The Making of the Royal Society Corpus

Jörg Knappen **Stefan Fischer** **Hannah Kermes** **Elke Teich**
Sprachwissenschaft und Sprachtechnologie
Universität des Saarlandes
{j.knappen, h.kermes, e.teich}@mx.uni-saarland.de
stefan.fischer@uni-saarland.de

Peter Fankhauser
Institut für Deutsche Sprache (IDS)
fankhauser@ids-manheim.de

Abstract

The Royal Society Corpus is a corpus of Early and Late modern English built in an agile process covering publications of the Royal Society of London from 1665 to 1869 (Kermes et al., 2016) with a size of approximately 30 million words. In this paper we will provide details on two aspects of the building process namely the mining of patterns for OCR correction and the improvement and evaluation of part-of-speech tagging.

1 Introduction

The Royal Society Corpus is built in an agile process (Cockburn, 2001; Voormann and Gut, 2008) aiming for continuous improvement from the OCR'ed original texts to the annotated corpus. In this work we elaborate on some of the details of the corpus processing including our methods of OCR pattern finding and part-of-speech tagging evaluation.

2 Improving OCR Quality

The quality of OCR for historical text is a long-standing issue (Alex et al., 2012) in corpus building. We employ a pattern based approach to OCR correction, using the stream editor *sed*. In accordance with agile principles, we build the corpus repeatedly from scratch using a build script and strict versioning (a new build number is assigned to each build).

2.1 An Initial Set of Patterns

As initial set of patterns we use the list of 50,000 patterns by Underwood and Auvil (2012) encoded as an sed script. The patterns are full words and mainly geared to correct predictable substitutions like *s* to *f*, *h* to *li*, or *e* to *c*. In a next step we eliminate all patterns that are not used at all in our corpus and patterns that result in overcorrection (this includes all patterns that convert a word-final *f* into an *s*). We also change a few patterns that transform to the wrong words in the RSC (see Table 1).

pattern	original	corrected
baving	baying	having
fhe	she	the
frem	fresh	from
11th	lith	11th
liind	hind	kind

Table 1: Corrected OCR patterns.

2.2 New Patterns from Word Embeddings

In order to find additional corpus specific OCR errors we use word embeddings. The basic idea behind this approach is that misspelled words have a very similar usage context as their correctly spelled counterparts. Using the structured skip-gram approach described in Ling et al. (2015) we compute word embeddings as a 100-dimensional representation of the usage contexts. Other than the original skip-gram approach introduced in Mikolov et al. (2013), the structured skip-gram approach takes word order in the usage context into account, and thus tends to compute similar embeddings for words with similar syntactic context. Using all words with a minimum frequency of 10 we compute embeddings for 56,000 different types coming from about 190,000 tokens. The word embeddings are L2-normalized and then grouped into 2,000 clusters using k-means clustering.

In Table 2 a few selected clusters are shown.

no.	words
2	the, thle, tile, ’the, tlhe, tie, tle, thie, ofa, ’of, tihe, tthe, ttle, .the, thte, thee, .of, ithe, of-the, th-e, onl, tothe, t-he, oni, andthe, othe, fthe, thlle, onthe, atthe, to-the, *of, sthe, ttlat
16	have, been, had, has, having, already, hath, hitherto, previously, formerly, heretofore, hlave, lhave, hlad, hlas, ihave, lhad, ving, lhas, harre, hiave, ’have, 11ave, liad, ’they, bave, hlaving
24	from, fiom, firom, friom, fiomn, fiorn, ’from, fromn, firomn, ftom, srom, fiomr

Table 2: Some example clusters.

Cluster no. 24 is a pure cluster consisting entirely of the word *from* and corrupted spellings of it. We add all those spellings to the OCR correction patterns. Clusters no. 2 and 16 demonstrate that clusters do not necessarily consist of misspelled variants only, thus they cannot be used as such without manual inspection. Altogether, we derive approximately 370 corpus specific patterns from the clusters. Cluster no. 2 also gives us the confidence to interpret *tile* as a corruption of *the* and not as the genuine word *tile*.

2.3 Beyond Words: Prefixes, Suffixes, and Substrings

Sorting the patterns alphabetically reveals a lot of common prefixes in the patterns. Going from full word patterns to prefix patterns does not only lower the number of patterns but also increases their coverage of inflected and derived forms. In a similar way, there are common patterns for derivational and inflectional suffixes, specially for common endings like *-ion* or *-ing*. We show some prefix and suffix patterns in Table 3.

There are also a few patterns that are applied everywhere. Those patterns are carefully inspected such that they do not apply to otherwise correct words. We show some examples of substring patterns in Table 4.

2.4 Removing Remains of Hyphenation

We also use a special set of pattern to correct remains of hyphenation. Most of the hyphenation occurring in the original texts was already undone. Typical remains of hyphenation include hyphenation over page breaks, or cases like *trans-. parent* where a spurious full stop is added after the hyphen. We mined for the most frequent cases and created special sed patterns to repair them.

affix	patterns
circum	circllm, circnm, circtlm, circuln, circuml, circunl, circurn, circutn
experim	experilm, experiln, experilu, experiml, experinl, experinm, experirn, experitn
sub	sllb, stlb
under	ilnder, ullder, utlder
ally	allv
ing	illg, ilng, inlg, irng, itlg, itng
ion	ioil, ioll, ionn, iorn, iotn
ment	meIlt, melit, mellt, merlt, metlt

Table 3: Some prefix and suffix patterns.

substring	patterns
qu	qll, qtl
spher	spllr
th	tlh, tlz, t}l, t}z
wh	vvh

Table 4: Some substring patterns.

2.5 Remaining Cases

In total, there are about 42,000 OCR corrections that are found by about 2,000 sed patterns. We show the five most frequent substitutions in Table 5.

However, not in all cases a word corrupted by OCR errors can be reconstructed reliably. We encountered cases like *llow* that can come from now or how, or *tne* that can come from me or the. In those cases we currently don’t apply an automated correction. Future builds of the corpus may contain some context sensitive repair in those cases.

frequency	wrong	corrected
1346	tlle	the
1214	ofthe	of the
1140	anid	and
1093	thle	the
1032	fiom	from

Table 5: Top 5 OCR corrections.

3 Normalization

Normalization is part of the annotation step and precedes part-of-speech tagging. We chose VARD (Baron and Rayson, 2008), which detects spelling variants in historical corpora and suggests modern alternatives. It is geared towards Early Modern English an can be used semi-automatically after training. To this end, we trained it on a manually normalized subset of the corpus. In total, VARD automatically replaced 0.31% of the words by their modern spelling. The percentage of normalized words decreases strongly in later time periods (see Table 6).

time period	normalized words
1650s	1.47%
1700s	0.97%
1750s	0.25%
1800s	0.08%
1850s	0.06%

Table 6: Effect of normalization across time.

4 Part-of-Speech Tagging

We use TreeTagger (Schmid, 1994; Schmid, 1995) with the default parameter file for tokenization, lemmatization and annotation of part-of-speech (POS) information in the corpus. For the time being, we did not adapt the tagger to the historical text material by training or any other adjustment.

For evaluation we created a gold standard on a sample of 56,432 words, which were drawn from 159 texts covering all time periods. The sample was manually tagged by two annotators, who achieve an inter-annotator agreement of $\kappa = 0.94$ (Cohen's kappa). Differences (3,011 words) were reconciled after discussion and resulted in a gold standard, which we use in the evaluation.

4.1 Annotation Quality

A classic quantitative evaluation shows that compared to the gold standard TreeTagger has an accuracy of 0.94 (per token) on the sample corpus. In order to better judge the annotation quality and the reliability of the tagger, we additionally perform a detailed qualitative analysis of tagging errors. The goal is to identify typical errors of the tagger, possible regularities and error directions.

4.2 Detailed Evaluation of Tagging Results

In a first step, we calculate the F-score for each part-of-speech tag separately. This allows us to identify problematic pos-tags. In a second step, we use confusion matrices of pos-tags from the gold standard and the respective pos-tags assigned by TreeTagger. This allows us to identify regularities and error directions. As we are interested in the errors with the largest impact and for better readability we do not include all pos-tags of the Penn Treebank tagset in the second step but exclude tags with an F-score $>= 0.99$ as well as rarely used tags. We also collapse some of the fine-grained distinctions of the tagset.

Figure 1 shows a confusion matrix with the correct pos-tags from the gold standard on the y-axis and the pos-tag assigned by TreeTagger on the x-axis. The matrix is normalized for pos-tag frequency and allows to observe possible regularities and directions in the tagging errors.

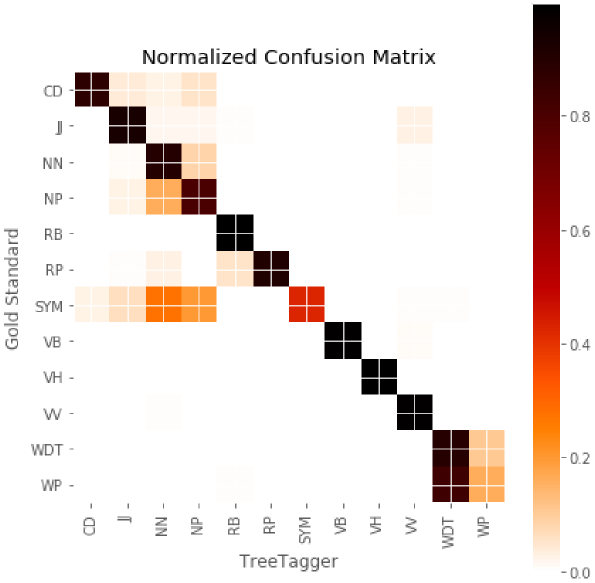

Figure 1: Normalized confusion matrix of POS annotation with the correct pos-tag on the y-axis, pos-tag assigned by TreeTagger on the x-axis.

From this we can draw the following observations. One major error source are symbols (SYM). Here we cannot really identify a direction of the errors. A closer look reveals that the pos-tag SYM is differently interpreted by the manual annotators than by TreeTagger. While the tagger assigns SYM only to single-character symbols, the annotators also tag longer words with SYM. Other error sources exhibit more obvious regularities and error directions. For example, TreeTagger often confuses common nouns (NN) with proper nouns (NP) and wh relative pronouns (WP) with wh relative determiners (WDT). In the latter case, *which*, e.g., is exclusively tagged as WDT. Although these error sources are unproblematic for a variety of linguistic annotations, they have a considerable impact on tagger performance.

The impact of the identified errors gets more obvious if we look at the confusion matrix with absolute frequencies of the pos-tags shown in Figure 2. As the figures are not normalized, only highly frequent observations are visible, and the shading is directly linked to the overall impact of the error. Thus, the error with the highest overall impact is the NN-NP error, followed by the WP-WDT and the NP-NN error. If we remove all noun related errors from the error list, tagging accuracy rises from 0.94 to 0.96.

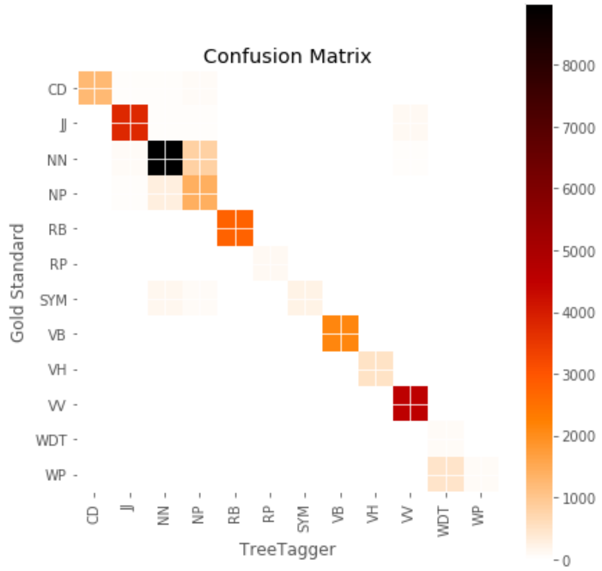

Figure 2: Confusion matrix of POS annotation with the correct pos-tag on the y-axis, pos-tag assigned by TreeTagger on the x-axis.

The NN-NP errors arise mainly from out-of-vocabulary words. While in contemporary English common nouns are always written in lower case, and capitalization indicates a proper noun, common nouns are still quite frequently capitalized in Late Modern English. Thus, a modern tagger has a strong tendency to tag capitalized words as proper nouns. We can also observe a decline of NN-NP errors over time in the RSC. Figure 3 shows the distribution of the ten most frequent tags across time. While most tags remain steady over time, the progression of NN and NP is remarkable. Their share is equal in the first two time periods (ca. 9%), then NN increases and NP decreases. Yet, the combined share of NN and NP remains the same (ca. 18%). We attribute this to the fact that in earlier time periods, capitalization of common nouns was still frequent, but decreases over time.

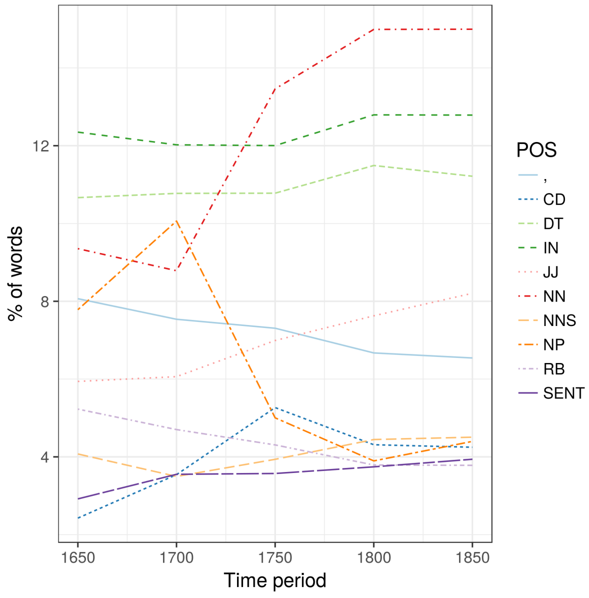

Figure 3: Most frequent POS tags across time.

4.3 Future Improvements

In order to tackle the identified typical errors, we opt for an improvement of the tagger lexicon, as we see a close relation to the major error sources. Thus, we extract all unknown words as well as all sentence internal capitalized words from the corpus. For the capitalized words, we construct lexicon entries (semi-)automatically using the tagger on the lower case version of the words. Besides, we manually construct lexical entries for frequent unknown words. Additionally, we extended the abbreviations lexicon of the tokenizer, in order to reduce segmentation errors due to unrecognized abbreviations. We extracted a list of candidate abbreviations from the corpus and checked them manually. As a result we added a list of 170 abbreviations to the tokenizer's list of abbreviations.

By the time of the workshop we will be able to present results of a new evaluation based on these improvements. Besides, we will also train the tagger on our data and compare the performance of both tagger versions.

5 Conclusion

We have presented an agile corpus building process to continuously improve the Royal Society Corpus. We have given details on our approach for OCR correction that may be helpful to other projects as well. We store all OCR corrections in a stream editor (sed) file that is applied to the corpus sources in each build with strict versioning. The agile approach extends to the stages of normalization and tagging where improvements are stored in parameter files for the tools we are using.

Both the general approach and some of the resources we created (like the patterns for OCR correction) can be applied to other corpus building projects.

The Royal Society Corpus (corpusBuild 2.0) has been made available for download and online query from the CLARIN-D centre at the Saarland University under the persistent identifier `http://hdl.handle.net/11858/00-246C-0000-0023-8D1C-0`. We also plan to release the OCR correction patterns in this context.

References

Bea Alex, Claire Grover, Ewan Klein, and Richard Tobin. 2012. Digitised historical text: Does it have to be mediOCRe? In *Proceedings of KONVENS 2012 (LThist 2012 workshop)*, pages 401–409, Vienna, Austria.

Alistair Baron and Paul Rayson. 2008. VARD 2: A tool for dealing with spelling variation in historical corpora. In *Proceedings of the Postgraduate Conference in Corpus Linguistics*, Birmingham, UK.

Alistair Cockburn. 2001. *Agile Software Development*. Addison-Wesley Professional, Boston, USA.

Hannah Kermes, Stefania Degaetano-Ortlieb, Ashraf Khamis, Jörg Knappen, and Elke Teich. 2016. The royal society corpus: From uncharted data to corpus. In *Proceedings of the LREC 2016*, Portorož, Slovenia, May 23-28.

Wang Ling, Chris Dyer, Alan Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of NAACL*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to german. In *Proceedings of the ACL SIGDAT-Workshop*.

Ted Underwood and Loretta Auvil. 2012. Basic OCR correction. `http://usesofscale.com/gritty-details/basic-ocr-correction/`.

Holger Voormann and Ulrike Gut. 2008. Agile corpus building. *Corpus Linguistics and Linguistic Theory*, 4(2):235–251.