An NLP Pipeline for Coptic

Amir Zeldes
Department of Linguistics
Georgetown University
amir.zeldes@georgetown.edu

Caroline T. Schroeder
Department of Religious Studies
University of the Pacific
cschroeder@pacific.edu

Abstract

The Coptic language of Hellenistic era Egypt in the first millennium C.E. is a treasure trove of information for History, Religious Studies, Classics, Linguistics and many other Humanities disciplines. Despite the existence of large amounts of text in the language, comparatively few digital resources have been available, and almost no tools for Natural Language Processing. This paper presents an end-to-end, freely available open source tool chain starting with Unicode plain text or XML transcriptions of Coptic manuscript data, which adds fully automatic word and morpheme segmentation, normalization, language of origin recognition, part of speech tagging, lemmatization, and dependency parsing at the click of a button. We evaluate each component of the pipeline, which is accessible as a Web interface and machine readable API online.

1 Introduction

Coptic emerged as a written language during the Roman era of Egypt's history, a period of significant transformation in literacy, religion, and culture (Cribiore 2001, Bagnall 2009, Frankfurter 1998). As the last phase of the Egyptian language family, it evolved from Demotic (which was widely attested in the Greek period) and ultimately the language of the ancient hieroglyphs. Although no longer in use as a living, spoken language, Coptic remains a liturgical language for the Coptic Orthodox Church. Additionally, American Copts have attempted to revive knowledge of Coptic as a mechanism for preserving cultural heritage in Egypt and the diaspora. Text corpora in this language thus hold significance for the identity formation for a current religious minority in the Middle East and U.S. as well as for research into a variety of Humanities fields, including History, Religious Studies, Classics and Linguistics, among many others.

The text corpora analyzed in this study illustrate the importance of access to original Coptic data. They originate from the formative or "classical" period of written Coptic, the fourth-fifth centuries, in the Sahidic dialect. New genres of writing emerge in this period: hagiography (saints' lives), monastic rules, Christian sermons and homilies. Coptic authors also transform and translate existing literary forms: formal epistles, gnomic sayings, and treatises. Finally, documentary sources (wills, receipts, contracts, transactional letters) as well as school exercises, prayers, magical texts, and literary fragments survive on scraps of papyri, potsherds (known as ostraca), or inscriptions and graffiti on monuments.

Our earliest witness to biblical passages in Coptic also survive as fragmentary documents or as quotations of scriptural passages within the classical Coptic texts. A fundamental, outstanding question for both biblical studies and the history of Christianity is whether our earliest known Coptic authors quoted from existing written translations of biblical books, or whether they translated scripture "on the fly" as they wrote and spoke. Coptologists have observed the influence of the Bible on Coptic composition patterns, describing some authors as writing in a biblical style (Goehring 1999:226, Schroeder 2006).

Coptic texts provide an important resource for the study of gender and language in premodern societies, as well. During a time when few texts about women were composed, and even fewer documents were written by women, Coptic letters by and about women have nonetheless sur-

Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities (LaTeCH), pages 146–155,
Berlin, Germany, August 11, 2016.

vived, shedding light on otherwise obscure facts (Bagnall and Cribiore 2006, Wilfong 2002).

Investigations of questions like the above benefit directly from a digitized corpus with linguistic, lexical and syntactic annotations, which are quite complex. Moreover, the structure of the language and the dearth of existing digital resources for Coptic mean that the creation of NLP tools for this 'low-resource language' is more challenging than for other Classical languages, such as Greek and Latin. As we will show below, in order to study texts in the Coptic language, substantial pre-processing must be accomplished: Coptic word forms can contain multiple lexical items of interest, manuscript spelling must be normalized to allow searchability, foreign words (mostly Greek) need to be recognized, and tagging, lemmatization and parsing can allow much more detailed searches for both Linguistics and other Humanities research questions (grammatical patterns, identifying proper names, and more). The need to make these resources available to a broader audience outside of Computational Linguistics motivates the creation of an easy to use interface, which starts with transcribed text and proceeds automatically through the needed levels of analysis. The ideal architecture for such an interface is an NLP pipeline with modular components and an online API (cf. WebLicht, Hinrichs et al. 2010). This paper therefore presents and evaluates the necessary components for a new online API for Coptic NLP.

2 NLP Components

The NLP pipeline presented below offers an end-to-end solution for processing Coptic text from UTF-8 plain text or XML to segmented, machine readable data. In the following sections we describe and evaluate the different NLP tools applied to input data, including bound-group segmentation, normalization, morphological analysis, POS tagging, lemmatization, language of origin detection for loanwords, and syntactic dependency parsing.

2.1 Segmentation

Like many other languages of the Near East, Coptic 'words' in the sense of space delimited units contain multiple subunits that need to be made actionable. Similarly to Arabic or Hebrew, prepositions, conjunctions and enclitic pronouns are spelled together with lexical units in what is known as 'bound groups' (Layton 2011: 12-20), as illustrated in (1).[1] Unlike Hebrew and Arabic, bound groups also contain verbal auxiliaries, such as the past tense base <a> in (1), and subject pronouns, such as <f> 'he'. We separate bound group elements with a '-', and smaller morphemes (e.g. affixes) with a '.'.

(1) <a-f-bōk mn̩-p-rōme>
PAST-he-go with-the-man
'he went with the man'

The situation in Coptic is further complicated, compared to some Semitic languages, since compounds are also spelled together (unlike Semitic construct states), and derivational prefixes may be added to lexical stems as well, as shown in (2) and (3). These must be handled, among other reasons, because we want to carry out language of origin detection later on: it is possible for only part of such a complex word to be a Greek stem, as in (3).

(2) <pe-ʃbr̩.r̩.hōb>
the-friend.do.act
'the accomplice' (lit. 'act-do-friend')

(3) <t-mn̩t.ref.hetḇ.psyxē>
the-ness.er.kill.soul
'the soul-killing'

In (3), only the incorporated object of the nominalized verb 'to soul kill' is of Greek origin (cf. 'psyche'). The agentive and abstract affixes corresponding to English *-er* and *-ness* demonstrate the incorporation (lit. 'soul-kill-er-ness'). For this paper, we will refer to the space delimited units such as <a-f-bōk> 'he went' and <mn-p-rōme> 'with the man' as 'bound groups' – these appear without spaces or hyphens in Coptic. Their constituents, such as <p> 'the' or <rōme> 'man' will be referred to as 'word units', while smaller parts (affixes, compound constituents) will be called morphs.

The first level of segmentation is separation into bound groups. Although early Coptic manuscripts were written without spaces entirely, scholars making use of our pipeline generally introduce spaces between bound groups as they transcribe. We therefore do not attempt to solve

[1] Throughout this paper, we will use angle brackets to denote Coptic graphemes (<b> the letter 'b' or Beta in Coptic), slashes for phonemes (the phoneme /b/), and square brackets for reconstructed pronunciations leading to spelling variation (e.g. /b/ may have been pronounced [p] and occasionally spelled as non-standard <p> by some). Syllabic consonants are marked with a vertical line below, and long vowels carry a macron.

the problem of segmenting continuous text into bound groups beyond the trivial whitespace and punctuation-based splitting.

The second level of splitting bound groups internally is the main challenge. In order to recognize the constituents of a bound group, we rely on an initial normalization, which amounts to stripping diacritics and expanding some contractions (see next section). These are harvested from our manually annotated training corpus of just over 50,000 word units. Of these, about 28,000 tokens come from Biblical texts translated from Greek, while the remainder comes from native literary Coptic texts, including sermons and letters by two abbots of the White Monastery, St. Shenoute of Atripe and Besa, as well as narrative texts from the Sayings of the Desert Fathers.[2]

Sequences known from our training corpus are immediately analyzed via majority vote, favoring the most frequent analysis in the training data.[3] For novel sequences, we rely on the assumption that each bound group contains only one open-class word unit (e.g. a noun or verb), notwithstanding compounds. Since compounds are considered single word units with multiple morphs, we can still rely on there being only one such word unit in the bound group.

We proceed to subject the bound group to a cascade of some 180 prioritized segmentation rules describing possible Coptic bound groups, which can be filled with open class items from our lexicon. The lexicon was constructed using items from the training corpus, over 4,000 items from the CMCL project (Orlandi 2004) and a further 1,700 Greek loan words from the Database and Dictionary of Greek Loanwords in Coptic[4], for a total of over 7,500 items.

Since Coptic bound group formation is non-recursive (no recursive compounding), we generate the finite set of possible derived forms using the lexicon, which accounts for compound nouns and denominal verbs. Open class items, whether listed in the lexicon or dynamically generated by this procedure, are subjected to morphological analysis. This allows us to output the final segmented form with all three levels: bound group, word units and morphs.

As an example, consider the following bound group, which is decorated with several over-dots in a manuscript:

(4) <j̈i-nt-a-i̇-ėr.monaxos>
since-REL-PAST-I-do.monk
'since I became a monk'.

The original Coptic form has a spelling variant <er> for normalized <r> 'do' and dots, partly decorative and partly indicating syllabicity on the <r>. After the dots are stripped, we look for a segmentation based on rule priorities. Since this is a rather long, complicated sequence, it is not matched until rule #156, which matches the structure:

conjunction+relative+aux+subject+verb

Since the subject is pronominal the only open class element in this constellation is the verb, which is however a complex, denominal verb, derived from <monaxos> 'monk': <r.monaxos> 'being a monk' can roughly be rendered as 'do-monk' or 'monk-ify'. While <er> is non-standard orthography, the common variant <er> for <r> is listed in our lexicon. The unlisted normalized verb form <r.monaxos> can be generated from the lists of verbs and nouns, allowing the analysis to go through, as well as the subsequent morphological analysis, which attempts to find the longest possible constituent first, and only matches the option of <r>+<monaxos>: 'do'+'monk'.

Table 1 gives the current accuracy of our results using 10-fold cross-validation: some 14,000 bound groups, from the dataset described above, are shuffled and sliced into 10 equal blocks, each of which is used as test data again the remaining 90% training data. The baseline represents accuracy when no segmentation is carried out – nearly 40% of bound groups require no segmentation. Rules and training data used together achieve just over 90% accuracy, with less than 1% standard deviation.

(n=14,410)	Ø % correct	sd
baseline	39.85	1.21
training	69.42	0.99
rules	87.28	1.01
rules+training	**90.21**	0.70

Table 1: Segmentation accuracy in 10-fold cross validation.

[2] For a complete list of corpora used in this paper with version information and stable URNs, see the corpus references at the end.

[3] Unlike in the Semitic languages, multiple valid segmentations of the same string are very rare, largely owing to the fact that Coptic spelling includes vowels – see more below on the comparison with Semitic languages.

[4] http://research.uni-leipzig.de/ddglc/

These results are somewhat behind the state of the art in similar tasks for languages with larger data resources, such as Hebrew (92.32%, Adler & Elhadad 2006), and Arabic (between 97.61 and 98.23 on Standard Arabic news text, or 92.1% on Egyptian Arabic, Monroe et al. 2014). However, it must be kept in mind that the amount of training data available for those languages is orders of magnitude larger than the 14K bound groups used here, and that the nature of our texts is less standardized or redacted than modern newswire data. On the other hand, the relatively good results are probably due to availability of vowel information in Coptic, which is missing in most Hebrew and Arabic data.[5]

2.2 Normalization

For historical texts, normalization is an essential component for ensuring machine-actionability of data (see Piotrowski 2012: 69-84). In Coptic, at least three kinds of normalization issues must be resolved for subsequent processing: 1. diacritics, 2. spelling variation and 3. abbreviations.

Coptic diacritics are used to express non-linguistic decorations, abbreviations, or reading pause signs in manuscripts (5), linguistic properties such as diphthongs marked with diaresis or syllabic consonants marked with superlinear strokes or dots (6), as well as paleographic information introduced by transcribers to indicate damage to the manuscript (7).

(5) ⲛⲛⲉ̀ⲯⲩⲭⲏ̀ <n-ne[n]-psyxē> ‘of our soul’ (with pausal apostrophe sign at the end and raised tilde for an abbreviated /n/)
(6) ⲛⲁ̈ⲓ ⲙ̄ⲛ <nai mṇ> ‘these and…’
(7) ⳟⲱⲡⲉ̣ <ʃōpe> ‘become’ (with underdot indicating damage to the /e/)

Although the variations in (6), which is shown in the original in Figure 1, are linguistically meaningful (consonant syllabicity can occasionally distinguish homographs), their presence is not reliable in many manuscripts, so that complete removal of diacritics is the safer strategy for input to subsequent stages in the pipeline.

Other spelling variations primarily affect vowels for which post-classical Greek pronunciation allows for confusion of similar sounds. Unlike the situation for older stages of English or other European languages (Reynaert et al. 2012, Archer et al. 2015), spelling is relatively stable in Coptic, partly due to the phonetic nature of the script system. Most frequently we see variation between ⲉⲓ and ⲓ for the vowel /i/ (8), and various Greek letters representing /i/, such as ⲏ or ⲩ (9) (similar issues occasionally affect /u/).


Figure 1: Diacritics in manuscript for (6). Image: Österreichische Nationalbibliothek, http://data.onb.ac.at/rec/RZ00002466

(8) ⲉⲣⲟⲉⲓ <eroei> ‘to me’; var. of ⲉⲣⲟⲓ <eroi>
(9) ⲥⲭⲩⲙⲁ <sxyma> ‘habit’, error for ⲥⲭⲏⲙⲁ <sxēma>, both pronounced [sk^{h}i:ma]

In non-Greek words, most texts adhere to a convention where semivowels /j/ and /w/ are spelled by a simple ‘i’ or ‘u’ after another vowel, and otherwise with a preceding ‘e’ or ‘o’ (Layton 2011: 17-18). For Greek words and violations of these conventions in non-Greek words, the only recourse is to look up the word with the expected spelling of i/u in a lexicon and retrieve the normalized counterpart.

Finally, for abbreviations, such as sacred names (10), a list of common cases is maintained, which is consulted during normalization. Additionally, for some common abbreviations, such as an isolated stroke representing line-final /n/, the lexicon can be consulted.

(10) ⲓⲥ <is> ‘Jesus’ (for ⲓⲏⲥⲟⲩⲥ <iēsous>)

To evaluate our normalization component, we use only literary Coptic manuscript data, since the Bible data is partly edited (less than 2% of training data required normalization for the Bible dataset). Table 2 gives the results for 10-fold cross-validation.

normalization	% correct (sd)	tokens
baseline (ident)	61.12 (0)	21,400
training	89.76 (3.86)	21,400
deterministic	97.24 (1.19)	21,400
both	**98.01 (1.11)**	21,400

Table 2: Normalization accuracy.

As the table shows, the baseline of assuming the actual manuscript form is already correct is fairly high, at 61%, since very many of the most

[5] At the same time, vowels introduce a possible locus for false segmentations, meaning their availability, and the resulting longer words, are not always an advantage.

frequent function words show virtually no variation (e.g. past auxiliary <a>, words like <auō> 'and'). Consulting 90% of the data to predict the correct form in each 10% of test data is also fairly successful, at 89% accuracy, since most common abbreviations will already be attested elsewhere in the corpus. However, consulting the deterministic list of most frequent variants and spelling adjustments (about 20 rules), as well as automatic handling of diacritics and capitalization variation already gives us almost optimal performance at 97%, while combining both strategies reaches 98%. It therefore appears that normalization of literary manuscripts on (gold segmented) data works well, with only about 2 words in 100 showing an unpredictable, aberrant spelling.

It should be noted, however, that our corpus focuses on prestigious, carefully copied works: a toy evaluation on 3 documentary papyri (personal contracts and letters) with only 281 word units taken from papyri.info (see Sosin 2010) resulted in 85.97% accuracy, improving on a baseline of 63.28% for this much harder dataset.

2.3 Tagging and lemmatization

Part of speech tagging and lemmatization are crucial, both in order to investigate grammatical patterns and to find different senses of the same word (e.g. as a noun or a verb, often having the same form in Coptic) or to generalize across inflected forms of the same word for non-linguistic research. Additionally, if special tags are given to items such as proper nouns, we can use a tagger to find mentions of people and places in texts, which ultimately contributes to named entity recognition (an NER component is planned for future work, see Section 5).

Previous work on tagging low resource languages has focused on annotation projection (see Yarowsky et al. 2001) from similar languages with larger training data that is available in translation in the target language. Most often, this has been the Bible, which is also available in Coptic. However, Coptic is structurally rather different from the typical 'large coverage' languages, and annotation projection approaches have typically produced results for comparatively 'general', not very language specific tag sets, with accuracies in the 70-90% range (Agić et al. 2015, Kim et al. 2015).[6] Additionally, since many native texts beyond the Bible are available for Coptic, we decided to annotate and train a tagger on a larger variety of texts.[7]

For part of speech tagging, we use a set of 46 tags, most of which correspond to closed classes of auxiliary conjugation bases (15), pronouns (6), or complementizers (also known as 'converters' in Coptic grammar, 4). The main lexically open categories are verbs (4 classes) and nouns (common and proper), as well as some adverbs (Coptic has no open class of adjectives). The tagger's two main challenges are therefore guessing the tag for open class items that are either unfamiliar, or can be both a noun and a verb, and disambiguating closed class items. The latter can be highly ambiguous: for example, the most common functional elements in the language, <e> and <n>, can each carry 8 different tags (e.g. <e> is the prepsotion 'to', an adverbial complementizer, a form of 2nd person feminine pronoun, etc.).

In order to speed up manual tagging, and also for higher performance on noisy data, we also tested a more coarse grained tag set, collapsing several categories for a total of 24 tags. The main differences in the smaller tag set are not distinguishing each of the auxiliaries (which usually have distinct forms) and complementizers (which often do not), and collapsing all verbs to one tag (V), as well as common and proper nouns (N).

For tagging we use the TreeTagger (Schmid 1994), a fast, robust and trainable, language independent tagger based on decision trees. TreeTagger also has the advantage of carrying out lemmatization concurrently with lemma selection based on the induced tag sequence. Table 3 gives results for different subsets of the data described in Section 2.1, using 10-fold-cross validation (this time using randomly shuffled sentences instead of individual words, to maintain n-gram integrity).

tagging	% fine (sd)	% coarse (sd)	tokens
baseline (N)	14.21 (0)	15.32 (0)	50,300
all data	94.48 (1.95	95.12 (1.43)	50,300
no fragments	**94.99 (0.50)**	**95.65 (0.40)**	49,400
Bible only	95.89 (0.99)	96.14 (0.87)	28,600
documentary	87.54 (0)	92.52 (0)	281

Table 3: Tagging accuracy.

[6] The higher end of the spectrum contains some European languages, such as Lithuanian, while Afroasiatic languages such as Hebrew are at more modest, near 70% performance using only annotation projection.

[7] This contrasts with Agić et al.'s titular situation 'when all you have is a bit of the Bible'.

The baseline figure is obtained by assigning the most frequent tag, N (common noun) to all items. Despite the relatively modest amount of data, performance on the entire data set is over 94%, which is above annotation projection results in previous work on other languages. Removing fragmentary sentences (under 1000 tokens) from the corpus, which contain lacunae in the original manuscripts, increases accuracy by 0.5%, though realistically such sentences are expected to occur in the Coptic data. Reducing the dataset to include only Biblical material, which is linguistically simpler than untranslated, native Coptic literature, sees a gain of almost 1%.

Switching to the coarse tag set offers a surprisingly modest gain, especially in the cleaner text of the Bible. However, we also ran a tentative test on the 281 words of non-literary papyri mentioned above: when tagging based on training data from the literary material, the coarse tag set is nearly 5% more accurate.

Lemmatization, which was also carried out via TreeTagger, is a considerably easier task for Coptic, since most words are uninflected (only about 5% of nouns and 17% of verbs in our data differ in form from their lemma). As a result, the baseline of assuming that a word has its own form as the lemma is fairly high (63%). Additionally, our lexical resources from CMCL and the Greek lemma list from DDGLC provide excellent coverage for literary Coptic, resulting in the tagger primarily having to disambiguate the correct tag to find the right lemma (under 97% accuracy). If we then assume that unknown forms have themselves as a lemma, we arrive at over 97% accuracy. Table 4 summarizes our results based on the subset of data which has been lemmatized so far, using 10-fold cross-validation.[8]

lemmas	% correct (sd)	tokens
baseline (=word)	63.01	37,800
stochastic lookup	96.78 (1.14)	37,800
no unknown	**97.23 (1.13)**	37,800

Table 4: Lemmatization accuracy.

2.4 Language of origin detection

Recognizing words of Greek and other origins is of great interest to a variety of humanities disciplines (Torallas-Tovar 2010), including religious studies, cultural history and contact linguistics. The influence of the Greek lexicon on the Coptic stage of the Egyptian language was substantial (Grossman 2013); in our data set we find about 8% word units of Greek origin in Bible data, and about 6% in native literary Coptic.

However, not all 'Greek' words in Coptic are of ultimately Greek origin: many words that are of Biblical Hebrew origin, as well as Latin words (especially official and legal terms) are well attested in Coptic. Although arguably all such words were loaned into Coptic from Greek, it is often difficult to tell – is the word <komes> 'governer, count' the Latin word *comes* or its Greek counterpart, *komes*? We therefore follow the guideline of assigning each word its earliest identifiable donor language, with the understanding that a total count of 'Greek' words may be extracted by considering all loans of this type.

Our language of origin recognizer component is fed the same normalized word units given to the tagger, which are outputted by the tokenizer and normalizer chain. They are matched against a list of items taken from DDGLC and our manually tagged data, amounting to a lexicon of over 2,700 loanword types. Additionally, we match some highly probable patterns, such as words ending in the typically Greek endings <os> or <ēs>, if they are not known to the recognizer (currently we have 8 such affix rules).

To evaluate language of origin tagging we used double-checked 7,200 word units from the Sayings of the Desert Fathers, which were translated from Greek, and three open letters by Archmandrite Shenoute and his successor Besa, abbots of the White Monastery in upper Egypt, which were originally composed in Coptic. The total accuracy for this subset (including correct negatives for all Coptic words) was 99.47%. However the entire dataset contained only 476 loanwords, meaning that a 'negative' baseline (guessing all words are native) gives 93.39% accuracy. Nevertheless, precision and recall within the data flagged by either annotators or the language recognizer was high, with 99.54% precision (almost no false positives) and 92.43% recall, for an F1 score of 95.85. Our results show that the DDGLC lemma list is very comprehensive for our data. Recall failures were largely due to (often Biblical Hebrew) proper names or their variant spellings which were not on the list.

2.5 Parsing

Syntactic parsing is an essential component in enabling information extraction (e.g. finding out

[8] This is the same data set evaluated above, but excluding two of Shenoute's sermons and some of the Bible data which have not been checked yet.

all predicates associated with the subject lemma 'angel' in a text), subsequent entity recognition (providing nominal phrase spans, identifying appositions) and of course the study of syntax itself. Recent approaches to parsing for low-resource languages have harnessed fully unsupervised, and semi-supervised methods, learning parsing models via simulations based on smaller datasets (Sun et al. 2014) or by analogy to larger data in similar languages (Duong et al. 2015). These approaches excel at requiring little to no manual annotation, but deliver parsing accuracy below 80%. As with tagging, we therefore opted to develop training data manually, which we complement with rule-based post-processing.

Because the construction of manually annotated treebank data is difficult and time consuming, especially for full constituent parses resembling the Penn Treebank scheme (Bies et al. 1995), we have chosen to focus on dependency parsing with a relatively simple scheme, following the Universal Dependencies project (de Marneffe et al. 2014), as used also in Duong et al.'s work. Universal Dependencies (UD) are a 'lexico-centric' formalism focused on marking relations between lexical heads, such as verbs and their arguments, while assigning functional elements such as prepositions and auxiliaries a dependent status. For example, prepositions are seen as 'case markers', dependent on nouns. Figure 2 illustrates a UD tree for Coptic.

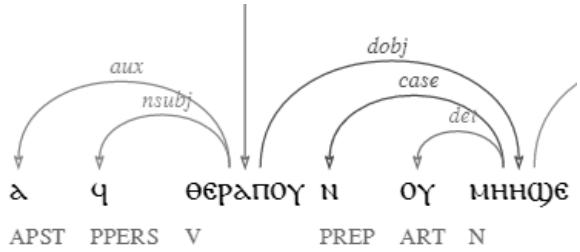

Figure 2: Coptic Universal Dependency tree from Mark 1:34: *He healed a multitude*.

Our inventory of labels follows the latest tag set at `http://universaldependencies.org/`, which includes as many as 40 labels (some rare labels, such as *reparandum* and *remnant* are not yet attested in our annotated data). Common labels include subject and object tags for nominals and clauses (*nsubj*, *dobj*, *csubj*, *ccomp*), case markers as seen in Figure 2, and nominal modifiers (*nmod*), among others (see de Marneffe et al. 2014 for a full discussion).

Our training data set is still very small, currently only 4,361 word units, coming from the sermons, Biblical material, and the Sayings of the Desert Fathers. The data is annotated with the fine-grained tags[9] described in section 2.3, as well as the university dependency labels and automatically generated universal POS tags as defined by the UD project. The data set is freely available for download under a CC-BY license from the UD website.

As a result of the small amount of data, only a rudimentary parsing model could be trained for the pipeline. As a baseline for parser performance we take the most frequent label for all items and assume each token attaches to its predecessor. We then test two approaches to parsing the data: using a rule based parser called DepEdit, which can apply attachment and labeling rules based on POS tag sequences, and MaltParser (Nivre 2009), a freely available trainable dependency parser implementing a variety of algorithms. Since DepEdit is not trainable, we evaluate it against the entire dataset; for MaltParser we use 10-fold cross-validation with random sentence ordering.

	attach (sd)	**label (sd)**	**both (sd)**
baseline	34.41 (0)	11.78 (0)	0.29 (0)
depedit	80.04 (0)	84.72 (0)	79.29 (0)
malt	85.72 (2.1)	85.83 (2.0)	80.09 (2.1)
malt+depedit	**85.85 (2.3)**	86.74 (2.1)	80.08 (2.1)
malt+morph+depedit	85.36 (2.4)	**87.51 (2.3)**	**81.06 (2.7)**

Table 5: Parser performance on 4,361 word units.

The rule-based DepEdit parser uses some 80 attachment and labeling heuristics, which achieve 80% attachment accuracy, almost always with correct labels (accuracy on both = 79%). These rules correspond more or less to the possible bound group configurations (e.g. connecting a verb to its subject and auxiliary with correct labels), plus some heuristics for clause juncture (attaching relative and adverbial clauses).

[9] An anonymous reviewer has suggested trying to train the parser on the coarse tag set and then using features from the parse to disambiguate coarse tags into fine ones. Although we were unable to test this idea before the deadline, it is an interesting prospect to go back from parses to the tagger or attempt joint inferences (cf. Bohnet & Nivre 2012). It should however be noted that Coptic subject and object pronouns are only distinguished in the fine-grained subset, which is therefore likely to be helpful for the parser.

The first Malt model[10] in the table beats DepEdit's attachment, by over 5%, with similar labeling accuracy. However, since DepEdit can apply rules to already parsed data, we tested a combined approach, in which Malt output is passed through a set of the most reliable DepEdit heuristics (60 rules) to correct very certain cases for which the small training data does not ensure correct parses. This approach maximizes attachment accuracy (85.85%). Finally, we tested automatic addition of morphological features for definiteness, gender, finiteness and subordination using adjacent articles (for nouns) or subordinators and infinitive markers (for verbs). Giving these to MaltParser produced the last model, with best labeling (87.51%) and labeled attachment accuracy (81.06%), at the cost of a small drop in attachment-only accuracy (85.36%).

3 Pipeline architecture and merging

The components outlined above are freely available as standalone command line tools, and as a pipeline wrapped inside a Python controller script. The pipeline can be accessed using a web interface, or also addressed programmatically, using a RESTful API (cf. Fielding 2000).

Communication between components uses the vertical SGML markup format used by the TreeTagger and codified by the IMS Corpus Workbench (CWB or CQP vertical format, see Hardie 2012: 390). In this format, minimal tokens of the running text are presented in a one token per line format, while XML opening and closing tags, each occupying their own line, designate span annotations encompassing multiple tokens. Spans of bound groups, morphemes, normalization, tagging and lemmatization are all expressed in this format, illustrated below.

```
<norm_group norm_group="ⲧⲙⲛⲧⲙⲟⲛⲁⲭⲟⲥ">
<norm xml:id="u5" pos="ART" lemma="ⲡ" norm="ⲧ" func="det"
head="#u6" >
ⲧ
</norm>
<norm xml:id="u6" pos="N" lemma="ⲙⲛⲧⲙⲟⲛⲁⲭⲟⲥ"
norm="ⲙⲛⲧⲙⲟⲛⲁⲭⲟⲥ" func="dobj" head="#u3">
<morph morph="ⲙⲛⲧ">
ⲙ̄ⲛ̄ⲧ
</morph>
<morph morph="ⲙⲟⲛⲁⲭⲟⲥ" xml:lang="grc">
ⲙⲟⲛⲁⲭⲟⲥ
</morph>
</norm>
</norm_group>
```

In this example, which analyzes the bound group *t-mnt.monaxos* 'the monkhood', the entire group is encompassed by a `<norm_group>` tag and normalized by removing diacritics from 'mnt'. The feminine article 't' is recognized, split off by the tokenizer, tagged 'ART' and lemmatized by the tagger. The subsequent complex noun is also morphologically analyzed and assigned a Greek language of origin in the second morpheme. Finally the first 'norm' unit is assigned the syntactic function 'det(erminer)' and its syntactic head is set to the noun's xml:id. These pieces of information are added sequentially, as each component reads input from the tags it expects (usually the 'norm' tag) and injects its analysis as a further tag or attribute where appropriate (morphological analysis injects `<morph>` tags, tagging injects `pos` attributes in `<norm>` tags, etc.).

The format used above is also tolerant of hierarchy conflicts (hence SGML and not XML), which may arise if other span annotations exist in the input data, if it has been marked up for other properties, such as document structure using TEI XML (Burnard & Bauman 2008). Since pipeline components only look for and interact with specific tag names, any other markup in the data is simply preserved. Most frequently, such markup includes pages, columns and line break information from the manuscripts.

Individual components may be switched off, so that partial processing is possible. In practice, users may want to stop the pipeline early, e.g. after tokenization, in order to correct partial output and obtain better results on subsequent tasks. Correcting tokenization will prevent inevitable tagging errors, both on mistokenized words and their immediate neighbors. Subsequently, users can continue processing using the corrected data. Our ultimate goal is to integrate the NLP tools into an editing environment for transcribing Coptic manuscripts, so that annotators can consult the tools and get improved analyses of their data.

4 Access

All of the tools and data created within this project are open source and freely available: corpus data under Creative Commons licenses and tools under the Apache 2.0 license. An online interface and a REST API for the pipeline are available at: `https://corpling.uis.georgetown.edu/coptic-nlp/`.

Source code for both the pipeline wrapper controller script and the individual command line tools can be freely downloaded from

[10] We used the stackeager parsing algorithm and liblinear classifier throughout, as these achieved the best results.

http://github.com/CopticScriptorium. For more information on the NLP tools and for access to the corpus data sets, see http://copticscriptorium.org/ and the URLs in the references.

5 Conclusion and outlook

The NLP pipeline presented here is a first solution for largely automatic handling of Coptic text for Humanities research. By offering a pipeline that begins with raw, unsegmented, non-normalized text and automatically applying segmentation, normalization, tagging, lemmatization, language of origin detection and parsing, users only need to provide a transcription of the text they are working on, and receive a good approximation of a linguistic analysis of their data.

Beyond improving the existing components, and especially the tokenizer and parser, which leave substantial room for improvement, we plan to extend the pipeline to named entity recognition next, by developing lexical resources for contemporary entities (lists of people and places in 1st millennium Egypt) and harnessing nominal phrase boundary detection using the POS tagger and parser. This will enable us to approach quantitative questions spanning multiple annotation layers, such as who is mentioned where and how often, who does what to whom, what are typical sequences of events involving certain types of participants, where these differ, and more.

Acknowledgments

The research and corpus development for this paper was supported by grants from the National Endowment for the Humanities (HD-51907, PW-51672-14, HG-229371) and the German Federal Ministry of Education and Research (BMBF, 01UG1406). We thank the editors and annotators of the digital corpora used in this study: Rebecca Krawiec, Paul Lufter, Christine Luckritz Marquis, So Miyagawa, Tobias Paul, Elizabeth Platte, Janet Timbie, David Sriboonreuang, Lauren McDermott, Elizabeth Davidson, Alexander Turtureanu. Lexical resources were provided by the Corpus dei Manoscritti Copti Letterari (CMCL, http://www.cmcl.it/) and the Database and Dictionary of Greek Loanwords in Coptic (DDGLC, http://research.uni-leipzig.de/ddglc/). We would also like to thank three anonymous reviewers for their valuable comments on an earlier version of this paper. The usual disclaimers apply.

References

Meni Adler and Michael Elhadad. 2006. An Unsupervised Morpheme-Based HMM for Hebrew Morphological Disambiguation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*. Sydney, 665–672.

Zeljko Agić, Dirk Hovy and Anders Søgaard. 2015. If All You have is a Bit of the Bible: Learning POS Taggers for Truly Low-resource Languages. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*. Beijing, 268–272.

Dawn Archer, Merja Kytö, Alistair Baron and Paul Rayson. 2015. Guidelines for Normalising Early Modern English Corpora: Decisions and Justifications. *ICAME Journal* 39:5–24.

Roger S. Bagnall.. 2009. *Early Christian Books in Egypt*. Princeton, NJ: Princeton University Press.

Roger S. Bagnall and Raffaella Cribiore. 2006. *Women's Letters from Ancient Egypt, 300 BC-AD 800*. Ann Arbor: University of Michigan Press.

Ann Bies, Mark Ferguson, Karen Katz and Robert MacIntyre. 1995. *Bracketing Guidelines for Treebank II Style. Penn Treebank Project*. Technical Report, University of Pennsylvania.

Bernd Bohnet and Joakim Nivre. 2012. A Transition-based System for Joint Part-of-speech Tagging and Labeled Non-projective Dependency Parsing. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Jeju Island, Korea, 1455–1465.

Lou Burnard and Syd Bauman. 2008. *TEI P5: Guidelines for Electronic Text Encoding and Interchange*. Available at: http://www.tei-c.org/Guidelines/P5/.

Raffaella Cribiore. 2001. *Gymnastics of the Mind: Greek Education in Hellenistic and Roman Egypt*. Princeton, NJ: Princeton University Press.

Long Duong, Trevor Cohn, Steven Bird and Paul Cook. 2015. A Neural Network Model for Low-Resource Universal Dependency Parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP2015)*. Lisbon, 339–348.

Roy Thomas Fielding. 2000. *Architectural Styles and the Design of Network-based Software Architectures*. PhD Thesis, University of California, Irvine.

David Frankfurter. 1998. *Religion in Roman Egypt: Assimilation and Resistance*. Princeton, NJ: Princeton University Press.

James Goehring. 1999. The Fourth Letter of Horsiesius and the Situation in the Pachomian Community Following the Death of Theodore. In James Goehring (ed.), *Ascetics, Society, and the Desert*. Harrisburg, PA: Trinity Press, 221-240.

Eitan Grossman. 2013. Greek Loanwords in Coptic. In Georgios K. Giannakis (ed.), *Encyclopedia of*

Ancient Greek Language and Linguistics. Leiden: Brill, 118–119.

Andrew Hardie. 2012. CQPweb - Combining Power, Flexibility and Usability in a Corpus Analysis Tool. *International Journal of Corpus Linguistics* 17(3):380–409.

Erhard W. Hinrichs, Marie Hinrichs and Thomas Zastrow. 2010. WebLicht: Web-Based LRT Services for German. In *Proceedings of the ACL 2010 System Demonstrations*. Uppsala, 25–29.

Young-Bum Kim, Benjamin Snyder and Ruhi Sarikaya. 2015. Part-of-speech Taggers for Low-resource Languages using CCA Features. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP2015)*. Lisbon, 1292–1302.

Bentley Layton. 2011. *A Coptic Grammar*. Third Edition, Revised and Expanded. (Porta linguarum orientalium 20.) Wiesbaden: Harrassowitz.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre and Christopher D. Manning. 2014. Universal Stanford Dependencies: A Cross-Linguistic Typology. In *Proceedings of 9th International Conference on Language Resources and Evaluation (LREC 2014)*. Reykjavík, Iceland, 4585–4592.

Joakim Nivre. 2009. Non-Projective Dependency Parsing in Expected Linear Time. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Singapore, 351–359.

Tito Orlandi. 2004. Towards a Computational Grammar of Sahidic Coptic. In Mat Immerzeel and Jacques van der Vliet (eds.), *Coptic Studies on the Threshold of a New Millennium. Proceedings of the Seventh International Congress of Coptic Studies*. . Vol. 1. Leiden: Peeters, 125–130.

Michael Piotrowski. 2012. *Natural Language Processing for Historical Texts*. Morgan & Claypool. San Rafael, CA.

Martin Reynaert, Iris Hendrickx and Rita Marquilhas. 2012. Historical Spelling Normalization. A Comparison of Two Statistical Methods: TICCL and VARD2. In *Proceedings of the Second Workshop on Annotation of Corpora for Research in the Humanities*. Lisbon.

Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the Conference on New Methods in Language Processing*. Manchester, UK, 44–49.

Caroline T. Schroeder. 2006. Prophecy and Porneia in Shenoute's Letters: The Rhetoric of Sexuality in a Late Antique Egyptian Monastery. *Journal on Near Eastern Studies* 65 (2): 81-97.

Joshua Sosin. 2010. Digital Papyrology. In *26th Congress of the International Association of Papyrologists, 19 August 2010*. Geneva. Available at: http://www.stoa.org/archives/1263.

Liang Sun, Jason Mielens and Jason Baldridge. 2014. Parsing Low-resource Languages using Gibbs Sampling for PCFGs with Latent Annotations. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar, 290–300.

Sofia Torallas-Tovar. 2010. Greek in Egypt. In Egbert J. Bakker (ed.), *A Companion to the Ancient Greek language*. Oxford: Willey-Blackwell, 253–266.

Terry G. Wilfong. 2002. *Women of Jeme: Lives in a Coptic Town in Late Antique Egypt*. Ann Arbor: University of Michigan Press.

David Yarowsky, Grace Ngai and Richard Wicentowski. 2001. Inducing Multilingual Text Analysis Tools via Robust Projection across Aligned Corpora. In *Proceedings of the First International Conference on Human Language Technology Research (HLT '01)*. San Diego, CA, 1–8.

Corpora

Coptic SCRIPTORIUM. 2015a. *apophthegmata.patrum*, urn:cts:copticLit:ap, v1.5, 2015-10-04. http://data.copticscriptorium.org/urn:cts:copticLit:ap.

Coptic SCRIPTORIUM. 2015b. *sahidica.1corinthians*, urn:cts:copticLit:sahidica.1corinthians, v1.2.0, 2015-07-30. http://data.copticscriptorium.org/urn:cts:copticLit:sahidica.1corinthians.

Coptic SCRIPTORIUM. 2015c. *sahidica.mark*, urn:cts:copticLit:sahidica.mark, v1.4, 2015-09-27. http://data.copticscriptorium.org/urn:cts:copticLit:sahidica.mark.

Coptic SCRIPTORIUM. 2015d. *shenoute.abraham*, urn:cts:copticLit:shenoute.abraham, v1.3.0, 2015-09-08. http://data.copticscriptorium.org/urn:cts:copticLit:shenoute.abraham.

Coptic SCRIPTORIUM. 2015e. *shenoute.eagerness*, urn:cts:copticLit:shenoute.eagerness, v1.1, 2015-05-27. http://data.copticscriptorium.org/urn:cts:copticLit:shenoute.eagerness.

Coptic SCRIPTORIUM. 2015f. *shenoute.fox*, urn:cts:copticLit:shenoute.fox, v1.2, 2015-05-28. http://data.copticscriptorium.org/urn:cts:copticLit:shenoute.fox.

Coptic SCRIPTORIUM. 2016a. *besa.letters*, urn:cts:copticLit:besa, v1.4.1, 2016-03-28. http://data.copticscriptorium.org/urn:cts:copticLit:besa.

Coptic SCRIPTORIUM. 2016b. *papyri.info*, urn:cts:copticDoc:papyri_info, v1.3, 2016-03-21. http://data.copticscriptorium.org/urn:cts:copticDoc:papyri_info.

Coptic SCRIPTORIUM. 2016c. *shenoute.a22*, urn:cts:copticLit:shenoute.a22, v1.6.1, 2016-03-28. http://data.copticscriptorium.org/urn:cts:copticLit:shenoute.a22.