Towards a Book of English: MorphAdorner and the Annotation of Large-Scale Early Modern Text Corpora

1 Introduction and Summary

The following is a shortened and revised version of a proposal for a project that has been funded by the Mellon Foundation. The project involves the linguistic annotation of the TCP corpora with MorphAdorner, a Natural Language Processing tool suite that was first developed by Philip R. Burns in the context of the WordHoard and MONK projects.[1] The work, which will be done by Philip Burns and Martin Mueller, will involve substantial improvements to the software and its associated training data. We see it as a first and important step towards a larger project that one might call English Epochs Electronically or The Book of English and which would over time create

- A large, growing, collaboratively curated, interoperable, and public domain corpus
- Of written English since its earliest modern form
- With full bibliographical detail
- And light but consistent structural and linguistic annotation. [2]

While MorphAdorner is a broadly based NLP tool suite its chief comparative advantage lies in its treatment of Early Modern data, and it is the only such tool fully in the public domain with special rules and training data for the annotation of Early Modern text corpora. The proposed work will build on that comparative advantage and focus on improving methods for dealing with the representation of the orthographic and typographic conventions of Early Modern print culture in digital surro-

---

[1] WordHoard, an application for the close reading and analysis of deeply tagged texts, is available from http://wordhoard.northwestern.edu. It provides access to lemmatized and morphosyntactically tagged texts of Early Greek epic, Chaucer, Spenser, Shakespeare, as well as a collection of 320 English plays written between 1515 and 1660. For the MONK project see http://monkproject.org.

[2] The phrase "English Epochs Electronically" with its abbreviation EEE is shamelessly plagiarized from the German project DeutschDiachronDigital(DDD), first proposed by Anke Lüdeling,Thorwald Poschenrieder and Lukas Faulstich in "Deutschdiachrondigital-- Ein Diachrones Korpus des Deutschen." *Jahrbuch für Komputerphilologie* (2004): 119-36 (http://computerphilologie.uni-muenchen.de/jg04/luedeling/ddd.html). Such projects take time: a reference corpus of Old German is now under construction (http://www.deutschdiachrondigital.de). I have written about the English version of such a corpus in "The Book of English: Towards Digital Intertextuality and a Second-Generation Digital Library." In *Electronic Publishing: Politics and Pragmatics*. ed. Gabriel Egan. New Technologies in Medieval and Renaissance Studies, Vol. 2. Tempe, Arizona 2010, pp. 185-1204.

gates, such as the EEBO-TCP texts. Much of the work, however, will benefit the annotation of any corpus. In particular, the work will focus on

1. tokenization and the intersecting problems of identifying abbreviations and establishing sentence boundaries
2. detecting and tagging non-English text regions
3. procedures for partial or iterative annotation of heterogeneous text regions where these are explicitly marked in XML encoding
4. multi-word expressions
5. Fully TEI-compatible output options
6. RESTful services for Web-based delivery of MorphAdorner services
7. More diverse and granular training data
8. Automatic detection and correction of incompletely or incorrectly transcribed words

In our work on MorphAdorner 2.0 we will continue the close collaboration with Abbot, the "conversion tool for text interoperability" developed by the Centre for Digital Research in the in the Humanities at the University of Nebraska (http://abbot.unl.edu). Abbot grew out of efforts in the MONK project to harmonize corpora encoded with different flavours of TEI. The highly interoperable TEI texts produced by Abbot are an excellent input for MorphAdorner routines. In turn the output of a workflow that leads from TCP texts or OCR generated surrogates of Early Modern texts through Abbot to MorphAdorner creates texts whose structural and linguistic metadata greatly improve the opportunities for corpus-wide inquries. Metadata both level and articulate difference. They create "diggable data" and will spur the refinement of search engines that can advantage of them, such as

- ANNIS (http://www.sfb632.uni-potsdam.de/d1/annis/)
- CQPWeb (http://cqpweb.lancs.ac.uk),
- KorAP(http://www.ids-mannheim.de/kl/projekte/korap/),
- Philologic(https://sites.google.com/site/philologic3/),

The deliverables of this project will include

1. MorphAdorner 2.0
2. A diverse set of training data for the annotation of Early Modern corpora
3. An "Abbotized" and "MorphAdorned" corpus of ~42,000 EEBO, ECCO and Evans TCP texts, created on the basis of iterative runs of MorphAdorner over various subsets of these Early Modern corpora

4. Detailed analyses of how well or badly MorphAdorner does at providing linguistic annotation for different subsets of Early Modern texts segmented by time and genre. These analyses will accompany the project in all its phases and will be available over the Web.

2 MorphAdorner: Its history and future plans

2.1 Why bother with linguistic annotation?

If you are a linguist the uses of a linguistically annotated text are obvious. If you are not — and most people are not — you may not take kindly to the jargon of that discipline and feel some sympathy with the charge that Shakespeare's peasant rebel Jack Cade brought against the Lord Say:

> It will be proved to thy face that thou hast men about thee
> that usually talk of a noun and a verb, and such abominable
> words as no Christian ear can endure to hear.
> (*2 Henry 6*, 4.7.35-39)

There is a small community of historical linguists for whom annotated Early Modern texts are an important resource, but the linguistic annotation of sizable numbers of Early Modern texts needs to be justified in terms of its benefits to the much larger and varied scholarly communities that profit from the corpus-wide exploration of Early Modern texts but have little or no interest in the technical problems of professional linguists.

This justification is found in the fact that some forms of linguistic annotation are a basic requirement for making text corpora computationally tractable and for supporting any of the more complex uses that go beyond delivering texts for reading. A printed page is a form of encoding that calls on extraordinary and largely tacit skills on the part of human readers. Light linguistic annotation is best thought of as metadata at the word level that introduce some rudiments of readerly knowledge in a manner that a machine can process. Brian Athey, a medial researcher at the University of Michigan observed that "Agile data integration is an engine that drives discovery" (http://blog.orenblog.org/?s=agile). In the world of text-centric disciplines such agility is achieved by XML based corpora in which texts are surrounded with metadata at the top level of bibliography, the mid level of document structure, and the bottom level of individual words. If the levels of this triple-decker structure of metadata can be queried separately or in combination you have the foundations for the claim of the MONK Project's acronym that "Metadata Offer New Knowledge."

2.2 MorphAdorner history

MorphAdorner 1.0 was developed by Phil Burns in response to various problems that arose in the context of the WordHoard and MONK projects. It is a Java command-line program that acts as a pipeline manager for a variety of Natural Language Processing tasks. In the context of moving towards MorphAdorner 2.0 the following tasks will receive particular attention:

1. Tokenization
2. Sentence splitting
3. Language detection
4. Finding multi-word units
5. Element aware annotation
6. Fully TEI-compatible output options
7. Workflows that deliver MorphAdorner services over the Web and integrate it with other tools
8. Automatic detection and correction of incompletely or incorrectly transcribed words

Sustained work on MorphAdorner ended with the MONK project in 2009. There is a useful comparative evaluation of MorphAdorner and some other NLP suites by Matthew Wilkens (http://mattwilkens.com/2009/01/27/evaluating-pos-taggers-conclusions/).

MorphAdorner is a fast program. On a first generation Intel I7 3Ghz based system it will annotate a million words per minute (all of Shakespeare). Running or re-running MorphAdorner with quite large data sets does not create bottlenecks for workflows, and large tasks can be divided among several machines. More sophisticated forms of parallelization are possible, but they are not a high priority item.

2.3 Training data

Iterative improvement of training has been a key task in the development of MorphAdorner. In the course of the MONK project, the following texts were annotated with MorphAdorner after being transformed by a prototype of Abbot into a shared version of TEI-P5 from the different earlier flavours of TEI in which they had been encoded:

Collection	# Works	# Authors	# Words (in millions)
Shakespeare	42	1	0.9
TCP-EEBO (1500-1700)	691	281	39.4
TCP-ECCO (18th century	1077	196	34.2
Nineteenth Century Fiction	250	102	39.4
Documenting the American South	113	68	8.6
Early American fiction	111	16	5.2
Wright American fiction (1851-75)	301	159	23.5

The training data for the MONK corpus were developed from WordHoard data, and were reviewed by Mueller with considerable help from Burns. The procedures involved

1. Tagging a new data set on the basis of existing training data
2. Correcting the new data by hand or with semi-automatic routines and adding them to the training data
3. Repeating #1 and #2 with additional data

For 16th and 17th century texts we enriched the original Shakespeare data by adding half a dozen works from different genres. The 19th-century training data were developed by tagging a dozen American and English novels with the Early Modern training and then correcting them. The 18th-century data were tagged with a combination of the Early Modern and 19th-century training data.

The Monk corpus in its current form is the uncorrected product of tagging its 150 million words with the training data just described. Since then, a subset of the MONK corpus, about six million words from 280 non-Shakespearean Early Modern plays, went through a very thorough review by Mueller and five Northwestern undergraduates in the summer of 2010. Plays make for good training data because they are linguistically varied, ranging from colloquial or dialect speech through formal prose to verse of various kinds. The training data now available to MorphAdorner are larger, more varied, and more accurate than they were at the end of the MONK Project.

2.4 NUPOS tag set

MorphAdorner can work with any tag set, but the default tag set is NUPOS, which was designed by Mueller in the course of the WordHoard project. Mueller was work-

ing at the time with a Shakespeare text tagged with the CLAWS tagger and a Chaucer text annotated by Larry Benson with an ad hoc tag set. He discovered that you could with minimal compromise merge the two sets in a hybrid set capable of expressing major morphosyntactic phenomena of English from Chaucer to the present day. This set in turn could be seen as a pure subset of the more granular tag set required for the rich morphological structure of Homeric Greek.[3]

NUPOS was used in WordHoard and subsequently in the MONK corpus of some 150 million English words from the early 1500s into the 20th century and from both sides of the Atlantic. The tag set differs in two important respects from the widely used Penn Treebank and CLAWS sets:

- it treats contracted forms as single tokens although it provides separate analyses for each component
- it does not split off the "Saxon genitive" as a separate token

Both these choices have to do with the need to apply POS tags consistently across texts with a wide range of orthographic and morphological variance. To begin with the latter, before 1750 a phrase like "her mother's daughter" was more likely to be spelled "her mothers daughter." Conversely, a spelling like "canoe's" is more likely to be a plural than a genitive. In a diachronic corpus you need a solution that finds a consistent solution for the '-s' suffix with or without an apostrophe.

Contractions are mostly a form of spoken language. In modern English, they nearly always use the apostrophe to mark the boundary between their separate components. But this is not true in earlier English, where you find 'wilbe', 'Ile', 'sheel', 'didna', or 'wiltow', which are awkward to split. Moreover, where two words are written without a space between them, the practice typically reflects the shared sense of a language community that the words are in an important way one unit. Hence the decision to base tokenization on orthographic practice but to allow for distinct analysis of the components.

---

[3] For a fuller account of the NUPOS tag set see my "NUPOS: A part of speech tag set for written English from Chaucer to the present" (http://panini.northwestern.edu/mmueller/nupos.pdf)

3 The changes in MorphAdorner 2.0

3.1 The universe of digitally transcribed Early Modern English texts

While MorphAdorner is a general purpose tool kit for various NLP tasks, the development work in this project will concentrate on features that will improve the linguistic annotation of Early Modern English texts, i.e. texts published before 1800 and catalogued in the English Short Title Catalogue (ESTC). Most digital surrogates of ESTC texts come from one of the following sources:

- The archive of pre-1700 books in the Text Creation Partnership (TCP) collection of Early Modern English Books (EEBO). These texts have been manually transcribed into SGML. They currently include upward of 30,000 titles, but the collection is expected to grow to ~70,000 and include at least one transcription of every unique title before 1700.
- About 2,000 manually transcribed TCP texts from Eighteenth-Century Online (ECCO), less than 1% of surviving imprints from 18th century Britain.
- About 6,000 manually transcribed TCP texts (15%) of the 40,000 titles available in the Evans Early American Imprints Collection.
- "Dirty" OCR of ~200,000 18th century imprints from the ECC, about two thirds of 18th century imprints in the ESTC. The transformation of these OCR surrogates into good enough TEI versions is a major goal of the 18thConnect initiative and its director, Laura Mandell (http://www.18thconnect.org)
- An unknown number of 18th-century imprints (15,000 +/- 5,000?)) that exist in Google books, are in principle in the public domain, and are also on the horizon of the data curation initiative of 18thConnect.

By contemporary standards, the aggregate of these books is no longer a Gargantuan thing. It is a large, but finite and relatively static corpus of between ten and twenty billion words. MorphAdorner 2.0 will be an excellent tool for the linguistic annotation of any of these texts, especially if they have gone through the Abbot workflow developed by Brian Pytlik Zillig and Stephen Ramsay at the University of Nebraska. As described in more detail below, MorphAdorner 2.0 can also play a role in algorithmically based identification and curation of incompletely or incorrectly transcribed words.

3.2 Tokenization

Extracting words and sentences from a text are fundamental operations and form the basis for higher-level language processing functions. Word tokenization splits a text into words and punctuation marks. Sentence splitting assembles the tokenized text into sentences.

The first step in word tokenization is recognizing word boundaries. A tokenizer typically uses white space such as blanks and tabs as the primary cue for splitting the text into tokens. Punctuation marks are split from the initial tokens. This is not as easy as it sounds. For example, when should a token containing a hyphen be split into two or more tokens? When does a period indicate the end of an abbreviation as opposed to a sentence or a number or a Roman numeral? Should a comma always be split out from the text in which it appears? (Not when the comma appears within a number, for example.)

When should a single quote be split from a word? Early modern English included many contractions such as "'tis" with a leading quote. Sometimes a period terminates an abbreviation and a sentence at the same time. Sometimes whitespace is missing around a period and must be added to achieve correct tokenization (e.g., "stairs.The" should read "stairs. The").

Roman numerals in older texts exhibit considerably more orthographic variation than contemporary usage allows. For example, the letters 'j' and 'u' are often substituted for 'i' and 'v'. Runs of letters may exceed the nominal length, e.g., 'iiiii' may be used where 'v' would normally appear in current usage. Particularly in early modern texts, numerals may be preceded and/or followed by a period, as in '.XVI.' Some Roman numerals are often followed by superscripted letters, as in "DCCXXV^o," where the Latin inflection markers need to be stripped in order to retrieve the base form "DCCXXV." MorphAdorner attempts to recognize many of these variants so that they can tagged as numbers.

Little problems of this kind pose no difficulties for human readers but are surprisingly difficult for machines. In general, the current tokenizers of MorphAdorner use a number of heuristics and a list of common abbreviations to produce a sequence of punctuation and spellings that will be consistent with the subsequent operations of sentence boundary identification, part of speech tagging, and lemmatization.

3.3 Sentence splitting

MorphAdorner's current default sentence splitter uses the ICU4JBreakIterator class (from the International Components for Unicode) along with a large set of heuristics for determining if two or more sentences generated by ICU4JBreakIterator should be joined into one sentence (http://site.icu-project.org/home/why-use-icu4j). The heuristics include special treatment of sentence-ending brackets (right parenthesis, right bracket, and right brace), abbreviations, and interjections. The resulting sen-

tence extraction is not perfect but is better than ICU4JBreakIterator's splitting and much better than naive splitting methods.

To further improve sentence splitting , we will incorporate portions of the Punkt algorithm for tokenization and sentence splitting, which treats abbreviations as a special form of collocation, where a character string habitually collocates with a final dot (http://www.linguistics.ruhr-uni-bochum.de/~strunk/ks2005FINAL.pdf).

The implementation of the Punkt algorithm will not only improve sentence splitting, it will also lead to a much more reliable identification of abbreviations. The TCP texts include a great many scientific, theological, and other learned texts with thousands of obscure and rarely consistent abbreviations. The Punkt algorithm was developed in part to help contemporary biomedical researchers to keep track of abbreviations. Scholars of Early Modern religious disputes will be just as grateful for a much more robust system of identifying abbreviations for books of the Bible, the Church Fathers, and the theologians of the 16th and 17th centuries.

The Punkt algorithm is language agnostic and will also help greatly with the sentence splitting of text in other languages.

We expect to incorporate Punkt into MorphAdorner as follows.

1. We will decide if a word is an abbreviation or a sentence-ending ordinary word by looking at its usage across an entire corpus.
2. We will implement a program that generates a corpus-specific abbreviation list using the Punkt algorithm from step (1). The output will replace/enhance the current manually specified lists of abbreviations.
3. We will implement generic tokenization and sentence splitting for non-English text using the latest ICU algorithms. We will supplement the ICU algorithms using the Punkt-derived abbreviation lists to improve ICU's regular expression-based tokenization and sentence-splitting methods. We will look at using the existing heuristics that correct certain badly punctuated English texts for languages other than English.
4. We will implement the basic Punkt tokenizer and sentence splitter as an alternative to the existing tokenizers and sentence splitters. This is mostly to provide enhanced baselines for non-English portions of texts. We expect the combined ICU and Punkt abbreviation detection approach will yield superior results compared to a plain Punkt implementation.
5. For TEI XML texts we will implement a multi-language tokenizer and a multi-language sentence splitter. These will adapt tokenization and sentence split-

ting on the fly to the language specified by the nearest ancestral xml:lang attribute of a text-enclosing TEI tag.

3.4 Language detection

Early Modern English texts include an astonishing variety of non-English stretches, sometimes a few words and sometimes whole paragraphs or chapters. In the TCP texts, foreign passages are not reliably marked as such. In texts before 1800 orthography is a very unreliable guide to identifying the language of a word: English, French, and Latin spellings often overlap in confusing ways.

MorphAdorner includes a simple statistical method based upon character n-grams and rank-order statistics to determine the principal language of a text and list of possible secondary languages.[4] During the Monk project we used this language detector to identify documents that were nominally English but contained much unmarked foreign language text. We found that the language detector test scores offered a reliable way to identify non-English text regions. The specific language labels were not quite so reliable. Especially in short stretches of text it is difficult to distinguish between Latin and its French, Italian, or Spanish daughters. Similarly Scots and English are hard to tell apart, especially in very early texts.

In MorphAdorner 2.0 we will use the existing language detection methodology to develop a preprocessing utility that adds appropriate language markup to sections of text in TEI XML documents. If all or most of the words in a particular container element (<p> for paragraph or <l> for line of verse) are identified as, say, French, the program will add a language attribute in the form of 'xml:lang="fr"'. Inline snippets will be tagged with <seg> elements and appropriate language attributes. When MorphAdorner runs through a text it will then skip elements with a non-English language attribute. This is a rough kind of justice and may require some manual tweaking to make it perfect, but it will be a great improvement over the current state. By excluding non-English text regions with substantial reliability it will reduce errors in the tagging of English texts.

MorphAdorner 2.0 will make no attempt to map non-English words to their lemmata or provide morphosyntactic descriptions for them. Latin is by far the most dominant foreign language in early modern texts: the volume of Latin in some 30,000

---

[4] William B. Cavnar and John M. Trenkle." N-Gram-Based Text Categorization," *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval.* 1994. (http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.53.9367) MorphAdorner's implementation follows one written by Frank S. Nestel.

EEBO-TCP texts is between five and ten times the size of Shakespeare's works. It might be a worthwhile project to apply linguistic annotation to that sizable body of Latin with an appropriate tool. By identifying Latin sections with considerable accuracy MorphAdorner 2.0 will at least lay the foundation for such a project.

3.5 Multi-word units

A multi-word unit is a special type of collocate in which the component words comprise a meaningful phrase, e.g. "Knight of the Round Table." In the WordHoard project we used the notion of a pseudo-bigram to generalize the computation of bigram (two word) statistical measures to phrases (n-grams) longer than two words, and to allow comparisons of these measures for phrases with different word counts.
Any multi-word phrase can be looked at as a "pseudo-bigram" having a left part and a right part. The Fair Dispersion Point Normalization offered by Silva et al. transforms any phrase of any size into a "pseudo-bigram" which reflects the "glue" among the individual words in the original phrase.[5] Loosely speaking, the Fair Dispersion Point Normalization splits the phrase at each possible position, computes a bigram association measure for each split, and uses the arithmetic average (mean) of the individual association measure values as the association measure value for the entire phrase.

For example, the phrase "Knight of the Round Table" can be split into a bigram in four different ways:

Knight	of the Round Table
Knight of	the Round Table
Knight of the	Round Table
Knight of the Round	Table

If we treat each "half" of the split phrase as a single word, we can compute a bigram association measure value for that split. Taking the arithmetic mean of the four association measure values gives the Fair Dispersion Point Normalization Value.

The LocalMAXS algorithm, which is language independent, accepts a text as input and generates multi-word units from that text. For each phrase, the algorithm looks at all subphrases which are one word shorter contained within that phrase as well

[5] Joaquim Ferreira da Silva and Gabriel Pereira Lopes. A Local Maxima method and a Fair Dispersion Normalization for extracting multi-word units from corpora. (http://ssdi.di.fct.unl.pt/aadm/aadm1011/slides/MOL99.pdf)

as all superphrases which are one word longer that contain the phrase. When the phrase's association measure achieves its maximum value compared to those for all the subphrases or superphrases, the phrase is considered a probable multi-word unit (compositional phrase).

WordHoard applied the LocalMAXS algorithm to the pseudo-bigrams in a text to identify potential compositional phrases that "stand out" in that text. The early version of LocalMAXS implemented in WordHoard was time and memory intensive, and not suited to processing large corpora. Later versions of LocalMAXS significantly reduce the time and space requirements and render LocalMAXS more suitable for use with large corpora. We intend to generalize the algorithm we implemented for WordHoard to allow its use with documents produced by MorphAdorner.

There are several things you can do with multi-word phrases once you have identified them. It would be possible to use inline tags (e.g. the <seg> tag of TEI) to identify such phrases. Users might find it more helpful to encounter multi-word phrases in the form of lists. Once a text has been linguistically annotated it is a simple matter to construct a dictionary of all its lemmata with their counts. By the same token one could compile lists of all recurring multi-word phrases with their document and collection frequencies in a corpus. Such a dictionary will be useful for the analysis of many intertextual relations.

While Named Entity Extraction is not on the agenda for MorphAdorner 2.0 we will do a preliminary analysis of the question whether named entities can be usefully treated as a subset of multi-word phrases. We discovered in the MONK project that the business of discovering names is quite domain specific: tools designed for modern data did not work well for earlier texts, and within Early Modern texts it is also likely that segmentation by domain will be important. But it will be worth reviewing the results of the LocalMAXS algorithm and asking whether multi-word names can be extracted algorithmically and with a good balance of precision and recall from general multi-word lists. We are agnostic about the outcome of this preliminary analysis.

3.6 Iterative, partial, and element aware annotation

MorphAdorner is an unusually "element aware" tool. As it moves through an XML document, it distinguishes between 'soft', 'hard', and 'jump' tags. 'Soft' tags are like HTML 'inline' elements (<i>, <b>, etc.). Their presence is ignored in the sense that it makes no difference whether the words contained in them are or are not wrapped by those elements. 'Hard' tags (, <p>) mark discursive boundaries that are not crossed by sentences. 'Jump' tags (e.g. <stage>, <note>) interrupt the reading order

of the words before and after. MorphAdorner "jumps" across those tags, splices the words separated by them, and tags the content of the jump tags as a distinct unit.

This capability is important in the tagging of Early Modern texts, where marginal notes and embedded content of other types are extremely common. MorphAdorner 2.0 will build on this capability and offer algorithms for treating different text regions differently, whether by using different training data or tagging rules. Many texts include substantial regions that are not organized as sentence sequences. Tables of content, other tables, bibliographical notes, chapter headings, stage directions, and similar forms of "paratext" fall in this category. In a TEI-encoded document such text is contained in elements like <bibl>,<cell>, <head>, <item>, <note>, or <stage>. MorphAdorner 2.0 will be able to exclude such elements in a first pass and in a second pass tag them as atomic word units that make no claim to sentencehood.

The same technology that supports initial annotation of just a part of a text would also support the re-adornment of particular sections for whatever reason. The ability to re-adorn parts of a text is a very important tool for reducing error rates.

Are such complex routines worthwhile in the literal sense of the word? With cultural heritage objects the calculus of time is complicated by the fact that the further back you go in time, the fewer objects there are. Temporal distance adds value to each data point in a progressively vanishing past. The measure of the worthwhile changes with the mass of objects. Book from the early 1500's are counted by the dozens or hundreds per year. Newspapers from the late 1800s run into billions of words per year. MorphAdorner 2.0 will be a tool that supports the creation and correction or revision of quite fine-grained annotation for investigators who put a high value on the rare.

3.7 Output formats

MorphAdorner supports various forms of tabular and inline XML output. Tabular output is a critical tool for creating or revising training and for reviewing test data. In tabular output each token is a row in a table, and at a minimum there will be columns for

1. the ID assigned to each token
2. the spelling of the token
3. the lemma assigned by MorphAdorner
4. the part of speech tag assigned by MorphAdorner

MorphAdorner can also add additional columns that contain

1. The Xpath for the token
2. the previous token
3. the next token
4. A five-word KWIC context on the left
5. A five-word KWIC context on the right

In this very verbose output each data row can be thought of as a lexicographer's slip with sufficient data to evaluate the use of the word for most practical purposes. Data in this form are also very manipulable in relational database environments and support "group by"", count, filter, and sort routines that let you construct a dynamic and frequency-based dictionary of a corpus. These routines are critical in the creation and revision of training data as well as for the review of test data.

In the inline XML output of MorphAdorner each token is wrapped in a <w> element, and the metadata generated by MorphAdorner are added as attributes of that element. The default XML output of MorphAdorner 1.0 uses a set of attributes that require some modification of the TEI schema. In MorphAdorner 2.0 we will create output that validates under unmodified TEI P5 or some pure subset of it.

There is not at the moment a fully satisfactory and out-of-the-box model of lightweight linguistic annotation in TEI P5. Laurent Romary at INRIA has been working on an ISO proposal for a Morphosyntactic Annotation Framework(ISO/DIS 26411), which will be submitted for final approval later this year. Discussions continue about reconciling this framework with the TEI schema. The MAF framework can be implemented in stand-off markup, which is more flexible and powerful in many ways but may be harder for search engines to ingest. (http://atoll.inria.fr/~clerger/MAF/html/body.1_div.10.html) If these discussions produce results it may be worthwhile to integrate the MAF framework into MorphAdorner. But this would not be a trivial task.

The encoders of the TCP texts did not anticipate the subsequent addition of linguistic annotation. Adding such encoding to the texts is a different matter from annotating the British National Corpus, which was created from the ground up as a linguistic corpus. In adding linguistic annotation to the TCP texts (or similar corpora) MorphAdorner 2.0 must solve several pesky problems. Most of them have to do with the tagging of typographical phenomena that begin in the middle of one word and extend across one or more following words. Italics and superscripts account for most them. These are very uninteresting but surprisingly tricky problems to solve.

The current version of MorphAdorner "pretty-prints" its XML output on separate lines and encloses the white space of the source documents in separate <c> tags. For some purposes it may be preferable to create an XML output format that does not emit explicit <c></c> tags for spacing and does not "pretty-print" the output on separate lines. We intend to allow such output using either as a configurable option or as a post-processing program.

3.8 Parallel processing, RESTful services and Web services integration

MorphAdorner provides a skeleton for pipelining processes. MorphAdorner's use of Java interfaces allows easy substitution of different implementations into the pipeline (e.g., Template method pattern). It is straightforward to wrap adornment processes as web services using Rest-like interfaces. The web services can be accessed from any programming language and system that knows how to send and receive HTTP requests, or even a plain web browser. The services are automatically parallelized because of the way HTTP servers work. Many clients can access the same web service simultaneously.

In MorphAdorner 2.0 we will expose a variety of MorphAdorner services as RESTful web services, including

1. Returning the lemma for a spelling;
2. Returning a standard spelling for a variant;
3. Adorning a segment of plain text;
4. Adorning or re-adorning a segment of TEI encoded text;
5. Adorning a full TEI encoded text;

We intend to use the Restlet library to implement the web services. Restlet eases the development of Java-based RESTful services. Restlet also provides for generating OSGI bundles for services and allows services to be moved to the Android platform and Amazon cloud services. While implementing these latter functionalities is out of scope for this project, it is useful to know that Restlet supports them.

To demonstrate the MorphAdorner services, we will create a "drop-box" service that will accept one or more TEI XML texts through the web. The service will optionally invoke Abbot to preprocess the texts before adornment. The output will be returned as adorned texts in a ZIP or TAR file. We will implement a simple user authentication facility for this service. This will allow users to view the progress of

their adornment tasks as well as to retrieve the results without having to stay logged in.

The drop-box service will provide hooks for competent programmers to create distributed processing back ends. These could interface with popular cluster management software such as Condor, PBS, DRMAA, or Torque. Alternatively the back ends might use existing toolkits for integrating web services such as the Opal toolkit from the San Diego Supercomputing Center, or the SEASR toolkit from the NCSA. It would be useful to support multiple service machines transparently on the back end since adorning large batches of texts can be computationally expensive.

MorphAdorner web services will allow integration with other programs in a variety of ways. The services could be invoked from a Project Bamboo Services Platform layer. Embedding MorphAdorner web services in an OpenSocial wrapper would allow MorphAdorner services to appear in those social networking platforms that support OpenSocial. Existing scientific workflow systems such as Taverna, Kepler, and Triana could be used to create pipelines merging MorphAdorner web services with those from other sources. While we cannot investigate all of these types of connections during the course of our project, we expect others will be eager to try them out once we make the MorphAdorner web services implementation available.

We envisage that MorphAdorner users will create chains of web services that involve one or more MorphAdorner web services as part of the chain. A particularly useful example is a pipeline that starts with a text or corpus and moves it through Abbot, then through MorphAdorner, and then through a text indexer such as Philologic: that is to say a workflow that

1. submits a text/corpus to routines that turn it into a more standardized or interoperable object
2. adds metadata at the word level that support corpus-wide analytical operations despite differences in the surface forms of words
3. moves the annotated text/corpus into a search environment that takes advantage of the query potential of the metadata separately or in combination.

The example is not entirely hypothetical. MorphAdorner plays well both with Abbot and Philologic. The Abbot/MorphAdorner workflow has been extensively tested and used in the MONK project. We will be implementing an updated version of this combination as part of the drop box service. The Philologic search engine at the University of Chicago is currently undergoing a thorough revision and has used MorphAdorned data sets on a continuing basis. Other approaches for text indexing

include moving the adorned texts into an XML database such as eXist or BaseX. We may be able to build a sample pipeline of this type as part of this project if time permits.

3.9 More training data for MorphAdorner 2.0

To repeat a point already made, customized training data are the key to good enough linguistic annotation. With the exception of a few boutique projects linguistic annotation is a pretty rough enterprise. It is very difficult to drop below an error of 3% rate. You cannot manually correct large data sets, but you can customize training data by region, period, and genre. Characteristic patterns of error can be addressed by a "divide and conquer" strategy of tagging different batches of texts with appropriately modified training data.

Because it has become much cheaper and faster to process large data sets, the task of customization has also become easier. For instance, you can in a matter of hours run 5,000 EEBO texts through MorphAdorner, retrieve a sample of 50,000 annotations in tabular form, move them into a tool like Microsoft Access in less than an hour, and review the sample manually for clustering of errors by period, genre, or region. It is then a matter of days rather than weeks to work through that list in a triage mode. Some recurring errors can be fixed by tweaking the algorithm. Others will yield to improved training data. You may also come across texts (e.g. texts before 1500) that cannot be automatically annotated with acceptable error rates. It is at least useful to know what and where they are. The final result of such an improvement in training data will be a basic set of training data for each century with customized subsets.

Genre is probably the best guide to useful subsets. The linguistically annotated versions of sermons, travel writings, novels, letters, histories are likely to have characteristic errors. If there were a flexibly searchable version of the ESTC catalogue, it would be a trivial task to extract such batches and process them accordingly. There is not yet such a thing, but with some ingenuity and patience it is possible to assemble sample batches that follow a rough grid of time and genre and will work well enough.

We will conduct an experiment with several dozen 18th-century novels, using quotation marks and other typographical indicators to tag spoken sections with a <said> tag and analyze whether annotating spoken and narrative sections separately reduces the error rate significantly. 'Yes' and 'No' would be equally useful answers to this question.

3.10 MorphAdorner as a curation tool

MorphAdorner has considerable potential as tool for identifying and correcting incompletely or wrongly transcribed words in the TCP texts as well as in OCR generated surrogates of Early Modern texts. In MorphAdorner 2.0 we will make substantial improvements to those capabilities, which are structurally equivalent to spell-checkers in word processors.

If you think of the annotation of a large corpus as an iterative process, the processing tool can in a second or third run draw on the experience it has previously gained. From a first pass over a corpus it is not difficult to construct a frequency-based dictionary that associates each surface form with a count of the documents in which it is found (document frequency) and the total count in the corpus (collection frequency). The authority of a spelling can be measured roughly through a combination of its document and collection frequencies: a spelling that occurs once in each of twenty documents has more authority than a spelling that occurs twenty times in one document. Where MorphAdorner encounters an incomplete or otherwise suspect spelling it can look for close matches in an expanding horizon of immediate or more distant textual neighborhoods and make appropriate corrections based on the authority of such matches.

Such corrections will be entered into the source text in a manner that allows for subsequent algorithmic or manual review. For instance, an "Abbotized" TCP text will wrap an incompletely transcribed word with an <unclear> element. If such a word is corrected algorithmically, it can be wrapped in a <corr> element. There are other ways of avoiding the traceless overwriting of presumed errors.

A full implementation of "expanding neighbourhoods" will depend on being able to use the ESTC as a sophisticated text categorization tool. That is not likely to happen within the time frame of this project. It may be possible to generate a crude time grid from the not always reliable publication dates in teiHeader files. But in several experiments we have found that you get quite good results by simply treating an entire corpus as an undifferentiated neighbourhood. This is especially true of missing characters in the middle of a word in TCP. There are several million of them in the EEBO texts, and a good third of them can be corrected in this manner with high confidence.

Wrongly joined or wrongly split words are a quite common source of error in TCP texts and pose challenges for tokenization. We are not sure whether there are good algorithmic ways of dealing with such errors.

MorphAdorner 2.0 can also be used in connection with other curation tools, for instance Laura Mandell's Typewright project. The workflow may be different, but the principle is the same: you use data from the document and collection frequencies of spellings in Morphadorned texts to review suspect OCR transcriptions. (http://www.18thconnect.org/typewright/documents)

3.11 Miscellaneous internal optimizations

There are a number of miscellaneous optimizations and improvements that we can make while working on other aspects of MorphAdorner. These include improvements to the internal caching mechanisms; potential use of alternative thread-safe Java collections to enable easier parallelization; use of compressed text files for lexicons, spelling maps, and other configuration data; and a variety of other ways to reduce the time and space requirements of adornment processing.

4 Deliverables, benefits, and integration with other projects

The deliverables of this project will include:

1. MorphAdorner 2.0 with the improvements and extensions outlined above.
2. A set of training data based on some combination of genre and period and improved through iterative runs over different sample of test data.
3. An annotated TEI-P5 compliant corpus of all EEBO, ECCO, and Evans TCP texts currently available, processed with Abbot and MorphAdorner 2.0.
4. Detailed analyses of how well or badly MorphAdorner does at providing linguistic annotation for different subsets of Early Modern texts segmented by time and genre. These analyses will accompany the project in all its phases and will be available over the Web.

It is worth repeating that linguistic annotation is neither an end in itself nor intended primarily for linguists. It provides metadata at the word level that greatly improve the algorithmic tractability of texts, especially when used in conjunction with bibliographical and structural metadata. The "agile data integration" enabled by such metadata increases the power, speed, and complexity of corpus-wide inquiry for a wide variety of thematically, historically, or rhetorically based inquiries.

Until 2015, when the TCP texts will begin to pass into the public domain, access to the annotated texts will be limited to individuals who are associated with institutions that have licensed access to the source texts — a very substantial number of students and scholars all over the world. The existence of so large a corpus in a standards compliant form is very likely to spur the development or adaptation of

search tools that can take advantage of the query potential of this TEI encoded and linguistically annotated corpus.

In recent years we have seen increasingly sophisticated and NLP based analyses of the ideological components of political rhetoric, whether for scholarly or pragmatic purposes. George Lakoff's *Metaphors We Live By* (1980) was an important book in establishing the framework for this type of analysis. The MorphAdorned corpus of Early Modern English will be of considerable use in pursuing (dis)continuities of ideologically charged language over time. While contemporary NLP tools work very well with modern data they stumble over the orthographic or morphological variance of older English. It is a trivial task to derive from a MorphAdorned text a version of that text with standardized modern spelling, but as soon as you do this, a group of 16th century sermons may all of a sudden be (almost) comparable on an "apples to apples" basis with a contemporary campaign speech. A high percentage of the EEBO corpus consists of political/religious or broadly ideological prose. A MorphAdorned EEBO corpus offers a very promising high road to *longue durée* approaches to the study of rhetoric over time.

As a stand-alone and Web-accessible processing tools, MorphAdorner 2.0 will make life easier for individual scholars who need to create linguistically annotated texts for particular projects. The tabular output of MorphAdorner makes it easy and not especially time-consuming to correct a small corpus to any level of perfection desired. But we think that the greatest benefit of MorphAdorner will from its close association with Abbot as a combination of tools that can provide standards -based and highly interoperable textual data sets across a wide variety of genres and periods.

Abbot and MorphAdorner grew up together in the context of the MONK project and between them constructed an interoperable corpus of 2,500 texts (150 million words) from half a dozen corpora, all of them validating under some flavour of TEI, but for practical purposes incommensurable because of various and usually minor, but pesky, differences. The biggest lesson of the MONK project was that you could in fact create interoperable data sets without compromising the integrity or expressivity of each separate set. The Abbot project is making excellent progress in creating a version of its MONK prototype that is at once more flexible and more robust. MorphAdorner 2.0 will produce comparable advances. Together these two programs add up to a closely linked and solid workflow for delivering interoperable and high-quality data sets of historical English.

.

A few words about scale. The plain XML files of the TCP corpora add up to something like 10GB of data. Linguistic annotation and indexing can increase increase the size of a file by two orders of magnitude. A MorphAdorned corpus of current TCP texts adds up to a terabyte, not counting its backup. To vary Senator Dirksen's quip about budgets and billions: "A terabyte here and a terabyte there, and pretty soon you'll be talking about real data." That said, a billion dollars is not what it was in the sixties, a terabyte of data in 2015 will not be what it is now, and in terms of pure storage it is not a whole lot now.

It is not, however, the ambition of this project to maintain a permanent home for linguistically annotated TCP texts. That will be the responsibility of search environments that rise to the challenge of mediating the query potential of those texts to end users. Rather, the ambition of the project is to put these annotated texts out as a provocation:

> Here is a large set of standards based linguistically annotated texts that can become the fodder for a clever search engine. Do something with them!

We are confident that at least one project will rise to that challenge. In the longer run the question of how and where to mediate large text corpora through advanced and complex query tools will be a question for major research libraries to solve through cooperative and consortial endeavours.

4.1 Error reporting

The chief goal of this project is to produce better linguistic annotation for Early Modern texts. This is a project with outcomes that can be measured in terms of error rates. Iterative reviews of MorphAdorner output based on improved algorithms and more differentiated training data are key stages in this project, and they will form the basis of evaluation and reporting. We will keep a public website of our progress with detailed reports about error rates in successive runs.

We will also differentiate between types and severity of error. In a 16th century text, the spelling 'loues' could

1. the third person singular of the verb 'love' (vvz)
2. the plural of the noun 'love' (n2)
3. the possessive case of the noun 'love' (ng1)

Assigning the POS tag 'vvz' when it should be 'n2' or 'ng1' is a more serious error than confusing 'n2' and 'ng1'. The latter correctly identifies the word class of each token but makes an error at the more granular level of morphological analysis. The former counts as an error both at a coarser and a more granular level.

Classifying and weighting errors in this way will make it possible to track improvements in error rates in a quite precise and public manner. It would count as complete success if MorphAdorner operates at error rates between 3% and 4% across all genres and periods of written English before 1800. We probably will not achieve that across the board, but we will be able to report with considerable precision about the types of materials in which that error rate is exceeded and the types of error that may be expected.

While improving the accuracy of output for Early Modern data is the chief criterion for measuring the success of this project, measurable improvements in the ease of use for the tool are also important. Right now, the use of MorphAdorner presupposes a modest competence with working at the command line. MorphAdorner 2.0 aims at being usable via a browser interface for a wide variety of small or mid-sized operations. MorphAdorner 2.0 may also involve improvements in the speed of performance. Such improvements are easily measured and reported on, but they will not make much difference: the program is already fast enough for most purposes. More important than raw speed is the ease with which users at different levels of technical sophistication can handle MorphAdorner output. MorphAdorner 2.0 will have several improvements on that front, but claims for it must be verified by users. Interaction with both the Philologic project at the University of Chicago and the Abbot project at Nebraska will give use helpful user feedback, and this will be incorporated into the final report.