Creation and Annotation of Linguistic Resources

Dr. Duygu Ataman
Spring Semester 2021

Semester Project Report:

Annotated corpus in XML format

Based on texts from
Sweet's Anglo-Saxon Primer

written by:
Eyal Liron Dolev
Student no. 20-713-897
eyalliron.dolev@uzh.ch

Creation and Annotation of Linguistic Resources: Semester Project Report

Eyal Liron Dolev

June 15, 2021

Contents

1 Material

More than a hundred years ago, in 1905, Henry Sweet published his Anglo-Saxon Primer[1]. This is a collection of short Old English texts, pre-ambled by a short grammar. The Old English corpus is in fact a collection of writings by different authors, mostly of the West-Saxon dialect. As it is virtually always the

[1]Sweet 1905.

case with a non-standardized language such as Old English, the texts contain many spelling variants. In his Primer, Henry Sweet standardized the included texts and added diacritics to mark vowel length as well as palatalizations of of velar consonants. These diacritics contain valuable information and it should be attempted to retain them when creating a digital corpus.

The Anglo-Saxon Primer exists as raw text as part of the Gutenberg Project[2]. The diacritics have been retained and converted to Unicode characters.

1.1 Potential applications

An annotated corpus in XML format of the texts in the Primer might be used by students learning Old English with the Primer and wanting to be able to search and explore the corpus, or it might be used for corpus linguistic research. The biblical passages in the corpus may also be used later for multilingual alignment and for traning machine translation models.

2 Work

In my project, it was my intention to create a small corpus in form of XML files, containing the short texts from the Primer in tokenized form, as well as a normalized ASCII form, containing no diacritics or special characters (e.g. `æ`).

In the beginning, I thought further annotation such as PoS tagging would be have to be done manually. But, fortunately, I was able to find a Python library, CLTK (Classical Language Toolkit)[3], which contains i.a. a trained PoS-tagging model for Old English.

The CLTK library was not easy to handle. It seems buggy and some parts of it are faulty, but I was able to get it to work by reverting to older versions of it.

The result is a PoS-tagged corpus in XML format, which also contains lemmas and the normalized ASCII forms.

3 Building the corpus

3.1 Source material

3.1.1 Problems

I obtained a raw text version of the Primer from the Gutenberg Project[4] [5]. From this file, I manually extracted the six texts contained in the Primer into separate text files. These files (can be found in the `texts/raw` folder), displayed the following problems:

[2] Sweet n.d.
[3] Johnson et al. 2014–2021.
[4] Sweet n.d.
[5] `https://www.gutenberg.org/files/34316/34316-h/34316-h.htm`

1. Hard wrap: there is a carriage return character at around the 65th column in each line; this means sentences are broken up in the middle, a thing that would have to be fixed for parsing the files into XML, since the `<s>`-tag holds an entire sentence.
2. Empty lines: every second line is empty. Further, there are no double empty lines separating paragraphs, which will also have to be dealt with for paragraph parsing.
3. Page numbers and line numbers are included and need to be removed.

3.1.2 Solution

The script `converter.py` deals with the raw files and creates a version that would be fit for parsing. This means, no hard wraps, one paragraph per line, and no page or sentence numbers.

For joining paragraphs together to one line, the algorithm treats each line that ends with a full stop and is shorter by at least 15% than the average sentence as the last sentence of the paragraph and joins sentences up to that to one paragraph. For removing empty lines and numbers, regular expressions were used.

3.2 Annotation

3.2.1 PoS-tagging

Since Old English is not just a rare language, but has also not been spoken or written for at least 1000 years, there was only one system available to choose from, namely the CLTK Old English tagger. The tagger was trained using the *ISWOC* treebank[6].

Tagset Refer to table 1 for a list of PoS tags used in the treebank.

Tagging Models There are several tagging models to choose from. The two that perform best, according to the documentation, are the Conditional Random Field (CRF) and the Perceptron models, with an accuracy measure of 0.827 and 0.857 respectively[8][9]. I created two versions of the XML files, one with each tagger. A short comparison can be found in the analysis section.

3.2.2 Lemmatizer

The included lemmatizer is a naive lemmatizer based on a hand-built dictionary. If an input word is not found in the dictionary, it is simply returned[10]. As will be seen later, the lemmatizer does not perform well.

[6] `http://iswoc.github.io/`
[8] Johnson et al. 2014–2021.
[9] Bech and Eide 2014.
[10] Johnson et al. 2014–2021.

```
An on-ġinn is ealra þinga, þæt is God æl-mihtiġ. Se
ġe·lēafa þe biþ būtan gōdum weorcum, sē is dēad; þis sind
þāra apostola word. Ic eom gōd hierde: se gōda hierde
sęlþ his āgen līf for his sċēapum. Ūre Ā·līesend is se gōda
5
hierde, and wē crīstene męnn sind his sċeap. Se mōna his
leoht ne sęlþ, and steorran of heofone feallaþ. Swā swā
wæter ā·dwǣsċþ fȳr, swā ā·dwǣsċþ sēo ælmesse synna.
Ealle ġe·sċeafta, heofonas and ęnġlas, sunnan and mōnan,
steorran and eorþan, eall nīetenu and ealle fuglas, sǣ and
10
ealle fiscas God ġe·scōp and ġe·worhte on siex dagum; and
on þǣm seofoþan dæġe hē ġe·ęndode his weorc; and hē
be·hēold þā eall his weorc þe hē ġe·worhte, and hīe wǣron
eall swīþe gōd. Hē fērde ġeond manigu land, bodiende
Godes ġe·lēafan. Hē for·lēt eall woruld-þing. Se cyning
15
be·bēad þæt man scolde ofer eall Angel-cynn sċipu wyrċan;
and hiera wæs swā fela swā nǣfre ǣr ne wæs on nānes
cyninges dæġe. Se cyning hēt of·slēan ealle þā Dęniscan
męnn þe on Angel-cynne wǣron.
```

Figure 1: First paragraph in the first text before conversion

```
Ān on-ġinn is ealra þinga, þæt is God æl-mihtiġ. Se ġe·lēafa þe biþ būtan
gōdum weorcum, sē is dēad; þis sind þāra apostola word. Ic eom gōd hierde:
se gōda hierde sęlþ his āgen līf for his sċēapum. Ūre Ā·līesend is se gōda
hierde, and wē crīstene męnn sind his sċeap. Se mōna his leoht ne sęlþ, and
steorran of heofone feallaþ. Swā swā wæter ā·dwǣsċþ fȳr, swā ā·dwǣsċþ sēo
ælmesse synna. Ealle ġe·sċeafta, heofonas and ęnġlas, sunnan and mōnan,
steorran and eorþan, eall nīetenu and ealle fuglas, sǣ and ealle fiscas God
ġe·scōp and ġe·worhte on siex dagum; and on þǣm seofoþan dæġe hē ġe·ęndode
his weorc; and hē be·hēold þā eall his weorc þe hē ġe·worhte, and hīe wǣron
eall swīþe gōd. Hē fērde ġeond manigu land, bodiende Godes ġe·lēafan. Hē
for·lēt eall woruld-þing. Se cyning be·bēad þæt man scolde ofer eall Angel-
cynn sċipu wyrċan; and hiera wæs swā fela swā nǣfre ǣr ne wæs on nānes
cyninges dæġe. Se cyning hēt of·slēan ealle þā Dęniscan męnn þe on Angel-
cynne wǣron.
```

Figure 2: First paragraph in the first text after conversion

4 Result

The result is six XML files, one for each text included in the Primer. The XML files are structured according to the usual scheme for annotated corpora: `<text><p><s><w>text<\w><\s><\p><\text>`. The tag `<p>` marks paragraphs, `<s>` marks sentences and `<w>` marks words/tokens. Each `<w>`-tag includes three attributes, `pos=`, `lemma=` and `norm=` containing the PoS tag, lemma and normalized ASCII version, respectively. Refer to figure 3 for an example.

4.1 Analysis

The statistical and text analysis I have conducted consists of:

- Tag counts and tagger comparison
- Word counts
- Keyword analysis with tf-idf

4.1.1 Tag counts and tagger comparison

As mentioned in section 3.2.1, the CLTK library offers several Old English tagging models to choose from. I had annotated the corpus twice, once with each of the best performing taggers (CRF and Perceptron). The results are surprisingly different. With the CRF tagger, the most common tag is `C-` (conjunction, 2314 occurrences), followed by `NB` (common noun, 1974 occurrences) and by `V-` (verb, 1727 occurrences). (see figure 4) That the most common word class in the corpus is a conjunction seems questionable.

Using the Perceptron tagger, on the other hand, the most common tag is `NB` (3089 occurrences), followed by `V-` (1827 occurrences) and `NE` (proper noun, 1663 occurrences). (see figure 5) The tag `C-` reaches the 6th place (640 occurrences). This seems much more reasonable.

```
<?xml version='1.0' encoding='utf-8'?>
<text title="SENTENCES.">
  <p>
    <s id="s-1" lang="ang">
      <w id="s-1-w-1" pos="V-" lemma="Ān" norm="An">Ān</w>
      <w id="s-1-w-2" pos="R-" lemma="on" norm="on">on</w>
      <w id="s-1-w-3" pos="NE" lemma="crist" norm="">-</w>
      <w id="s-1-w-4" pos="NB" lemma="ġinn" norm="ginn">ġinn</w>
      <w id="s-1-w-5" pos="V-" lemma="is" norm="is">is</w>
      <w id="s-1-w-6" pos="PY" lemma="eal" norm="ealra">ealra</w>
      <w id="s-1-w-7" pos="NB" lemma="þing" norm="thinga">þinga</w>
      <w id="s-1-w-8" pos="F-" lemma="," norm="">,</w>
      <w id="s-1-w-9" pos="PD" lemma="þæt" norm="thaet">þæt</w>
      <w id="s-1-w-10" pos="V-" lemma="is" norm="is">is</w>
      <w id="s-1-w-11" pos="NE" lemma="god" norm="God">God</w>
      <w id="s-1-w-12" pos="A-" lemma="æl" norm="ael">æl</w>
      <w id="s-1-w-13" pos="NE" lemma="crist" norm="">-</w>
      <w id="s-1-w-14" pos="PY" lemma="mihtig" norm="mihtig">mihtiġ</w>
      <w id="s-1-w-15" pos="NE" lemma="." norm="">.</w>
    </s>
```

Figure 3: The first sentence in the corpus

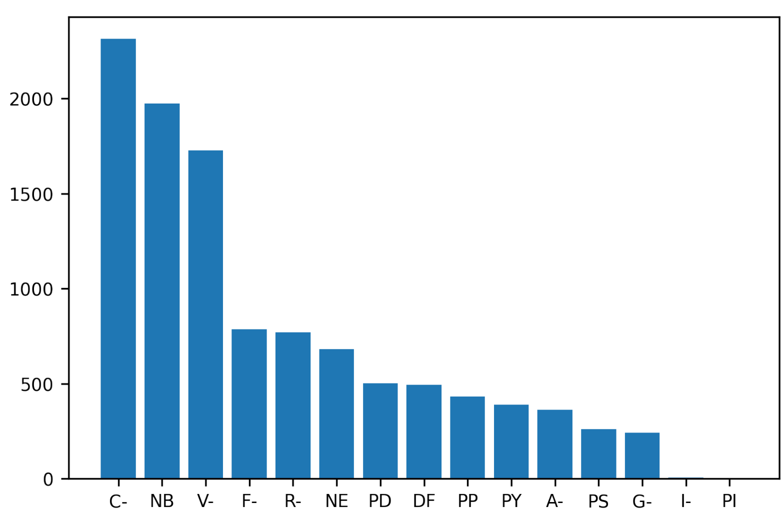


Figure 4: Distribution of PoS tags with the CRF tagger

Tag	Description
A-	Adjective
DF	Adverb
NB	Common noun
C-	Conjunction
DU	Interrogative adverb
F-	Foreign word
I-	Interjection
N-	Infinitive marker
PD	Deomnstrative pronoun
PI	Interrogative pronoun
PP	Personal pronoun
PS	Personal possessive pronoun
PX	Indefinitive pronoun
R-	Preposition
NE	Proper noun
PY	Quantifier
G-	Subjunction
V-	Verb

Table 1: List of PoS tags used in the corpus[7]

4.1.2 Word counts

The most common lemmas in the corpus are *and* 'and', *he* 'he', *þā* 'then', *on* 'on', *se* 'that, the, he', *þe* relative pronoun 'that, who, which' and *tō* 'to'. Table 2 lists the ten most common lemmas in the corpus. As can be seen, the lemmatizer does not perform well, since it treats *he* and its emphatic version *hē* as two separate words.

Zipf's law seems to apply to this corpus as well. As can be seen in figure 6, the frequency of words reduces exponentially; few words occur very often while most of the words occur rarely.

4.1.3 Keyword analysis with tf-idf

Tf-idf (token frequency-inverse document frequency) is a known algorithm for creating document vectors and/or extracting keywords from a document. It weighs the term frequency for each document with the inverse document frequency, such that words that occur often in a certain document but rarely in other documents get a higher score.

The script `tfidf.py` applies the tf-idf algorithm to the corpus. It first calculates the token frequency $\mathrm{tf}_{t,d} = \mathrm{count}(t, d)$ for each file, and then the inverse document frequency $\mathrm{idf}_t = log(\frac{N}{\mathrm{df}_t})$ where N is the total number of documents and df_t is the number of documents in which the term t occurs. Finally, it com-

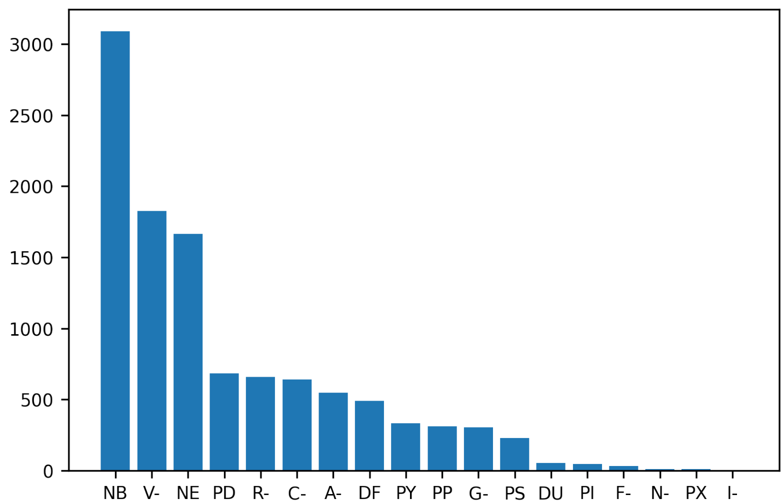


Figure 5: Distribution of PoS tags with the Perceptron tagger

	O.E.	Mn.E.	Counts
1	*and*	and	582
2	*he*	he	297
3	*þā*	then	212
4	*on*	on	203
5	*se*	that, the, he	176
6	*þe*	that, who, which	166
7	*tō*	to	160
8	*þǣm*	the, this (dative)	147
9	*þæt*	that	131
10	*hē*	he (emphatic)	128

Table 2: The top ten most common words in the corpus

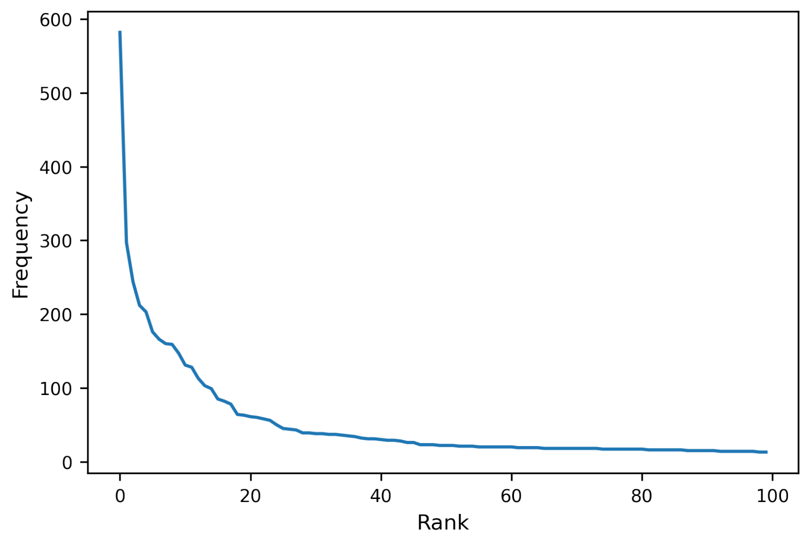


Figure 6: The frequencies of the 100 most common words in the corpus

putes the weighted values of the frequency counts with the inverse document frequency[11]:

$$w_{t,d} = \mathrm{tf}_{t,d} \times \mathrm{idf}_t$$

Taking a look at some examples of the weighted values of the files in the corpus, we see that this works well for keyword extraction. The top values for the third text are for instance *cwen* 'queen', *wisdom* 'wisdom', *Abrahām* 'Abraham', *Isaāc* 'Isaac' and *Daniēl* 'Daniel'. This is obviously a biblical text.

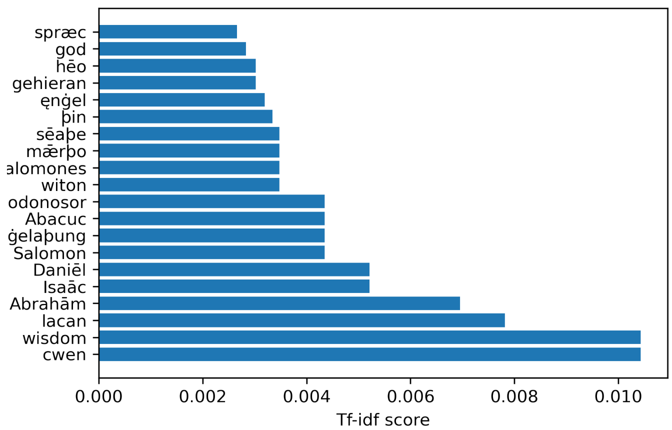
In the fifth text in the corpus, 'From the chronicle', the top weighted values are *seaxe* 'Saxons', *Brettas* 'the British (Celts)', *Hęnġest* 'Hengist'.[12] This chronicle about the invasion of the Angel-Saxons in Britain.

A File documentation

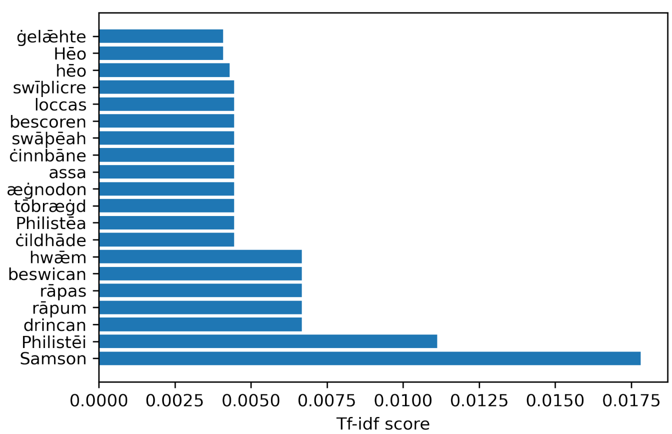
- converter.py - script for converting the obtained text file into one paragraph per line without hard wrap format
- normalizer.py - function for creating the normalized ASCII form
- stats.py - script for generating statistics (PoS tag and word counts)

[11] Jurafsky and Martin 2019, p. 105-106.

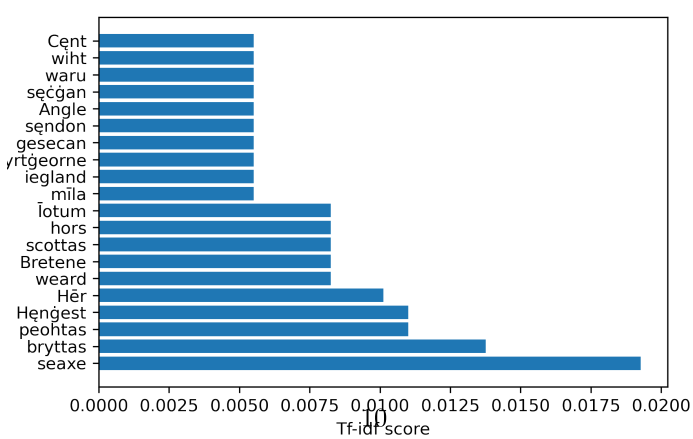
[12] Hengist is said to be one of the leaders of the Angels, Saxons and Jutes in their invasion in Britain.



(a) Text 3: Old Testament pieces



(b) Text 4: Samson



(c) Text 5: From the chronicle

Figure 7: Tf-idf graphs

- tfidf.py - script for generating tf-idf value for the XML corpus
- xml_generator - script for generating annotated XML files from the raw text files
  - texts\converted - the converted raw text files
  - texts\raw - the text files as obtained from the Gutenberg Project
  - texts\XML_crf - XML files annotated with the CRF tagger
  - texts\XML_perceptron - XML files annotated with the Perceptron tagger

B Software used

- NLTK: The Natural Language Toolkit, for word and sentence tokenizing
- CLTK: The Classical Language Toolkit 0.1.121, for Old English annotation, models trained with the ISWOC treebank.
- lxml for XML generation and parsing
- matplotlib[13] for plot generation
- LaTeX(XeLaTeX) for typesetting

List of Figures

List of Tables

[13]Hunter 2007.

[14]*Syntacticus: Development guide* n.d.

References

Bech, Kristin and Kristine Eide (2014). *The ISWOC corpus.* URL: `http://iswoc.github.io.`.

Hunter, J. D. (2007). "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3, pp. 90–95. DOI: `10.1109/MCSE.2007.55`.

Johnson, Kyle P. et al. (2014–2021). *CLTK: The Classical Language Toolkit.* URL: `https://github.com/cltk/cltk`.

Jurafsky, Dan and James H. Martin (2019). *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition, third edition draft.* 3rd ed. Pearson Prentice Hall.

Sweet, Henry (1905). *An Anglo-Saxon Primer.* 1st ed. Oxford: Oxford University Press. ISBN: 0-19-811178-9.

— (n.d.). *The Project Gutenberg eBook, Anglo-Saxon Primer, by Henry Sweet.* URL: `https://www.gutenberg.org/files/34316/34316-h/34316-h.htm`.

Syntacticus: Development guide (n.d.). URL: `http://dev.syntacticus.org/development-guide/#lemma-part-of-speech-and-morphology`.