an epsilon-phonogram, then γ must be a suffix of the tape content (recall that we want epsilon-phonograms to be applied as late as possible). Phonetic determinatives are similar, except that the required string γ need not occur near the beginning of the tape.

Spurious functions require that the previously applied function is not a jump or epsilon-phonogram. A jump with value j , which can be positive or negative, is allowed for current position i of the head provided the previously applied function was not a jump or epsilon-phonogram, and provided the new position $i + j$ is not preceding the beginning of the tape nor beyond the end of the tape. An epsilon-phonogram is only allowed if the head is at the end of the tape.

Our model has a number of specialized functions in place of the generic typographical functions as they occur in the corpus. For example, the three strokes, for ‘plurality or collectivity’, in the model correspond to three different functions with different preconditions and postconditions. First, the three strokes may be purely semantic, in the writing of a collective noun in singular form, where they do not represent any letters. This function behaves much like a determinative not specific to any word, except that it can only occur at the end of a word. For this reason, the flag fl_{end} is set to **true**. The purpose of this flag is to prevent that further letters are appended behind the end of the tape, until possibly an Egyptological ‘=’ symbol marks the end of the current word proper, before a suffix pronoun.

The plural strokes may also signify plurality in the grammatical sense, in which case it corresponds to the *-w* ending of masculine plural, or to the *-wt* ending of feminine plural. A separate function is needed for the two genders, both of which set fl_{end} to true. Apart from the flag fl_{end} , the function of three strokes for masculine plural has preconditions and postconditions identical to those of the phonograms. The case of feminine plural will be discussed further below.

Similarly, our model distinguishes between three uses of the multiplicative functions with value ‘3’, with different preconditions and postconditions. As in the case of the plural strokes, their meaning may be purely semantic, without a word being grammatically plural, or they may be used as markers of either masculine plural or feminine plural.

In our corpus we have linked functions marking plural only to the *w* from the ending, whether it is the *-w* ending of masculine plural or the *w* that is the first letter of the *-wt* ending of feminine plural. This is because the *t* of the feminine ending would normally be accounted for already by another sign, which could be a phonogram or logogram, as illustrated in Figures 8a and 8b.

a: “beautiful (women)”

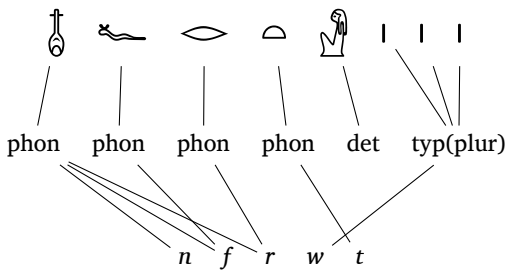
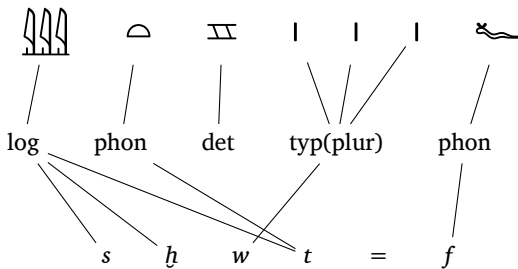


Figure 8:

Annotations of feminine plural words (Papyrus Westcar)

b: “its fields”



The same two examples also illustrate the challenge that feminine plural poses to a left-to-right automaton model. When the feminine t is written to the tape, the function justifying the w in front of the t is not seen until many steps later. The use of jumps to handle this seems inappropriate, as jumps were designed for phonetic complements. Another potential solution is to use lookahead, but this appears difficult to extend with probabilities.

We have chosen for a different solution, using the flag fl_{fp} , for ‘feminine plural’. This flag is set to **true** when a feminine plural is predicted by (nondeterministically) putting an extra w on the tape, in one of two cases. The first is if a logogram of a feminine word is seen, and the second is if a phonogram for t is seen.

The rest of the computation then has the obligation to reset fl_{fp} to **false**, and this can only happen if a function for plurality (either the three strokes or a multiplicative function with value ‘3’) is seen later. While $fl_{fp} = \mathbf{true}$, analysis of the current word cannot be completed.

Concretely for Figure 8a, we now have two functions ‘ $\text{phon}(t,t,\mathbf{false})$ ’ and ‘ $\text{phon}(t,wt,\mathbf{true})$ ’. Both correspond to a phonogram for the letter t (the first feature), but realized as t or wt in the transliteration (the second feature), while possibly predicting feminine plural (the third feature). The first function has the preconditions and postconditions of a normal phonogram (cf. Table 1), while the second writes wt on the tape instead of just t and sets fl_{fp} to **true**. The resulting computation is in Figure 9.

Similarly for Figure 8b, we now have two functions ‘ $\text{log}(sht,sht,\mathbf{false})$ ’ and ‘ $\text{log}(sht,shwt,\mathbf{true})$ ’. Both are logograms for the same sign for lemma sht (first feature) while they are realized differently in the transliteration (second feature), possibly predicting feminine plural (the third feature). The first function behaves like a normal logogram, but the second writes $shwt$ on the tape and sets fl_{fp} to **true**. The resulting computation is in Figure 10.

Our model presently has no special provisions for the phenomenon of honorific transposition (Section 4). This implies that accuracy is poor for the (few) cases of honorific transposition in the corpus. To address this, one may consider refinements of the model that allow ‘gaps’ in the hieroglyphic writing to be filled in later, along the lines of the Operation Sequence Model (Durrani *et al.* 2015).

	↓	$\mathbf{fl}_{fp} = \mathbf{false}, \mathbf{fl}_{end} = \mathbf{false}$
phon(<i>nfr</i>)	<i>n f r</i> ↓	$\mathbf{fl}_{fp} = \mathbf{false}, \mathbf{fl}_{end} = \mathbf{false}$
jump(-2)	<i>n</i> ↓ <i>f r</i>	$\mathbf{fl}_{fp} = \mathbf{false}, \mathbf{fl}_{end} = \mathbf{false}$
phon(<i>f</i>)	<i>n f</i> ↓ <i>r</i>	$\mathbf{fl}_{fp} = \mathbf{false}, \mathbf{fl}_{end} = \mathbf{false}$
phon(<i>r</i>)	<i>n f r</i> ↓	$\mathbf{fl}_{fp} = \mathbf{false}, \mathbf{fl}_{end} = \mathbf{false}$
phon(<i>t, wt, fp = true</i>)	<i>n f r w t</i> ↓	$\mathbf{fl}_{fp} = \mathbf{true}, \mathbf{fl}_{end} = \mathbf{true}$
det(<i>female</i>)	<i>n f r w t</i> ↓	$\mathbf{fl}_{fp} = \mathbf{true}, \mathbf{fl}_{end} = \mathbf{true}$
typ(<i>plur</i>)	<i>n f r w t</i> ↓	$\mathbf{fl}_{fp} = \mathbf{false}, \mathbf{fl}_{end} = \mathbf{true}$

Figure 9:
Computation for *nfrwt*

	↓	$\mathbf{fl}_{fp} = \mathbf{false}, \mathbf{fl}_{end} = \mathbf{false}$
log(<i>shwt, shwt, fp = true</i>)	<i>s h w t</i> ↓	$\mathbf{fl}_{fp} = \mathbf{true}, \mathbf{fl}_{end} = \mathbf{true}$
jump(-1)	<i>s h w</i> ↓ <i>t</i>	$\mathbf{fl}_{fp} = \mathbf{true}, \mathbf{fl}_{end} = \mathbf{true}$
phon(<i>t, t, fp = false</i>)	<i>s h w t</i> ↓	$\mathbf{fl}_{fp} = \mathbf{true}, \mathbf{fl}_{end} = \mathbf{true}$
typ(<i>plur</i>)	<i>s h w t</i> ↓	$\mathbf{fl}_{fp} = \mathbf{false}, \mathbf{fl}_{end} = \mathbf{true}$
phon(=)	<i>s h w t =</i> ↓	$\mathbf{fl}_{fp} = \mathbf{false}, \mathbf{fl}_{end} = \mathbf{false}$
phon(<i>f</i>)	<i>s h w t = f</i> ↓	$\mathbf{fl}_{fp} = \mathbf{false}, \mathbf{fl}_{end} = \mathbf{false}$

Figure 10:
Computation for *shwt=f*

6

PROBABILITIES

After having captured the relation between sequences of signs and sequences of letters solely in terms of sequences of functions, the next step is to estimate their probabilities. An obvious candidate is a simple *N*-gram model:

$$P(f_1^n) = \prod_i P(f_i | f_1^{i-1}) \approx \prod_i P(f_i | f_{i-N+1}^{i-1})$$

Here f_1, \dots, f_n is a sequence of functions, ending in an artificial end-of-word function, and f_i^j is short for f_i, \dots, f_j . In our experiments, estimation of $P(f_i | f_{i-N+1}^{i-1})$ is by relative frequency.

About 4000 functions are compiled out of the entries of the sign list. Added to this are dynamically created functions, such as numbers, epsilon-phonograms and jumps. Because little training material is available, this means a considerable portion of these functions is never observed, and smoothing techniques become essential. We use Katz's back-off (Katz 1987) in combination with Simple Good-Turing (Gale and Sampson 1995).

Functions are naturally divided into a small number of classes, such as the class of all phonograms and the class of all logograms. Using these classes as states, we obtain a second type of model in

terms of (higher-order) HMMs (Rabiner 1989; Vidal *et al.* 2005). For fixed N , and with c_i denoting the class of function f_i , we have:

$$P(f_i | f_{i-N+1}^{i-1}) \approx P(c_i | c_{i-N+1}^{i-1}) * P(f_i | c_i)$$

Estimation of both expressions in the right-hand side is again by relative frequency estimation, in combination with smoothing.

It should be noted that not all sequences of functions correspond to valid writings. Concretely, in the configuration reached after applying functions f_1^{i-1} , the preconditions of function f_i may not hold. As a result, some portion of the probability mass is lost in invalid sequences of functions. We see no straightforward way to avoid this, as the model discussed in Section 5, which allows jumps of the tape head, cannot be captured in terms of finite-state machinery.

7

RESULTS

In our experiments, the training corpus was Papyrus Westcar and the test corpus was The Shipwrecked Sailor. We have considered but rejected the possibility of taking two disjoint parts of both texts together as training and test corpora, for example taking all odd words from both texts for training and all even words for testing. The argument against this is that many words occur repeatedly in the same text, and therefore there would be a disproportionate number of words that occur in both training and test material, potentially leading to skewed results.

Our objective is now to guess the correct sequence of functions, given the sequence of signs and the sequence of letters of a word. We determined recall, precision, and F-measure, averaged over all words in the test corpus. This was done after removing jumps and epsilon-phonograms, so that we could take the annotations from the corpus as gold standard. We have also ignored how functions are linked to letters; the main motivation for this was to be able to define a suitable baseline, as described next.

Among all sequences of functions that correspond to a given sequence of signs, the baseline model yields the one that maximizes the product of the (unigram) probabilities of those functions. Note that a function can correspond to one, two or more signs, so that all relevant partitions of the given sequence of signs need to be considered.

As this ignores the letters altogether, the baseline is independent of the model of Section 5, avoiding the intricacies of preconditions and postconditions.

For a concrete example, consider Figure 2b as gold standard. The ‘relevant’ items are (1) the logogram function of 𓆎 for the lemma \underline{db} , “finger”, tied to the first sign, (2) the typographical function of the three strokes, with meaning ‘plural’ and realized as letter w , tied to the next three signs, and (3) the phonogram function of 𓆏 with sound value k , tied to the last sign. Recall and precision are 100% if ‘retrieved’ are exactly these three items.

We implemented the N -gram models and HMMs from Section 6. An acyclic finite automaton is first created, with states representing configurations together with the last $N - 1$ functions or classes. Transitions are labelled by functions, and have weights that are negative log probabilities determined by the chosen probabilistic model. Most of the functions directly come from the sign list. Other functions are dynamically constructed, on the basis of the input signs, as for example typographical functions representing numbers. Another example is the class of multiplicative functions, which are generated if a pattern of one or more signs occurs two or more times. Final states correspond to configurations reached after processing all the signs of a word, with α equal to the transliteration of that word, $\beta = \varepsilon$, $\text{fl}_{\text{jump}} = \text{false}$ and $\text{fl}_{\text{fp}} = \text{false}$. A final state always exists, in the worst case by analyzing all signs as spurious, and applying one epsilon-phonogram for every letter.

The shortest path from the initial state to a final state is extracted using the shortest-path algorithm of Dijkstra (1959). The labels on this path then give us the list of functions on the basis of which we compute recall and precision.

Results are given in Table 2. It is unsurprising that the models with $N = 1$ improve over the baseline. Although the baseline is also defined in terms of unigram probabilities, it ignores consistency of the sequence of functions relative to the letters. The first-order HMM performs better than the unigram model. This can be attributed to smoothing. For example, the unigram model will assign the same low probability to a spurious function unseen in the training material as to an unseen phonogram, although phonograms overall are far more

Table 2:
Experimental results:
recall, precision, F-measure

	R	P	F1
baseline	86.0	86.0	86.0
<i>N</i> -gram			
<i>N</i> = 1	90.6	90.6	90.6
<i>N</i> = 2	94.4	94.4	94.4
<i>N</i> = 3	94.4	94.4	94.4
HMM			
<i>N</i> = 1	91.4	91.4	91.4
<i>N</i> = 2	91.8	91.8	91.8
<i>N</i> = 3	92.0	92.0	92.0
interpolation of <i>N</i> -gram and HMM			
<i>N</i> = 1	90.5	90.5	90.5
<i>N</i> = 2	94.8	94.8	94.8
<i>N</i> = 3	95.0	94.9	94.9

likely. The first-order HMM however suitably models the low probability of the class of spurious functions.

For *N* greater than 1, the HMMs perform less well than the *N*-gram models. This suggests that the probabilities of functions depend more on the exact identities of the preceding functions than on their classes. The best results are obtained with linear interpolation of the *N*-gram model and the HMM, weighted 9:1, for *N* = 3.

8 CONCLUSION AND OUTLOOK

Our contributions include the design of an annotated corpus of sign use, allowing quantitative study of the writing system, and serving to document rare uses of signs. The second main contribution is a probabilistic model of how signs follow one another to form words. The model is amenable to supervised training. Unsupervised training will be the subject of future investigation.

The probabilistic model is evaluated through computation of the most probable sequence *F* of functions given the sequence *S* of signs and the sequence *L* of letters, or formally $\operatorname{argmax}_F P(F | S, L) = \operatorname{argmax}_F P(F, S, L)$, where $P(F, S, L)$ is the joint model of Section 6. The model could also be the starting point for other tasks, such as automatic transliteration. However, evaluating $\operatorname{argmax}_L P(L | S) = \operatorname{argmax}_L \sum_F P(F, S, L)$, using the same model $P(F, S, L)$ as before, is

not likely to give satisfactory results. This is because, in general, a shorter sequence F tends to have a higher probability than a longer one, and handling of, for example, phonetic complements typically requires longer sequences involving jumps. As a consequence, overly long transliterations will be produced with repeated letters.

The solution we propose is to let the automaton model compute conditional probabilities $P(F, S | L)$, in combination with a prior model $P(L)$. This model would involve a probability distribution over the lengths of stems (most nouns and verbs have stems of two or three letters) and simple forms of morphosyntactic knowledge, including the Egyptological punctuation symbols. In the ideal case it would also include a lexicon. This is yet to be implemented and evaluated.

ACKNOWLEDGMENTS

This work evolved out of intense discussions of the first author with Serge Rosmorduc and François Barthélemy, in the summer of 2012. Gratefully acknowledged are the anonymous referee reports, which provided many insightful and useful suggestions.

REFERENCES

- J.P. ALLEN (2000), *Middle Egyptian: An Introduction to the Language and Culture of Hieroglyphs*, Cambridge University Press.
- F. BARTHÉLEMY and S. ROSMORDUC (2011), Intersection of multitape transducers vs. cascade of binary transducers: the example of Egyptian Hieroglyphs transliteration, in *Proceedings of the 9th International Workshop on Finite State Methods and Natural Language Processing*, pp. 74–82, Blois, France.
- S. BILLET-COAT and D. HÉRIN-AIME (1994), A Multi-Agent Architecture for an Evolving Expert System Module, in *Database and Expert Systems Applications, 5th International Conference*, volume 856 of *Lecture Notes in Computer Science*, pp. 581–590, Springer-Verlag, Athens, Greece.
- A.M. BLACKMAN (1932), *Middle-Egyptian Stories – Part I*, Fondation Égyptologique Reine Élisabeth.
- A.M. BLACKMAN (1988), *The Story of King Kheops and the Magicians*, J.V. Books.
- P.F. BROWN, S.A. DELLA PIETRA, V.J. DELLA PIETRA, and R.L. MERCER (1993), The Mathematics of Statistical Machine Translation: Parameter Estimation, *Computational Linguistics*, 18(4):263–311.

- P.T. DANIELS and W. BRIGHT, editors (1996), *The World's Writing Systems*, Oxford University Press, New York.
- C.T. DE VARTAVAN (2010), Snt[r]/snt[r] means '[Divine/Godly] Scent', *Advances in Egyptology*, 1:5–17.
- E.W. DIJKSTRA (1959), A Note on Two Problems in Connexion with Graphs, *Numerische Mathematik*, 1:269–271.
- N. DURRANI, H. SCHMID, A. FRASER, P. KOEHN, and H. SCHÜTZE (2015), The Operation Sequence Model — Combining N-Gram-Based and Phrase-Based Statistical Machine Translation, *Computational Linguistics*, 41(2):157–186.
- W.A. GALE and G. SAMPSON (1995), Good-Turing Frequency Estimation Without Tears, *Journal of Quantitative Linguistics*, 2(3):217–237.
- L. GALESCU and J.F. ALLEN (2002), Pronunciation of proper names with a joint n-gram model for bi-directional grapheme-to-phoneme conversion, in *Proceedings of the Seventh International Conference on Spoken Language Processing (ICSLP-2002)*, pp. 109–112, Denver, CO, USA.
- A. GARDINER (1957), *Egyptian Grammar*, Griffith Institute, Ashmolean Museum, Oxford.
- R. HANNIG (1995), *Grosses Handwörterbuch Ägyptisch-Deutsch: die Sprache der Pharaonen (2800-950 v.Chr.)*, Verlag Philipp von Zabern, Mainz.
- G. IFRAH (1981), *Histoire Universelle des Chiffres*, Editions Seghers, Paris.
- S.M. KATZ (1987), Estimation of probabilities from sparse data for the language model component of a speech recognizer, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(3):400–401.
- K. KNIGHT and J. GRAEHL (1998), Machine Transliteration, *Computational Linguistics*, 24(4):599–612.
- A. LOPRIENO (1995), *Ancient Egyptian: a linguistic introduction*, Cambridge University Press.
- M.G.A. MALIK, C. BOITET, and P. BHATTACHARYYA (2008), Hindi Urdu machine transliteration using finite-state transducers, in *The 22nd International Conference on Computational Linguistics*, volume 1, pp. 537–544, Manchester, UK.
- M.-J. NEDERHOF (2008), Automatic Alignment of Hieroglyphs and Transliteration, in N. STRUDWICK, editor, *Information Technology and Egyptology in 2008, Proceedings of the meeting of the Computer Working Group of the International Association of Egyptologists*, pp. 71–92, Gorgias Press.
- S. POLIS, A.-C. HONNAY, and J. WINAND (2013), Building an Annotated Corpus of Late Egyptian, in S. POLIS and J. WINAND, editors, *Texts, Languages & Information Technology in Egyptology*, pp. 25–44, Presses Universitaires de Liège.
- S. POLIS and S. ROSMORDUC (2013), Réviser le codage de l'égyptien ancien. Vers un répertoire partagé des signes hiéroglyphiques, *Document Numérique*, 16(3):45–67.

A probabilistic model of Ancient Egyptian writing

- S. POLIS and S. ROSMORDUC (2015), The Hieroglyphic Sign Functions. Suggestions for a Revised Taxonomy, in H. AMSTUTZ, A. DORN, M. MÜLLER, M. RONSDORF, and S. ULJAS, editors, *Fuzzy Boundaries: Festschrift für Antonio Loprieno*, volume 1, pp. 149–174, Widmaier Verlag, Hamburg.
- L.R. RABINER (1989), A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceedings of the IEEE*, 77(2):257–286.
- S. ROSMORDUC (2002/3), Codage informatique des langues anciennes, *Document Numérique*, 6:211–224.
- S. ROSMORDUC (2008), Automated Transliteration of Egyptian Hieroglyphs, in N. STRUDWICK, editor, *Information Technology and Egyptology in 2008, Proceedings of the meeting of the Computer Working Group of the International Association of Egyptologists*, pp. 167–183, Gorgias Press.
- W. SCHENKEL (1971), Zur Struktur der Hieroglyphenschrift, *Mitteilungen des deutschen archäologischen Instituts, Abteilung Kairo*, 27:85–98.
- W. SCHENKEL (1984), *Aus der Arbeit an einer Konkordanz zu den altägyptischen Sargtexten*, volume 12 of *Göttinger Orientforschungen, IV. Reihe*, Harrassowitz.
- R. SPROAT (2000), *A Computational Theory of Writing Systems*, Cambridge University Press, Cambridge.
- A. TSUKAMOTO (1997), Automated Transcription of Egyptian Hieroglyphic Texts: via transliteration using computer, *Journal of the Faculty of Culture and Education*, 2(1):1–40, Saga-University.
- E. VIDAL, F. THOLLARD, C. DE LA HIGUERA, F. CASACUBERTA, and R.C. CARRASCO (2005), Probabilistic Finite-State Machines — Part II, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1026–1039.

This work is licensed under the Creative Commons Attribution 3.0 Unported License.

<http://creativecommons.org/licenses/by/3.0/>

