Linguistic and Sociolinguistic Annotation of 17th Century Dutch Letters

Marijn Schraagen, Feike Dietz, Marjo van Koppen
Utrecht University, The Netherlands
{M.P.Schraagen, F.M.Dietz, J.M.vanKoppen}@uu.nl

Abstract

Developments in the Dutch language during the 17th century, part of the Early Modern period, form an active research topic in historical linguistics and literature. To enable automatic quantitative analysis, a corpus of letters by the 17th century Dutch author and politician P.C. Hooft is manually annotated with parts-of-speech, document segmentation and sociolinguistic metadata. The corpus is developed as part of the Nederlab online research portal, which is available through the CLARIN ERIC European research infrastructure. This paper discusses the design and evaluation of the annotation effort, as well as adding new annotations to an existing annotated corpus.

Keywords: Early Modern Dutch, POS tagging, sociolinguistic annotation, data integration

1. Introduction

In the late 16th and 17th century, the Dutch language was subject to a series of standardization and modernization developments, described in historical linguistics as the transition into (Early) Modern Dutch (Donaldson, 1983; Burke, 2005). These developments include changes in spelling and vocabulary, as well as syntactic changes regarding case marking and negation. The adoption of developments differs not only between authors, but also within the same author (van der Wal et al., 2012; Nobels and Rutten, 2014), cf. intra-speaker variation in speech corpora (Schilling-Estes, 2002; Szmrecsanyi, 2005) or intra-author variation in English literature (Leech, 1969; Busse, 2002). Variation is influenced by linguistic contexts as well as sociolinguistic factors (e.g., intended audience). To aid the study of language variation in this time period, a corpus is developed containing annotations on both the linguistic and sociolinguistic level. This corpus is intended to be used directly as linguistic research data, to study phenomena of interest both within the corpus itself as well as in comparison to external resources. Moreover, the manual annotation effort can be leveraged as training data for automatic methods in the field of historical linguistics.

2. Corpus Development

The corpus includes documents from the correspondence archive of the author and politician P.C. Hooft (1581-1647), which is available online from the Digital Library of Dutch Literature (Committee DBNL, 2015). Every document in this corpus corresponds to a single letter. Hooft is known to have been interested in studying the development of the Dutch language, and advancing this development in his own writing. The full correspondence consists of over 1300 letters with a total length of 300k tokens, from which 333 letters from the period 1600-1638 have been annotated (108k tokens in total). The boundary of 1638 is selected based on a shift in the use of negation in the work of Hooft, however the letters from this time period contain a number of other interesting linguistic phenomena as well.

The annotation task was performed by a pool of eight students with a background in linguistics and/or historical literature. Annotations consist of lemmas, part-of-speech tags (including various features for each tag) and sociolinguistic variables on document and person level. POS tagging makes use of the tagset described by Van Eynde et al. (2000) for contemporary Dutch, which is developed for the Spoken Dutch Corpus (CGN Consortium, 2003). For the current corpus several features have been added to accomodate historical linguistic phenomena, such as case marking and negation clitics. To increase annotation efficiency, the methodology was based on post-correction of tags generated by the Adelheid tagger for Middle Dutch (van Halteren and Rem, 2013). Middle Dutch shares a number of interesting phenomena with Early Modern Dutch (e.g., case marking, clitics, pronoun compounding) which are not found in Modern Dutch, therefore this tagger provided a useful starting point for manual annotation. However, major differences between the two historical language varieties exist as well, which necessitates a full check on all generated tags. Moreover, the manual tagging effort was used to extend the original tagset with additional features.

Document characteristics	
Category	business, personal
Type	regular, appendix
Goal	express thanks, compliment, excuse, ask a favour, ask information, ask advice, admonish, inform, remember, persuade, order, allow, invite
Topic	business, literature, domestic affairs, love, death, news, religion/ethics
Correspondent characteristics	
Group	name
Individual	name, birth/death date, gender, occupation, literary author, relation to P.C. Hooft
Letter segmentation	
Introductory greeting, opening (optional), narrative, closing (optional), final greeting	

Table 1: Sociolinguistic annotation set.

The sociolinguistic annotation set (provided in Table 1) consists of document characteristics, correspondent characteristics and text structure segmentation. The set of letter segments corresponds to the use in previous research, e.g.,

token	*translation*	*annotation 1*	*annotation 2*	*description*
fraejicheden	fine things	fraaiheid	fraaiigheid	lemma difference
gedicht	written poetry	N(nonnom,sg)	V(lex,pp)	past participle vs. noun
ijet	something	PRON(indef,3,sg,nonnom)	PRON(indef,nonnom)	missing features
kan	can	V(simple,pres,nonlex,sg,3)	V(simple,pres,nonlex,sg,1)	1/5 features different
etc	etc	SPEC(unclear)	SPEC(foreign)	ambiguous feature

Table 2: Tagging disagreement examples.

(Rutten and van der Wal, 2014, p. 86), whereas document categories are based on general rhetorical frameworks of letter writing (cf., for example, (Stowers, 1986, pp. 15-16)). The categories for correspondent characteristics are developed specifically for this corpus.

3. Inter-Annotator Agreement

For each annotator, a number of documents with a total of approximately 1000 words has been assigned to a second annotator as well. Using this data, inter-annotator agreement (IAA) can be computed on various aspects of annotation, i.e., token-based annotation of lemma, POS and features, document-based annotation of text segments and letter characteristics. For POS agreement a fine-grained measure is desired to differentiate various types of disagreement. For example, a token can be assigned different main tags or different features within the same main tag, feature categories can contain missing values instead of different values, various main tags have a different number of feature categories which influences the potential for disagreement, and some feature categories can be considered more salient than others, or more ambiguous. Some examples are provided in Table 2.

In related work, for various resources only a single IAA figure is reported (Voutilainen, 1999; Brants, 2000; Gimpel et al., 2011; Plank et al., 2014), or IAA is mentioned as desirable but not computed due to resource development constraints (Oostdijk et al., 2008; van Halteren and Rem, 2013). Widely used tagsets, such as the Penn Treebank tagset for English (Santorini, 1990) or the *STTS-small* tagset for German (Schiller et al., 1999), do not make an explicit distinction between main tags and features, and the amount of (implicit) features is usually small (e.g., the Penn tagset contains 4 noun features and 6 verb features, vs. 11 and 18 in the current tagset, respectively), which could explain the absence of feature agreement measurements. Indeed, for the Spoken Dutch Corpus (on which the tagset for the current corpus is based), two separate figures are reported for agreement on tags with features and agreement on main tags only (Zavrel and Daelemans, 1999). To extend this approach, in Figure 1 agreement measures are provided[1] for lemmas, full POS tags, main tags only, agreement on single features instead of feature sets, and agreement on feature categories where both annotators have provided a value (i.e., features in a category, such as *verb tense*, are discarded for a token if a value is missing for one of the annotators). In Table 3 the application of the measures to the examples in Table 2 is provided.

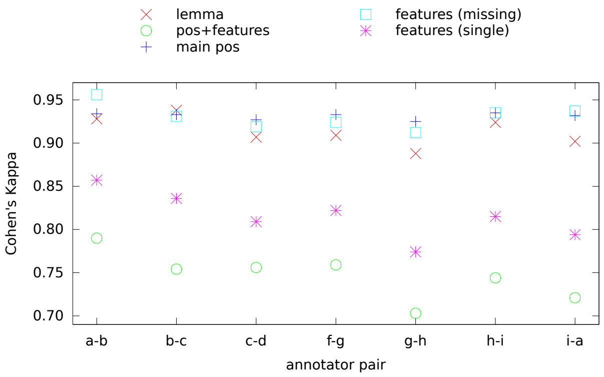

Figure 1: POS inter-annotator agreement.

token	pos +	main	features	features
	features	pos	(missing)	(single)
gedicht	0	0	0	0
ijet	0	1	1	0.5
kan	0	1	0.8	0.8
etc	0	1	0	0

Table 3: Application of agreement measures to examples from Table 2.

The figure shows that agreement on full tags (i.e., the measure generally reported for this type of annotation task) is relatively low (~ 0.75). When features are not taken into account (i.e., agreement on lemma or on main POS), agreement is higher (> 0.9). Similarly, agreement is high when features from a particular feature category are counted only if both annotators provided a value for this feature category (denoted by *features (missing)* in Figure 1). The first condition (full agreement) may be considered too strict, while lemma, main POS, and missing features[2] are arguably too permissive. A more balanced measure therefore may be the agreement on individual features (~ 0.81).

For document-based annotation the amount of data is much smaller compared to POS tagging (24 letters in total), therefore counts are computed instead of statistical measures, as

[1] Originally, the tagging task was assigned to a pool of nine annotators, each of which was assigned an individual set of documents as well as a selection of documents from the set of the previous annotator for the measurement of inter-annotator agreement. However, one of the annotators (annotator *e*) left the pool, which means that the agreement for the pairs *d*–*e* and *e*–*f* could not be measured, resulting in a total of seven pairs in Figure 1.

[2] This condition applies to 12% of all feature agreement measurements, averaged over annotator pairs. Note that the issue could have been avoided by enforcing annotation guidelines, either in the annotation software or through more explicit instruction and monitoring.

annotation feature	*agreement*	*proportion*
business/personal	21	0.875
regular/appendix	16.5	0.686
letter goal	12	0.500
letter topic	18	0.750
greeting	61	0.241 (0.492)
opening	175	0.449 (0.900)
narrative	6166	0.948 (0.907)
closing	370	0.754 (0.750)
final greeting	311	0.920 (0.926)
segment totals	7083	0.888 (0.795)

Table 4: Agreement on sociolinguistic annotation.

shown in Table 4. In the table, for the document-level annotations the numbers represent the amount of documents for which annotators agree (out of 24). The annotation *unknown* counts as 0.5 agreement. For the text segments the table shows the number of common tokens, proportional to total (union) segment length and (in parentheses) averaged proportion per document. Agreement on letter goal is low, which may indicate that specific letter categories need to be merged. Agreement on segmentation is not unreasonable, especially when taken into account that some differences can be easily corrected (e.g., some annotators include the address line in the *greeting* segment, while others only include the actual greeting). The table shows two different segmentation agreement measurements: the number of words in agreement proportional to the total number of words in the segment in all documents (where the segment length for a document is counted as the length of the union of the segments for both annotators), and the average of all proportions in individual documents. The first measure has the advantage that each token is given equal weight, whereas the second measure gives each document equal weight. The second measure may be more informative, given the large differences in segment length across documents. Additionally, the second measure allows missing optional segments (*opening* and *closing*) to be counted as full agreement, whereas for the first measure missing segments are discarded.

The correspondent-based annotation was performed by a small number of annotators only, therefore agreement on these annotations has been omitted.

3.1. Sociolinguistic annotation schema

As noted in Section 2, the sociolinguistic annotation schema is based on general rhetorical frameworks of letter writing. Descriptions of these frameworks were available in the Netherlands in the 17th century. Therefore, it is likely that P.C. Hooft was familiar with the division into rhetorical categories and, furthermore, that his knowledge of rhetorics influenced his own letter writing. Therefore, the established categories were selected as a framework for sociolinguistic tagging of the current corpus. However, given the fact that agreement on letter goal and topic is low, the framework may need to be revised. In order to provide recommendations for improvement of the category division, a qualitative assessment of the categories has been performed on a sample of 100 letters from the corpus.

Original framework	
Goal	express thanks, compliment, excuse, ask a favour, ask information, ask advice, admonish, inform, remember, persuade, order, allow, invite
Topic	business, literature, domestic affairs, love, death, news, religion/ethics
Revised framework	
Goal	prompt for action, honour, help, inform, keeping contact, ask for reply
Topic	political work, literary work, current events, social circle

Table 5: Proposed categories for sociolinguistic tagging.

A first observation is that the number of categories for goal and topic (see Table 1) is large and the boundaries of the categories are not always clear. Therefore a new division is proposed with a smaller number of categories, where several closely related categories have been grouped together. The category names have been adjusted accordingly to clarify the scope of each category (see Table 5).

A second observation is that, even with the simplified category division, several letters can be categorized with two distinct goals. As an example, consider a letter dated October 6th, 1631. This letter is written following a decision of the State Counsil to modify tax law. As administrator of the region, Hooft was responsible for the implementation of this law. In the letter Hooft informs the administrative college about the new law. More importantly, however, in the letter the members of the college are ordered to take an official oath to enforce this law. The main goal of the letter can therefore be classified as *prompt for action*, with the secondary goal *inform*. However, in the sample of the qualitative assessment, only 11 out of 100 letters have been classified as having a secondary goal.

A further improvement in the consistency of sociolinguistic tagging can be obtained by providing more explicit tagging instructions. This is particularly useful for secondary letter goals, which should be annotated only if the secondary goal can be considered an important part of the letter, instead of being embedded as a courtesy phrase or a small digression.

4. Data Integration

To increase availability and accessibility of the annotations, the corpus has been developed as part of the Nederlab online research portal (Brugman et al., 2016). Nederlab offers a query and visualization interface for full text search in the Digital Library of Dutch Literature using Corpus Query Processor syntax (Christ, 1994). Nederlab is included in the CLARIN ERIC European research infrastructure (Hinrichs and Krauwer, 2014), which offers access through a majority of universities and research institutes in Europe.

All data in the Nederlab research infrastructure has been annotated automatically using the Frog NLP suite (van den Bosch et al., 2007) with lemma, POS and named entity information. The current annotations are used as a pilot for the aim of Nederlab to facilitate incorporation of user-generated data. This type of annotation is expected to be

type	*different*	*equal*	*ratio*
lemma			
manual vs. default	27,179	70,560	0.28
manual vs. modernized	18,052	53,648	0.25
main POS	23,365	82,830	0.22
features (all)	140,888	67,273	0.68
features (missing)	46,725	67,273	0.41

Table 6: Differences between automatic and manual annotation.

fragmented and the quality, while possibly rather high, may be unknown. To allow researchers using the Nederlab portal to differentiate between user-generated information and full coverage core annotations with known performance characteristics, the additional annotations are added as explicity marked alternatives.

Merging new annotations with existing data needs to account for possible differences in tokenization, resulting from preprocessing, tagger tokenization algorithms, and manual tokenization decisions made by annotators. To address this issue, the annotation data is aligned on token level, and multiple tokenization layers are represented in the annotation data model.

The merging process allows for measurements on the differences between automatic and manual annotations for this data, which are provided in Table 6. The table shows that the differences are substantial for all categories. Note that the measurements include an automatic modernization layer for lemmas, implemented using look-up on an authoritative source (de Vries and te Winkel, 1998). The quality of these modernized lemmas is high, however the coverage of this source is not fully complete. For measurements on features, note that manual annotations contain additional feature categories and values for which no automatic alternative is available. In the last row these features have been disregarded, providing a measurement of different feature values only.

Regarding POS-tagging the manual annotation has the additional advantage of providing a clean separation of the main text and additional document elements as compared to automatic tagging. For example, the source text contains modern editorial notes, figure captions, and various layout markers such as page or line numbers. In the manual annotation these elements have been identified, while during automatic annotation these elements have been considered as regular text.

4.1. Alignment algorithm

Token alignment is represented as a one-to-one greedy brute force algorithm, which is sufficient for this task given the serial nature of this specific alignment problem. After alignment, gaps are resolved if possible (or confirmed otherwise) in order to merge the annotation layers.

An outline of the algorithm is presented in Algorithm 1. First, all *anchor tokens*, defined as tokens with a frequency of 1 in both documents (case sensitive), are fixed in the alignment (line 4). The anchor tokens divide the document into pairs of small subsequences (5-6), which are aligned individually. An exception to this heuristic applies when

Algorithm 1: Alignment between documents.

```
   Data: two tokenized documents A, B
   Result: token-level alignment
 1 foreach token in document A do
 2     if token has frequency 1 then
 3         find this token in document B;
 4         if token has frequency 1 then
 5             subsequence $S_A$ = [previous aligned
                 token in A..current token in A];
 6             subsequence $S_B$ = [previous aligned
                 token in B..current token in B];
 7             if length difference($S_A$, $S_B$) $\leq L$ then
 8                 AlignSubsequences($S_A$,$S_B$);
 9                 ResolveGaps($S_A \leftrightarrow S_B$)
10             end
11         end
12     end
13 end
14 function AlignSubsequences($X$,$Y$)
15     currentAlign $\leftarrow X_i = Y_i$ for all tokens $i$;
16     $\Delta_{\text{current}} \leftarrow \sum_i \texttt{edit}(X_i, Y_i)$;
17     $\Delta_{\text{prune}} \leftarrow \texttt{edit}(\texttt{string}(X), \texttt{string}(Y))$;
18     while $\Delta_{\text{prune}} < \Delta_{\text{current}}$ do
19         for $j \leftarrow i$ to final token in $Y$ do
20             introduce a gap for $Y_{j-1}$ (if $j > i$);
21             align $X_i$ with $Y_j$;
22             if partial distance $\leq \Delta_{\text{prune}}$ then
23                 recursively align $X_{i+1}, Y_{j+1}$;
24             end
25         end
26         introduce a gap for $X_i$;
27         if partial distance $\leq \Delta_{\text{prune}}$ then
28             recursively align $X_{i+1}, Y_i$;
29         end
30         if at end of either sequence then
31             if final distance $\leq \Delta_{\text{current}}$ then
32                 currentAlign $\leftarrow$ final alignment;
33                 $\Delta_{\text{current}} \leftarrow$ final distance;
34             end
35         end
36         $\Delta_{\text{prune}} \leftarrow \Delta_{\text{prune}} + 1$;
37     end
38     return currentAlign;
39 end
40 function ResolveGaps(alignment $X \leftrightarrow Y$)
41     foreach token $X_i$ aligned to a gap do
42         get closest token $X_{i-k}$ aligned to $Y_j$;
43         if concat($X_{i-k} \ldots X_i$) is suffix of $Y_j$ then
44             store alignment $[X_{i-k} \ldots X_i] \leftrightarrow Y_j$;
45         end
46         get closest token $X_{i+l}$ aligned to $Y_m$;
47         if concat($X_i \ldots X_{i+l}$) is prefix of $Y_m$ then
48             store alignment $[X_i \ldots X_{i+l}] \leftrightarrow Y_m$;
49         end
50     end
51 end
```

the difference in length of the subsequence in a pair exceeds a threshold (7), in which case the anchor token is discarded and the subsequences are extended to the next anchor token pair. This happens occasionally in case of tokenization differences or tokens appearing outside of the main text (e.g., in a footnote). This is illustrated in Figure 2.1, where the position of the anchor token *den* in the two documents is far apart. Indeed these are different tokens, as the token in Text A is produced by end of line hyphenation, while the token in Text B is produced by a split of the token *aenden* into *aen* (*to*) and *den* (dative *the*). The other anchor tokens are retained, producing four subsequences for this example.

The subsequence pairs are assigned a trivial alignment (illustrated in Figure 2.2) where the tokens of each subsequence are aligned in order (15), and the total edit distance of this alignment is computed (16), used as a stop condition for expanding the search space. The search uses a pruning strategy for the alignment distance, with the maximum distance initialized as the edit distance of the full phrase (discarding token boundaries, line 17), and iteratively increased (36) until either an alignment is found (31) or the stop condition is reached (in which case the trivial alignment is returned). The search is executed recursively in three parallel directions (illustrated in Figure 2.3): align the current two tokens (20), introduce a gap by skipping a number of tokens in one document (19-25), and introduce a gap by skipping a number of tokens in the other document (26-29). Note that in Algorithm 1 the first direction is implemented as a special case of the second direction (i.e., skipping 0 tokens).

Subsequence alignments resulting from this procedure may contain gaps. The next stage of the algorithm attempts to resolve these gaps (illustrated in Figure 2.4) by concatenating gap-aligned tokens to the surrounding tokens (42, 46). If the combined tokens match with the align target of a constituent token, the combination is retained (44, 48), and otherwise the gap is confirmed. Finally, the alignment (including combinations and remaining gaps) is added to the XML representation of the automatic annotation output. A slightly simplified XML serialization example, including token alignment, is presented in Figure 3. In this example the automatically assigned lemma *salmon*, correct in modern usage, is manually retagged using historically correct tokens *shall* and *they*.

A custom tool was developed to facilitate the annotation process for the current corpus. This tool provides a web interface with document navigation based on the specific organization of the current corpus, integration with the Adelheid tagger, tokenization adjustments, propagation of manual corrections, integration of sociolinguistic annotation, and alignment of new and existing annotations. Integration of this interface with the Nederlab research infrastructure is currently under investigation.

5. Application of the Corpus

As an example application, the differences between main clauses and subordinate clauses have been analyzed regarding occurrences of bipartite negation. This type of negation is composed of a negative token (such as *not, never, nonetheless*), complemented with the negation clitic *en* (cf. *ne ... pas* in contemporary French). During the timeframe

1. **Text A** Aenden$_1$ Advocaet$_1$ | van$_{17}$ Hollandt$_1$ [...] verschej-$_1$ den$_1$ onwaerdicheden$_1$| zijn$_4$ toegedreven$_1$
Text B Aen$_1$ den$_1$ Advocaet$_1$ | van$_9$ Hollandt$_1$ [...] verschejden$_1$ onwaerdicheden$_1$ | zijn$_3$ toegedreven$_1$

2.

Aenden	Advocaet	$\emptyset$	Δ align $= 17$
Aen	den	Advocaet	$\Delta_{\text{prune}} = 1$

3.

Aenden Aen			$\Delta = 3 \not\leq 1$, prune.
$\emptyset$ Aen	Aenden den		$\Delta = 3 \not\leq 1$, prune.
Aenden $\emptyset$	Advocaet Aen		$\Delta = 12 \not\leq 1$, prune. set $\Delta_{\text{prune}} = 2$
$\vdots$			set $\Delta_{\text{prune}} = 6$
Aenden Aen			$\Delta = 3 \leq 6$, continue.
Aenden *Aen*	Advocaet den		$\Delta = 9 \not\leq 6$, prune.
Aenden *Aen*	$\emptyset$ den		$\Delta = 6 \leq 6$, continue.
Aenden *Aen*	$\emptyset$ *den*	Advocaet Advocaet	$\Delta = 6 \leq 6$, keep.

4.

Aenden Aen	$\emptyset$ den	Advocaet Advocaet	*original*
Aenden Aen+den		Advocaet Advocaet	*concat left* *keep*
Aenden Aen		Advocaet den+Advocaet	*concat right* *discard*

Figure 2: Alignment algorithm examples. 1: Anchor nodes and subsequence boundaries. 2: Trivial alignment. 3: Parallel recursive alignment with iterative pruning, showing the final search path *no gap, gap bottom, no gap*. 4: Resolving gaps by attempting concatenation in both directions.

```
<w xml:id="p.3.s.15.w.30" class="WORD">
  <t>zalmen</t>
  <!-- salmon -->
  <lemma class="zalm"/>
  <pos class="N(common,pl)" head="N">
    <feat class="common" subset="ntype"/>
    <feat class="mv" subset="number"/>
  </pos>
  <!-- shall -->
  <t textclass="gustave-cb">zal</t>
  <lemma class="zullen" textclass="gustave-cb"/>
  <pos class="V(fin,+nonlex,sg,3,pres)" head="V">
    <feat class="fin" subset="vtype"/>
    <feat class="+nonlex" subset="+lexical"/>
    <feat class="sg" subset="number"/>
    <feat class="3" subset="person"/>
    <feat class="pres" subset="tense"/>
  </pos>
  <!-- they -->
  <t_2 textclass="gustave-cb">men</t_2>
  <lemma_2 class="men" textclass="gustave-cb"/>
  <pos_2 class="PRON(indef,+nom)" head="PRON">
    <feat class="indef" subset="prtype"/>
    <feat class="+nom" subset="case"/>
  </pos_2>
</w>
```

Figure 3: Token alignment in XML serialization.

of the corpus the bipartite negation construct was in decline, before it disappeared from Dutch completely in later centuries. The Dutch language exhibits different word order in main clauses and subordinate clauses, which also affects the use of bipartite negation. Specifically, the verb in a main clause is positioned between the clitic and the negative (*hij en is niet*, English *he [clitic] is not*), while in a subordinate clause the verb is positioned at the end (*dat hij niet en is*, English *that he not [clitic] is*, meaning *that he is not*). It is hypothesized (Hoeksema, 2014) that the adjacent combination of the clitic and the negative in the subordinate clause became idiomatic, which has lead to a higher frequency of occurrence and a slower decline than the main clause counterpart which does not have adjacent tokens. This process is assumed to be most visible in the canonical form using the adverb *not*.

A quantitative analysis on the current corpus shows that the relative frequency of bipartite negation in subordinate clauses (∼ 4%) is higher than for main clauses (∼ 3%). Moreover, the frequency of *en niet* in subordinate clauses (∼ 78% of all bipartite negations in subordinate clauses) is higher than in main clauses (∼ 68%). From the *en niet* occurrences in subordinate clauses a majority (∼ 55%) consists of directly adjacent tokens. Therefore, this preliminary quantitative analysis is consistent with the idiom hypothesis, although other hypotheses might explain the observations equally well.

Note that the identification of subordinate clauses and bipartite negation within a clause is a non-trivial problem given a corpus with part-of-speech tags only. Therefore, the percentages mentioned above are approximate and the observations need to be confirmed in future research. However, this example does show the potential of the current corpus to collect useful research examples in a fast and automatic way.

Other possible applications include the use of the annotated data for classification algorithms (e.g., to predict a topic category given a letter) or a part-of-speech tagger for historical text. In general, for historical (socio)linguistics the amount of annotated data is limited, which impacts the application of corpus linguistic methods and natural language processing in general. The current corpus aims to improve this situation, within the domain of 17th century Dutch, but also within digital humanities as a whole.

6. Bibliographical References

van den Bosch, A., Busser, B., Canisius, S., and Daelemans, W. (2007). An efficient memory-based morphosyntactic tagger and parser for Dutch. In *Selected Papers of the 17th Computational Linguistics in the Netherlands Meeting*, pages 99–114. Netherlands Graduate School of Linguistics.

Brants, T. (2000). Inter-annotator agreement for a German newspaper corpus. In *Proceedings of LREC 2000*.

Brugman, H., Reynaert, M., van der Sijs, N., van Stipriaan, R., Tjong Kim Sang, E., and van den Bosch, A. (2016). Nederlab: Towards a single portal and research environment for diachronic Dutch text corpora. In *Proceedings of LREC 2016*, pages 1277–1281.

Burke, P. (2005). *Towards a Social History of Early Modern Dutch*. Amsterdam University Press.

Busse, U. (2002). *Linguistic Variation in the Shakespeare Corpus*. John Benjamins Publishing Company.

Christ, O. (1994). A modular and flexible architecture for an integrated corpus query system. In *Proceedings of the 3rd Conference on Computational Lexicography and Text Research*, pages 22–32.

Donaldson, B. (1983). *Dutch. A linguistic history of Holland and Belgium*. Martinus Nijhoff.

Van Eynde, F., Zavrel, J., and Daelemans, W. (2000). Part of speech tagging and lemmatisation for the spoken dutch corpus. In *Proceedings of LREC 2000*, pages 1427–1434.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. (2011). Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings ACL 2011*, pages 42–47. ACL.

van Halteren, H. and Rem, M. (2013). Dealing with orthographic variation in a tagger-lemmatizer for fourteenth century Dutch charters. *Language Resources and Evaluation*, 47(4):1233–1259.

Hinrichs, E. and Krauwer, S. (2014). The CLARIN research infrastructure: Resources and tools for eHumanities scholars. In *Proceedings of LREC 2014*, pages 1525–1531.

Hoeksema, J. (2014). The Middle Dutch negative clitic: Status, position and disappearance. *Lingua*, 147:50–68.

Leech, G. (1969). *A Linguistic Guide to English Poetry*. Pearson Education Limited. Edition: Routledge, 2013.

Nobels, J. and Rutten, G. (2014). Language norms and language use in seventeenth-century Dutch: negation and the genitive. In Gijsbert Rutten, editor, *Norms and usage in language history, 1600-1900. A sociolinguistic and comparative perspective.*, pages 21–48. John Benjamins Publishing Company.

Oostdijk, N., Reynaert, M., Monachesi, P., van Noord, G., Ordelman, R., Schuurman, I., and van de Ghinste, V. (2008). From D-Coi to SoNaR: A reference corpus for Dutch. In *Proceedings of LREC 2008*.

Plank, B., Hovy, D., and Søgaard, A. (2014). Learning part-of-speech taggers with inter-annotator agreement loss. In *Proceedings of EACL 2014*, pages 742–751. ACL.

Rutten, G. and van der Wal, M. (2014). *Letters as Loot: A sociolinguistic approach to seventeenth- and eighteenth-century Dutch*, volume 2 of *Advances in Historical Sociolinguistics*. John Benjamins Publishing Company.

Santorini, B. (1990). Part-of-speech tagging guidelines for the Penn treebank project. Technical Report MS-CIS-90-47, University of Pennsylvania. 3rd Revision.

Schiller, A., Teufel, S., Stöckert, C., and Thielen, C. (1999). Guidelines für das Tagging deutscher Textcorpora mit STTS (Kleines und großes Tagset). Technical report, Universität Stuttgart.

Schilling-Estes, N. (2002). Investigating stylistic variation. In J.K. Chambers, et al., editors, *The Handbook of Language Variation and Change*, pages 375–401. Blackwell.

Stowers, S. (1986). *Letter Writing in Greco-Roman Antiquity*. The Westminster Press.

Szmrecsanyi, B. (2005). Language users as creatures of habit: a corpus-based analysis of persistence in spoken English. *Corpus Linguistics and Linguistic Theory*, 1(1):113–149.

Voutilainen, A. (1999). An experiment on the upper bound of interjudge agreement: the case of tagging. In *Proceedings of EACL '99*, pages 204–208.

van der Wal, M., Rutten, G., and Simons, T. (2012). Letters as loot: Confiscated letters filling major gaps in the history of Dutch. In Marina Dossena et al., editors, *Letter Writing in Late Modern Europe*, pages 139–161. John Benjamins Publishing Company.

Zavrel, J. and Daelemans, W. (1999). Evaluatie van part-of-speech taggers voor het Corpus Gesproken Nederlands. Technical report, Universiteit Antwerpen.

7. Language Resource References

CGN Consortium. (2003). *Spoken Dutch Corpus*. Dutch Language Institute.

Committee DBNL. (2015). *Digital Library of Dutch Literature*. National Library of the Netherlands, `www.dbnl.org`, in Dutch.

de Vries, Matthias and te Winkel, Lammert. (1998). *Woordenboek der Nederlandsche Taal*. Dutch Language Institute, `gtb.inl.nl`, in Dutch.