

**Tagging and Lemmatization Manual
for the Corpus van Reenen - Mulder
and the Adelheid 1.0 Tagger-Lemmatizer**

Margit Rem

Hans van Halteren

Introduction

This manual describes the tagging and lemmatization that is used in two resources relating to historical Dutch text:

- The Corpus van Reenen – Mulder
- The Adelheid Tagger-Lemmatizer

The former was created in the 1980's at the Vrije Universiteit Amsterdam on the initiative of Piet van Reenen and Maaïke Mulder. An extensive description of the corpus can be found in (Rem, 2003; p. 21-38). The CRM was annotated semi-automatically and corrected various times. The annotation with word class tags was done originally with numerical codes, but these were recently transformed to a nowadays more accepted mnemonic form. In this manual we will often also include the numerical version of the tags for those of us who, because of ample experience with the CRM, are more used to this form.

Adelheid was created in the last few years by Hans van Halteren en Margit Rem. An introductory description can be found in (van Halteren and Rem, to be published). As the system makes heavy use of the CRM as training material, its output also takes the same form.

In this manual, we will first describe the general principles that were adhered to in annotation, after which we describe each word class by itself. We conclude by giving a list of all individual tags in the tag set and compound tags that can be observed in CRM.

Acknowledgements

We could not have done this work if the creators of the CRM had not done theirs, so we are greatly indebted to Piet van Reenen, Maaïke Mulder and all others who, through the years, have made contributions to this corpus.

As to the work on this manual, we would like to give special mention to Janneke Diepenveen whose work performed on the KANTL corpus summarized in her 2004 report has been of great value.

General guidelines

Within the annotation process we can distinguish three types of decision

- What are the tokens and sentence boundaries (tokenization)?
- What lemma should be assigned to each token (lemmatization)?
- What tag should be assigned to each token (tagging)?

In this section, we will describe in general what principles underlie these decisions, after which we will briefly remark on the annotation quality that should be expected from the Adelheid system.

In addition, we describe the treatment of discontinuous items in the CRM.

Tokenization: Sentence separation and punctuation

In this time period, standardized conventions for punctuation were mostly absent, which results in the fact that sentences often do not end in a full stop. We therefore do not add any sentence boundaries, but merely mark the punctuation that is present.

Tokenization: Token separation and layout

Compounding or separating words similarly depended on the choices of a writer, since no broad conventions were established or maintained in older forms of Dutch. We attempt to impose some order on this by (partially) linking what we observe in the usage in modern Dutch. This gives rise to three situations

- We respect the tokenization that is observed in the manuscript.
- We separate observed tokens into two or more annotated tokens.
- We merge one or more observed tokens in a single annotated token.

In the following, we describe each of these cases.

Original form respected: Compounding. Documents in the corpus show compound words with two or more elements that have to be written separately in Modern Dutch. The form *estewetene*, for example, is built up from *es + te + wetene* (**zijn + te + weten**, e.g. are + to + known). In many cases Adelheid respects these compound constructions but assigns a lemma and a morphological tag to every word that is recognizable in Modern Dutch. There are, however, also cases where the orthographic tokens are split into two or more parts, see below.

Original form respected: Clitic forms. The texts in this corpus also contain clitic forms, understood here as words that are written conjointly and in which phonemic loss or phonemic assimilation has occurred. The corpus contains examples such as *uten* = **uit + de** (out (of) + the) or *tenen* = **te + enen** (to + one). These forms are similarly left intact although they are assigned a double main word entry and a double morphological tag (cf. *jnt* = **in+dat** (in+that); Adp()+Art(def) (= adposition + article definite)).

Original forms separated. In some cases, orthographic tokens have been split into two or more parts. There are two main reasons for such treatment:

- One of the parts also has a relation with another word in the sentence (see Treatment of discontinuous forms below). E.g. in (...) *diere schuldech op was (...)* (who was guilty of this), *re* is split off so that it can be marked as belonging together with *op*.
- There are several very frequent token combinations which occur in both compound and separate forms in the corpus. We have decided to treat these identically throughout. Examples are numbers and preposition-article combinations. E.g. *sessendertech* (thirty six) has been split into *sess en dertech* (six and thirty) and *inde* (in the) has been split into *in de* (in the).

Note that such splitting is never used if the compound form shows phonemic loss or phonemic assimilation.

Original forms merged: Compounding elements. There are also elements that are written separately in these older texts but which should rather be compounded in Modern Dutch. It is attempted to connect these words using a tilde (~) and to assign them a single lemma within Adelheid. When for example the form *toe comende* (to + coming) occurs in a particular text, the element *toe* (to) is added to *comende* (coming) = *toe~comende*.

Treatment of discontinuous forms

Both in Modern Dutch as well as in older versions of the language there are grammatical constructions in which certain compound words do not appear next to each other. It is not possible to straightforwardly construct such cases with a tilde, for example when using a pronominal adverb: *diere deel an hebben* (*re* = **er** (there) + *an* = **aan** (to)); comprising the pronominal adverb *eraan* (to it, thereto)).

In the annotation, such situations are treated in two different tiers. Let us take an example represented in multi-column format:

die	die	die (that)	Pron(rel,forme)	-
daer	daer	daer (there)	PronAdv(dem)	ba4
metten	metten	met+de (with+the)	Adp()+Art(def,formn)	-
rechte	rechte	recht (right)	N(sing,forme)	-
toe	toe	toe (to)	PronAdv(pre)	ea0
horen	horen	horen (belong)	V(fin,pres,lex,formn)	-

The words *daer* and *toe* together form a pronominal adverb *daertoe*. In the tag tier this is indicated by tagging *toe* with PronAdv(pre), indicating that it is part of a separated PronAdv. In the fifth tier, it is shown which tokens should be linked together. The tag *ba4* in the second line indicates that there is another word (starting with the tag *ea0*) following the fourth symbol of the word *daer*. Together, these words comprise a compound. Such tags are constructed as follows:

b = first element (i.e.: the first element that has to be found within the text);

e = final element (i.e.: the second element that has to be found within the text);

An *a* has to be connected to another *a*. The current example contains only an *a*, but in case there are more discontinuous elements within the same context tags such as *bb* and *eb* can be used.

Numerals [0-9] signify the position of the symbols that are to be connected.

These tags are also used to denote parts of a separable (compound) verb, for example:

droeghen	droeghen	dragen	V(fin,pres,lex,formn)	ba0
----------	----------	---------------	-----------------------	-----

ouer~ene

ouerene

overeen

V(prtcl)

ea7

This yields the separable verb *overeendragen* (agree (on)). In this case, the secondary word with tag a has to be placed before to the first symbol of the primary word with tag a.

Although this annotation is present in the CRM, the Adelheid system does not attempt to reproduce this. It only provides the tags mentioned but not the linking elements.

Lemmatization

Spelling conventions were also not standardized in the Old Dutch, Middle Dutch and Early Modern Dutch periods. Nearly every composer had an own way of writing and spelling. The contemporary word *sint* (saint), for example, occurs in various forms in the 14th century, such as *sinte*, *sente*, *sante*, *sunte*, *seinte*, etc. Within Adelheid it has been attempted to group these diverse forms under one entry where possible (see below).

Various phonemic and/or spelling variations are categorized under one lemma. For example, forms such as *ute*, *vte*, *uut*, *vvt*, *uit*, *vit*, etc. are all grouped under the lemma **uit** (out). Within these lemmas there is abstraction from flexion phenomena. Assigning of lemmas is fairly straightforward:

1. Where possible a Modern Dutch form is assigned (*si* = **zij** (she), *stede* = **stad** (city)).
2. When there is no Modern Dutch form readily available, we chose to pick a Modern Dutch form that is closely related or a fictional Modern Dutch form (*voorseit* = **voorgezegd** (aforesaid), *eeuwelike* = **eeuwelijk** (eternal), *verlyen* = **verlijden** (confess), *manesse* = **manis** (summons));

Some choices are controversial. This is illustrated by the fact that we have chosen to regard the form *hem* (him) as Pron(refl,3,sing) of the lemma **zich** (itself). In addition, forms such as *luden* and *lieden* (men, people, folk) are categorized under the lemma **man** (man).

3. Loanwords are assigned a form that generally resembles the Dutch form most closely (e.g. *pays* = **pais**).

Tagging

The Adelheid tagset distinguishes the following main classes:

- N Noun
- Adj Adjective
- V Verb
- Adv Adverb
- PronAdv Pronominal adverb
- Pron Pronoun
- Art Article
- Num Numeral
- Adp Adposition
- Conj Conjunction
- Misc Miscellaneous
- Punc Punctuation

During the manual tagging of CRM, choices for word sort and number have been based on the word in its syntactic embedding. The syntactic environment clarifies whether the form *die* (that) is a pron(def) or a pron(dem). The form *jaer* (year) can occur both as singular or plural depending on the

context. Adelheid attempts to reproduce this behaviour, but is likely to make mistakes when a wider context is needed to determine the appropriate choice.

Reliability of the Adelheid System

The reliability of the lemmas and tags assigned by Adelheid depends on the type of text, its date of composition, and location of origin. Adelheid has been trained using the manually labeled and tagged CRM corpus of approximately 2,700 14th-century documents composed in Belgium and the Netherlands. The more a still to be coded text resembles a 14th-century text available in the corpus, the higher the reliability of the lemmas and tags that Adelheid provides for this new text. It has to be considered, however, that a tagged corpus of 2,700 documents is not entirely homogeneous with respect to tags and lemmas. Different people at different time intervals have contributed to the corpus, and although agreements have been made to create uniform tags and lemma labels this cannot rule out the occurrence of certain (arbitrary) choices entirely.

Noun

When assigning lemmas, the substantive is usually brought back to the nominative singular. We maintain the *-e* ending of the nominative if this is acceptable in Modern Dutch (for example in the case of the entry *hoeve* (farmstead)).

Diminutives are generally treated as a nominative singular. The diminutive *straetken* (little street) is assigned the lemma **straat** (street), for example.

Adelheid contains the following tags for substantives:

N(sing)/000/ = noun singular (example *jaer* = lemma **jaar** (year))

N(sing,forme)/001/ = noun singular flexion *-e* (*jaere* = lemma **jaar** (year))

N(sing,formn)/004/ = noun singular flexion *-n* (*jaeren* = lemma **jaar** (year))

N(sing,forms)/002/ = noun singular flexion *-s* (*jaers* = lemma **jaar** (year))

N(sing,formlatin)/[no such tag]/ = noun singular flexion Latin (*vicarius* = lemma **vicaris** (vicar))

N(sing,unclear)/009/ = noun singular form/flexion unclear (*[unreadable form in manuscript]* = lemma **jaar**; the word is illegible)

N(plu,formbase)/010/ = noun plural (no plural ending) (*jaer* = lemma **jaar** (year))

N(plu,forme)/011/ = noun plural flexion *-e* (*jaere* = lemma **jaar** (year))

N(plu,formn)/014/ = noun plural flexion *-n* (*jaeren* = lemma **jaar** (year))

N(plu,formr)/015/ = noun plural flexion *-r* (*kinder* = lemma **kind** (child))

N(plu,forms)/012/ = noun plural flexion *-s* (*zwaghers* = lemma **zwager** (brother in law))

N(plu,formlatin)/[no such tag]/ = noun plural flexion Latin (*prochianem* = lemma **parochiaan** (member of the parish))

N(plu,unclear)/019/ = noun plural form/flexion unclear

Note: As should be expected, the tag N(sing,formlatin) and the latter three tags are not found in combination with the substantive *jaer* (year).

Comments on the above tags:

The tag N(sing,forme) is assigned to all forms of the substantive that end in *-e*, even when it concerns a weak noun that has a stem ending in *-e* in Middle Dutch (like *vrouwe* (lady, wife)), or when a Modern Dutch form has a stem ending in *-e* (like *hoeve* (farmstead) or *bode* (messenger)).

The tag N(sing, formn) is assigned to all endings of the substantive that end with *-n*. In case the stem ends in an *-n*, the word is given the tag N(sing) without encoding for its particular ending.

The tag N(sing,forms) is assigned to all endings of the substantive ending in an *-s* or *-x* (for example in *coninx* (king), in which the *s* is part of the *x*; compare *conincs*). Tagging for this particular element has been a bit too extensive, causing words with a stem ending in *-s* also having been tagged as N(sing,forms) (examples include forms such as *huis* (house), *cijns* (tax), and *vonnis* (verdict)).

The tag N(plu,formbase) indicates that we are dealing with a plural without a plural ending. Older forms of Dutch show forms of neuter words with a heavy stem syllable but without a separate plural marker. Examples include *jaer* (year(s)) and *dinc* (lawsuit(s), thing(s)). Next to that, fourteenth century Dutch has forms such as *scepen* (alderman) and *morgen* (land measure) without plural ending.

The tag N(plu, forme) is assigned to cases in which a plural ending consists of stem + *-e*. In addition, this tag appears with plurals (based on the context) the stems of which already end in an *-e* (for example *oorkonde* (charter), *jaergetijde* (season), and *getuige* (witness)).

The tag N(plu,formn) is assigned when a plural consists of stem + *-en* (or a variant thereof).

The tag N(plu,formr) is found with words that have an *-er/-ere* plural ending in older variants of Dutch (compare the modern Dutch 'staple plurals' *kleed-kl(ed)eren* (cloth-cloth(e)s), *ei-eieren* (egg-eggs), *kind-kind~~er~~en* (child-child~~ren~~). This tag is occasionally found with words for which the stem ends in an *-r*, for example *bunder* (land measure).

The form N(plu,forms) is assigned if a plural ending consists of stem + *-s* (or an apt alternative). At times, this tag is found in plurals that have a stem ending in *-s* but which lack a proper plural ending (*gras* understood as 'grassen' (grasses), *mis* (masses), *huis* (houses), or *mens* (humans, people)).

Proper names denote individual entities. The following forms are distinguished:

- a. first names (*Johannes, Claes, Wouter*)
- b. family names, surnames (*(Jacob) van Leyden*)
- c. names of places, cities, or locations (*Leyden, Brussel, etc.*)
- d. names of streets (*Jacobstraet*)
- e. names of saints and holy men (*Martinus* (Martin), *Maria* (Mary), *Johannes* (John); also includes God)
- f. days of celebration (*kerstmis* (christmas), *sinxen* (Pentecost/Whitsun), *middenzomere* (midsummer), *midwinter* (midwinter))
- g. months (*januari* (January), *maert* (March))

Ad a. comments on first names: First names are not completely consistently traced back to the basic form (lemma) in the original corpus. This could lead to irregularities in the output of Adelheid.

Ad b. comments on family names (or surnames): It is quite often unclear whether we are already dealing with proper family names. In the example of *Johannes den cupre*, *den cupre* could either be a family name or a professional reference (which it would have been originally (cooper)). Since the context generally does not hint at a particular interpretation, the choice for either option seems to be an arbitrary one.

Ad e. and f.: Names that include days of celebration and masses for saints and holy men are generally tagged as follows. Modern Dutch *sint jansmis* (saint John's mass) = **sint** (saint) + **Johannes** (John) + **mis** (mass). Exceptions include *kerstmis* (Christmas), which has been given the lemma **kerstmis** (Christmas), although it is noted that this originates from Christ's mass.

When assigning lemmas to proper names, it is not always to the 'original' stem form that the new form is traced back. De family name *berghen* in *gheertrwd van berghen*, for example, is traced back to **bergen** (mountains) and not to **berg** (mountain). Similarly, the family name *coppen* is assigned the lemma **koppen** (heads).

Adelheid contains the following tags for proper names:

N(prop)/020/ = noun proper name

N(prop,forme)/021/ = noun proper name flexion *-e*

N(prop,formn)/024/ = noun proper name flexion *-n*

N(prop,forms)/022/ = noun proper name flexion *-s*

N(prop,formlatin)/[no such tag]/ = noun proper flexion Latin

N(prop,unclear)/029/ = noun proper name form/flexion unclear

Comments concerning these tags:

The tag N(prop) is assigned when the proper name has no particularly salient ending, such as *Willaem, dirc, haerlem en leyden*.

The tag N(prop,forme) is assigned when the proper name ends in an *-e*, for example in *voorhoute, agniese, van den bossche, bekensteyne*, but also if the stem tends to look like an *-e ending* (van *alcmade, tetterode*).

The tag N(prop,formn) is assigned when the proper name ends in an *-(e)n*, such as *aliden, romen, calenstraten, brunen*.

The tag N(prop,forms) is assigned when the (genitive) ending of the proper name ends in an *-s* (e.g. *boudewijns* (of Baldwin)), but also when the ending already ends in an *-s* (for example in *georgius* and *laurens*).

N(prop,formlatin) is assigned when the ending does not fit in the 'Dutch system' properly, for example word forms such as *augustio* or *octobri*.

The tag N(prop,unclear) is assigned when it is generally difficult to interpret the ending (because it is illegible or contains abbreviations that are hard to interpret).

Adjective

Assigning of lemmas:

The adjective is assigned a lemma without accounting for flexion. The adjective *grote* (big), for example, is assigned to the uninflected lemma **groot** (big).

Comparatives and superlatives:

The comparatives and superlatives are not categorized with the lemma of the positive but are instead given their own lemmas (cf. *ouder* (older) and *oudst* (oldest) are given the main entries **ouder** en **oudst**, respectively). This avoids problems that arise when these forms are assigned to the lemma **oud**, since the comparative form could coincide with forms having a case ending in *-er*.

The following words, representing problematic cases, are tagged as adjectives:

-sent(e), sint(e) etc. (saint) (*op den dach na sente agneten* (on the day after Saint Agnes))

-ene (*die ene zide* (the one side))

-ander (other) (*die ander zide* (the other side))

-vorseit, vornoemt, vorscreven (aforesaid, aforementioned, aforescited), *lestgeleden* (last) (forms that can occur both before and after a noun)

-groot can occur as a name for a coin (as in *enen halve oude groot* (a half old 'groot')). At the same time, however, there are cases such as *ouden groeten conings tornoysen* (old, big 'tornoysen' (= coins) of the king)). Adelheid is not immaculate in deciding whether we are dealing with an adjective or a noun.

Difference between an adjective and a participle

A participle functioning as an attributive adjective in a compound noun group is tagged as an adjective. *Gezegeld* (sealed) is an adjective in *de gezegelde brief* (the sealed letter), whereas it is tagged as a participle in *de brief gezegeld met ons zegel* (the letter sealed with our seal) because it is part of a complement.

The following tags of the adjective are used in Adelheid:

Adj()/100/ = adjective

Adj(forme)/101/ = adjective flexion *-e*

Adj(formn)/104/ = adjective flexion *-n*

Adj(formr)/105/ = adjective flexion *-r*

Adj(forms)/102/ = adjective flexion *-s*

Adj(unclear)/109/ = adjective flexion is unclear (part of the word has been shortened or is illegible)

Comments regarding these tags:

The tag Adj(forms) is also assigned to words such as *kommerloos* (without trouble, trouble free) and *schadeloos* (without damage, damage free) despite the fact that the stem of these words already carries a final *-s*.

Verb

General remarks:

Assigning of lemmas:

The lemma is always the infinitive.

Conjunctive:

Although the imperative has a separate designation, the conjunctive does not have any special tag.

Finite verb:

There is no explicit tag for the person of the finite verb. Consecutively, there is no way of telling whether we are dealing with a first person singular or a third person plural. It is possible, however, to distinguish between past or present tense.

The following tags of the verb are used in Adelheid:

V(infin)/250/ = verb infinitive

V(infin,forme)/251/ = verb infinitive flexion -e

V(infin,formn)/254/ = verb infinitive flexion -n

V(infin,unclear)/259/ = verb infinitive flexion unclear

V(fin,pres,aux_cop)/210/ = verb finite present auxiliary verb or copula

V(fin,pres,aux_cop,forme)/211/ = verb finite present auxiliary verb or copula flexion -e

V(fin,pres,aux_cop,formn)/214/ = verb finite present auxiliary verb or copula flexion -n

V(fin,pres,aux_cop,formnt)/217/ = verb finite present auxiliary verb or copula flexion -nt

V(fin,pres,aux_cop,formt)/213/ = verb finite present auxiliary verb or copula flexion -t

V(fin,pres,aux_cop,unclear)/219/ = verb finite present auxiliary verb or copula form/flexion unclear

V(fin,pres,lex)/200/ = verb finite present lexical

V(fin,pres,lex,forme)/201/ = verb finite present lexical flexion -e

V(fin,pres,lex,formn)/204/ = verb finite present lexical flexion -n

V(fin,pres,lex,formt)/203/ = verb finite present lexical flexion -t

V(fin,pres,lex,unclear)/209/ = verb finite present lexical form/flexion unclear

V(fin,past,aux_cop)/230/ = verb finite past auxiliary verb or copula

V(fin,past,aux_cop,forme)/231/ = verb finite past auxiliary verb or copula flexion -e

V(fin,past,aux_cop,formn)/234/ = verb finite past auxiliary verb or copula flexion -n

V(fin,past,aux_cop,format)/237/ = verb finite past auxiliary verb or copula flexion -nt

V(fin,past,aux_cop,format)/233/ = verb finite past auxiliary verb or copula flexion -t

V(fin,past,lex)/220/ = verb finite past lexical

V(fin,past,lex,forme)/221/ = verb finite past lexical flexion -e

V(fin,past,lex,formn)/224/ = verb finite past lexical flexion -n

V(fin,past,lex,formnt)/227/ = verb finite past lexical flexion -nt

V(fin,past,lex,formt)/223/ = verb finite past lexical flexion -t

V(fin,past,lex,unclear)/209/ = verb finite past lexical form/flexion unclear

V(imp)/240/ = imperative

V(imp,formt)/243/ = imperative flexion -t

V(participle,pres,forme)/271/ = verb participle presens flexion -e

V(participle,pres,forms)/272/ = verb participle presens flexion -s

V(participle,past,formn)/274/ = verb participle past flexion -n

V(participle,past,formt)/273/ = verb participle past flexion -t

V(participle, past, unclear)/279/= verb participle past form/flexion unclear (e.g. *betal, vernouch*)

V(prtcl)/285/ = verb particle

Comments regarding these tags:

Lexical verbs:

The tags V(fin,pres,lex) and V(fin,past,lex) indicate the present and past tense forms of lexical verbs, respectively.

Copular and auxiliary verbs:

The tags V(fin,pres,aux_cop) and V(fin,past,aux_cop) indicate the present and past tense of either copular or auxiliary verbs, respectively.

Copular verbs: zijn, wesen (to be), *bliven* (to stay), *werden* (to become), *schinen* (to appear (to)), *heten* (to be called), and *dunken* (to be of the opinion).

Auxiliary verbs:

time: *hebben* (have), *zijn* (be), *zullen* (shall);

aspect: *plegen* (durative) (commit), *beginnen* (start), *varen* (go), and *gaan* (ingressive) (go);

passive voice: *werden* (become), *zijn* (are);

modality: *willen* (will), *mogen* (may), *zullen* (shall), *moeten* (must), *durven* (dare, may, must, can), *kunnen* (can) and *hebben* (have);

causality: *doen* (do), *laten* (let, leave);

imperative: The imperative can be described using one of the verbs *zullen* (shall) or *willen* (want, will);

description: *doen* (do), *plegen* (commit).

Imperative:

The tag V(imp) indicates that we are dealing with an imperative (second person singular)

The tag V(imp,formt) indicates that we are dealing with an imperative ending in -t (second person plural)

Infinitive:

The tag V(infin) indicates the infinitive without a particular ending.

The tag V(infin,forme) is assigned to infinitives with an ending in -e, such as in: *hi began te spreken* (he began to speak). Nota bene: *van gheldene* is tagged as an infinitive.

The tag V(infin,formn) is assigned when an infinitive ends in an extra -(e)n (*wetenen* (to know), *houdenen* (to hold))

Participles:

The infinitive also serves as the lemma for participles.

The tag V(participle,pres,forme) indicates a present participle that is not used attributively (e.g. *Rorende van dier ander helfte* (concerning the other half).

The tag V(participle,pres,forms) is assigned to adverbs that are formed on the basis of present participles. The adverb *blijvens* (net), for example, finds its basis in the present participle *blijvend(e)* (remaining).

The tag V(participle,past,formt) is assigned to past participles of a weak verb that are not used attributively (*gehelijc dat vorseit es* (as said above)).

The tag V(participle,past,formn) is assigned to a past participle of a strong verb that is not used attributively (*is ghedaen* (has been done), *is ghelegen* (is/has been situated), etc.)

Separable (compound) verbs:

The tag V(prtcl) is assigned to the non-verbal part of the separable verb when this element is found unattached.

Tags of verbal inflection:

The tag formt is assigned to inflections ending in a *-d* or a *-t* (*werkt* (works), *zegt* (says), etc.). If the stem of the verb ends in a *-d* or a *-t* (*worden* (to become), *vinden* (to find), *moeten* (to have to), *houden* (to hold), etc.), the inflected forms of the verb (*hij houdt/hout* (he holds), *hij wordt/wort* (he becomes), *hij vind* (he finds)) do not receive an ending. The tag formt is consequently not assigned here.

In contrast, past participles that have a stem ending in a *t*-sound generally do receive the tag formt (*geschiet* (happened), *geset* (placed)).

The tag formn is assigned to past participles of strong verbs and with plural inflected verbs, but not with infinitives.

Adverb

An adverb modifies any sentence element, excepting nouns (in which case it would not be an adverb but an adjective). One of the characteristics of an adverb is that it is not subject to inflection and does not receive an ending except for its comparative and superlative form.

Examples:

Vele - meer - meest (many - more - most)

Wel - bat - best (good - beter - best)

These forms have been assigned main entries in the inflected form:

Veel - meer - meest (many - more - most)

Wel - bat - best (good - beter - best)

These forms have been assigned the tag adverb(gener)/500/.

The following adverbs are distinguished within Adelheid:

- a. adverb (general)
- b. demonstrative adverb
- c. relative adverb
- d. interrogative adverb
- e. indefinite adverb
- f. resuming adverb
- g. negative adverb

Ad a. This pertains to adverbials of time, for example, such as: *voorts* (hence), *immermeer* (evermore), *jaarlijks* (yearly). But also: *erfelijk* (hereditary), *ook* (also), *genoeg* (enough), etc.

Ad b. This pertains to demonstrative adverbs such as the word *alsoe* (so) in the sentence: *alsoe verre alst onder ons ghelegghen es* (as far as it is in our possession)

The word *aldus* (thus) in the subsequent sentence is a demonstrative adverb as well: (...) *dat al dese voerwaerden aldus geschiet zijn* (that all these preconditions have thus happened)'.

Ad c. This mainly pertains to the relative adverbs *daer* (there (where)) and *toen* (then (when)).

Examples are provided in the following two sentences:

(...) in der stat van Eyke jn eynich huys daer meyster heynric voers(creuen) of sine erfnamen denen

Cens ontfaen willen (in the city of Eyke (proper name) in a house where master heinric

(aforementioned) of his heirs the tax wants to receive)

Dit was ghedaen jnt jaer ons heren doe (= toen) men screef M ccc En(de) sess (This was done in the year of Our Lord when it was written M ccc and six)

Ad d. This mainly pertains to the interrogative adverbs *waer* (where) and *hoe* (how), illustrated in the following two sentences:

(...) onsen goed so waer dat ghelegghen es binnen ghend (our good as where it is situated within Ghent (a city in Belgium))'

(...) al dat recht dat wi hebben ane al die vischerie in die Lake ende in die Vliet hoe sie legghet ochte ligghen mach ochte hoe dat ghenoeemt si (al the rights that we have to all the fishing in the lake and in the stream and no matter how they are called, or where they are)

Ad e. This pertains to indefinite adverbs such as: *nergens* (nowhere), *ergens* (somewhere), *anderswaer* (elsewhere), and *so* (so, thus) in a sentence such as:
(...) ende alle de goede soe (= 'zo') waer dat sie ghelegghen sijn (...)

Ad f. This pertains to the resuming adverb *so* (so) in sentences such as:

In ghetughenisse der waerheyt so hebben wir schepen voerscreven partien onsen ghemeynen syegel aen desen brief doer desen anderen ghesteken ghehangen. (To confirm the truth so we have (...))

Ad g. This pertains to the negative adverbs *en*, *ne*, *-n*, and *niet* 'not'. Middle Dutch quite regularly shows forms of negative concord. In such cases, a negative particle *en* or *ne* is positioned before the finite verb, and the remainder of the sentence contains a second negative constituent.

Waeromme ne gaedi tot hare niet (why don't you go to her?)

The following tags are used for adverbs:

Adv(gener)/500/ = adverb general

Adv(dem)/510/ = adverb demonstrative

Adv(rel)/520/ = adverb relative

Adv(inter)/530/ = adverb interrogative

Adv(indef)/540/ = adverb indefinite

Adv(resum)/550/ = adverb resuming

Adv(neg)/560/ = adverb negative

Pronominal Adverb

A pronominal adverb is a combination of an adverb such as *er* (there), *hier* (here), *daar* (there), *waar* (where), and one or more prepositions (for example *aan* (on), *bij* (by, next to), *voor* (in front of)). These prepositions probably originated from older adverbs. The resulting pronominal adverbs have forms such as *daarvan* (thereof), *hierin* (herein), and *waardoor* (whereby).

Pronominal adverbs can occur as separated elements in a sentence:

(...) *die wert op een dobbelspel gesteken daer hi aen sterff* ((that one) was stabbed during a dice game and he died of this))

The following pronominal adverbs are distinguished:

- a. Personal pronominal adverbs (*erover* (over/across it))
- b. Demonstrative pronominal adverbs (*hiermede* (herewith), *daarmede* (therewith))
- c. Relative pronominal adverbs (*daeraen* (thereon))
- d. Interrogative pronominal adverbs (*waarom* (why), *waarop* (what (...) on/for))
- e. Indefinite pronominal adverbs (NB this tag does not occur in the CRM)

Furthermore, the tag `PronAdv(pre)/655/` is assigned when the preposition occurs separated from the adverb.

The following tags are used for pronominal adverbs:

`PronAdv(pers)/605/` = pronominal adverb personal

`PronAdv(dem)/615/` = pronominal adverb demonstrative

`PronAdv(rel)/625/` = pronominal adverb relative

`PronAdv(inter)/635/` = pronominal adverb interrogative

`PronAdv(indef)/645/` = pronominal adverb indefinite (NB this tag is does not occur in the CRM)

`PronAdv(pre)/655/` = pronominal adverb preposition

Pronoun

A pronoun generally is a word that denotes an independent entity (something or someone) such as 'that' or 'she'.

The following pronouns are distinguished:

- a. Personal pronouns
- b. Demonstrative pronouns
- c. Relative pronouns
- d. Interrogative pronouns
- e. Indefinite pronouns
- f. Possessive pronouns
- g. Reflexive pronouns

Ad a. Personal pronouns:

The inflections of the personal pronouns in Middle Dutch:

	first person	second person		
Singular	(I)	(you)		
Nom.	ic	du/-tu		
Gen.	mijns	dijns		
Dat.	mi	di		
Acc.	mi	di		
Plural	(we)	(you)		
Nom.	wi	ghi/-i		
Gen.	onser	uwer		
Dat.	ons	u		
Acc.	ons	u		
Third person				
Singular	(he)	(she)	(it)	
Nom.	hi/-i	si/-se	het/-(-e)t	
Gen.	sijns/-(-e)s	haer/-ere, -re, -er	-/-(-e)s	
Dat.	hem/-em, -en	haer/-ere, -re, -er	hem	
Acc.	hem/-en, -ene, -ne	haer/-se	het/-(-e)t	
Plural	(they)			
Nom.	si/-se			
Gen.	haer/-ere, -re, -er			
Dat.	hem, hen/-en			
Acc.	hem, hen/-se			

When assigning lemmas for the personal pronouns we chose not to distinguish between subject forms *ik, hi* (I, he), object forms *mi, hem* (me, him), and genitive forms *mijns, sijns* (mine, his). These forms have been subsumed under the subject form **ik, hij, wij, zij** (I, he, we, they) etc.

Pron(pers,1,plu)/404/ = pronoun personal first person plural (*wi, ons*)

Pron(pers,1,sing)/401/ = pronoun personal first person singular (*ic/ich, mi, mijns*)

Pron(pers,2,plu)/405/ = pronoun personal second person plural (*ghi, u, uwer*)

Pron(pers,2,sing)/402/ = pronoun personal second person singular (*du, di, dijns*) (NB this tag is does not occur in the CRM)

Pron(pers,3,plu)/406/ = pronoun personal third person plural (*sij, hen, haer, hoer*)

Pron(pers,3,sing)/403/ = pronoun personal third person singular (*hij, het, si, haar, hem, sijns*)

Ad b. Demonstrative pronouns:

The inflections of the demonstrative pronouns in Middle Dutch:

Masculine

Singular	Nom.	die/de
	Gen.	dies/des
	Dat.	dien/den
	Acc.	dien/den

Plural	Nom.	die
	Gen.	dier/der
	Dat.	dien/den
	Acc.	die

Neuter

Singular	Nom.	dat
	Gen.	dies/des
	Dat.	dien/den
	Acc.	dat

Plural	Nom.	die
	Gen.	dier/der
	Dat.	dien/den
	Acc.	die

Feminine

Singular	Nom.	die
	Gen.	dier/der
	Dat.	dier/der
	Acc.	die

Plural	Nom.	die
	Gen.	dier/der
	Dat.	dien/den
	Acc.	die

The following demonstrative pronouns are distinguished within Adelheid: *die* (definite that), *deze* (definite this), *dit* (indefinite this), *dat* (indefinite that), *gene* (that, the other), *zulk* (such), *alsulke* (such), *zodanige* (such), *dezelfde* (the same), *diezelfde* (that same), etc.

The Modern Dutch articles *de* and *het* originated from the unstressed pronouns *die* and *dat*. *Die* and *dat* can occur both as demonstrative pronouns or as articles in Middle Dutch. These words have been tagged as articles during the manual tagging of the Corpus veertiende-eeuws (CRM) unless the context clearly shows that they are meant to be read as demonstrative pronouns. An automated tagging program such as Adelheid obviously cannot incorporate such semantic considerations and could therefore produce some tags that will be contentious.

When assigning lemmas to demonstrative pronouns it is customary for Adelheid to generate the nominative form: the dative form *desen hospitale* (this hospital) is assigned the lemmas **dit** (this) + **hospitaal** (hospital). Note that this principle has not always been followed consistently in the CRM.

Pron(dem)/410/ = pronoun demonstrative (*dat, dit, die, alsulk*, enz.; all forms without a particular salient ending)
 Pron(dem,forme)/411/ = pronoun demonstrative flexion -e (*dese, gene, de, die*, enz.; all forms ending in -e)
 Pron(dem,formn)/414/ = pronoun demonstrative flexion -n (*desen, genen, den, dien*, etc.; all forms ending in -n)
 Pron(dem,formr)/415/ = pronoun demonstrative flexion -r (*deser, gener, der, dier*, enz.; all forms ending in -r)
 Pron(dem,forms)/412/ = pronoun demonstrative flexion -s (*deses, des, dies, alsulkes, diezelfs*; all forms ending in -s)
 Pron(dem,unclear)/419/ = pronoun demonstrative form/flexion unclear (*dem* (ending -m) or the form is illegible)

Ad. c. Relative pronouns:

Relative pronouns only occur in subclauses and serve as replacements for the antecedent. Middle Dutch examples include: *die* (who, whom, which), *dat* (that, which, who whom), *wat* (that, which, what), *wie* (who), *welke* (who, whom, which), *dewelke* (who, which, that), *hetwelk* (which).

Die and *dat*, as well as *wie* (who) and *wat* (what), are also found as relative pronouns with an internal antecedent:

Ende wie holder is des breves = 'en diegene die bezitter is van de brief'

And who holder is of this letter = and the one who is the possessor of this letter

The forms *dies* and *dien* can occur as the object and genitive form of *dat* 'that'.

Pron(rel)/420/ = pronoun relative (*dat, wat, welk*)
 Pron(rel,forme)/421/ = pronoun relative flexion -e (*die, wie, welke*)
 Pron(rel,formn)/424/ = pronoun relative flexion -n (*dien, wien, welken*)
 Pron(rel,formr)/425/ = pronoun relative flexion -r (*dier, wier, welker*)
 Pron(rel,forms)/422/ = pronoun relative flexion -s (*dies, wes, wiens, welcs*)
 Pron(rel,unclear)/429/ = pronoun relative form/flexion unclear (*doe, dem*; form is illegible)

Ad. d. Interrogative pronoun:

The following interrogative pronouns are found in older forms of Dutch:

		Masculine	Feminine	Neuter
Singular	Nom.	wie	wie	wat
	Gen.	wies/wes	wier	wies/wes
	Dat.	wien	wier	wien
	Acc.	wien	wie	wat
Plural	Nom.	wie		
	Gen.	wier		
	Dat.	wien		
	Acc.	wie		

Pron(inter)/430/ = pronoun interrogative (*wat*)

Pron(inter,forme)/431/ = pronoun interrogative flexion -e (*wie, we, welke*)

Pron(inter,formn)/434/ = pronoun interrogative flexion -n (*wien, welken*)
Pron(inter,formr)/435/ = pronoun interrogative flexion -r (*welker*)
Pron(inter,forms)/432/ = pronoun interrogative flexion -s (*wies*)
Pron(inter,unclear)/439/ = pronoun interrogative form/flexion unclear (*wiem*)

We observe that, in the CRM, forms such as *wiem* (who) receive tag Pron(inter)/430/ and Pron(inter,unclear)/439/, *wie* (who) has code Pron(inter)/430/ and Pron(inter,forme)/431/ (here as well tagging is not consistent). Apart from that, the form *wie* (who) has occasionally been coded as an indefinite pronoun. A large proportion of these cases could - and possibly should - be categorized with the interrogative pronouns, however. On top of that there are cases that constitute a relative pronoun with an internal antecedent.

Ad. e. Indefinite pronoun:

The following indefinite pronouns, amongst others, are distinguished in Middle Dutch: *ieghelijc* (each, every), *iets* (something), *niets* (nothing), *(ne)geen* ((not) any), *enig* (only), *al* (all), *men* (one), *elk* (every).

Pron(indef)/440/ = pronoun indefinite (*niement, elc*)
Pron(indef,forme)/441/ = pronoun indefinite flexion -e (*nymande, neghene*)
Pron(indef,formn)/444/ = pronoun indefinite flexion -n (*neghenen, jgheliken*)
Pron(indef,formr)/445/ = pronoun indefinite flexion -r (*negheere, jenigher*)
Pron(indef,forms)/442/ = pronoun indefinite flexion -s (*nyements, ighelixs*)

Ad. f. Possessive pronoun:

General remarks:

The possessive pronoun developed from the genitive form of the personal pronoun. The first case singular forms consist of *mijn* (mine), *dijn* (yours; e.g. *thine*), *sijn* (his), *haer* (her); and plural *onse* (ours), *uwe* (yours), and *hare* (theirs). The form *hare* is therefore used as the third person plural, although Modern Dutch third person plural *hun* is occasionally found in the form *honne* in fourteenth century Middle Dutch (CRM). *Dijn* (thine) is not found in the CRM, however, quite possibly due to the type of text that has been chosen as the basis for this corpus (e.g. charters). Possessive pronouns are usually inflected as adjectives, but also quite regularly do not receive any particular ending whatsoever.

Assigning tags:

Because forms of *hare* do not only denote the third person feminine singular (as in Present-day Dutch) but also third person plural, we chose to assign the lemma **haar** (her) to the third person feminine singular and the lemma **hun** (their) to the third person plural.

Examples:

- In the subsequent fragment the form 'hare' is assigned to the lemma **haar** (her):
(...) *de voerghenoemde lijsbeth ochte hare nakomelinge* (...) (the aforementioned Lijsbeth (proper name) or her offspring).

- In the subsequent fragment the form *hare* is assigned to the lemma **hun** (their):
(...) *dat zi* (third person plural) *hare zegelen hanghen ane desen letteren* (...) (that they their seals hang on these letter)

In addition, the Middle Dutch forms of the word *haar* (her) are quite diverse, cf.: *eer, oir, hoir, hare, ir, huer, huren, horen*, etc.

The following tags of the possessive pronoun are used in Adelheid:

Pron(poss)/450/ = pronoun possessive (*haer, mijn, sijn*)

Pron(poss,forme)/451/ = pronoun possessive flexion -e (*mine, hare, honne*)

Pron(poss,formn)/454/ = pronoun possessive flexion -n (*sinen, haren, onsen, minen*)

Pron(poss,formr)/455/ = pronoun possessive flexion -r (*siere, siner*)

Pron(poss,forms)/452/ = pronoun possessive flexion -s (*miins, ons, onses*)

Pron(poss,unclear)/459/ = pronoun possessive form/flexion unclear (*sinem*; form is illegible)

Ad. g. Reflexive pronoun:

The personal pronoun is used as reflexive pronoun in early Middle Dutch. This is also pertains to the third person: *hem* (himself). The reflexive pronoun *sich, sic*, in Modern Dutch *zich* (himself, herself, itself), is found in eastern dialects in the fourteenth century (CRM).

Assigning tags:

The reflexive pronouns are all - thus irrespective of form - assigned the lemma 'zich' (itself, himself, etc.).

Pron(refl,1,sing)/461/ = pronoun reflexive first person singular (*mi, mich*)

Pron(refl,2,sing)/462/ = pronoun reflexive second person singular (NB tag does not occur in the CRM)

Pron(refl,3,sing)/463/ = pronoun reflexive third person singular (*hem, home, haer, hoer, sich*)

Pron(refl,1,plu)/464/ = pronoun reflexive first person plural (*ons*)

Pron(refl,2,plu)/465/ = pronoun reflexive second person plural (*uch*)

Pron(refl,3,plu)/466/ = pronoun reflexive third person plural (*hun, hare, oen, um, dich*)

Pron(refl,ERROR)/469/ = pronoun reflexive form/flexion unreadable

Article

As already described under the heading of demonstrative pronouns, the Modern Dutch **definite articles** *de* and *het* originated from the unstressed pronouns *die* and *dat*. *Die* and *dat* can occur both as demonstrative pronouns or as articles in Middle Dutch. These words have been tagged as articles during the manual tagging of the Corpus veertiende-eeuws (CRM) unless the context clearly shows that they are meant to be read as demonstrative pronouns. An automated tagging program such as Adelheid obviously cannot incorporate such semantic considerations and could therefore produce some tags that will be contentious.

Adelheid is trained to generate the nominative form when assigning tags to articles.

In the fragment: (...) *liggende in dymmenhage alrenaest den bosche des vorscreven cloesters (...)* (situated in dymmenhage next to the wood of the prescribed monastery), the articles *den* (the) and *des* (of the) are both assigned the lemma **dat** (the (neuter)). What is generally decisive in such cases is the gender of the noun (in this case: neuter). Note that annotation here is extremely problematic, not only in application, but also in definition. As a result, tagging is prone to extensive inconsistencies on this point.

The following tags are used for the definite article:

Art(def)/470/ = article definite (*dat*, *t-*)

Art(def,forme)/471/ = article definite flexion *-e* (*de*, *die*)

Art(def,formn)/474/ = article definite flexion *-n* (*den*, *dien*)

Art(def,formr)/475/ = article definite flexion *-r* (*der*, *dier*)

Art(def,forms)/472/ = article definite flexion *-s* (*des*, *dies*, *s-*)

Art(def,unclear)/479/ = article definite flexion is unclear (*de?* with indecipherable ending)

The Modern Dutch **indefinite article** *een* originated from the unstressed numeral *een* (one). As with the definite pronoun, some contentious tags are unavoidable.

The following tags are used for the indefinite article:

Art(indef)/480/ = article indefinite (*een*)

Art(indef,forme)/481/ = article indefinite flexion *-e* (*eene*)

Art(indef,formn)/484/ = article indefinite flexion *-n* (*eenen*)

Art(indef,formr)/485/ = article indefinite flexion *-r* (*eenre*, *ener*, *ere*)

Art(indef,forms)/482/ = article indefinite flexion *-s* (*eens*, *enes*)

Numeral

Assigning tags:

Roman numerals are assigned an identical lemma (*ccc* is assigned the lemma **ccc** and *xij* the lemma **xij**). Numerals that are written out are assigned a lemma in letters: *zestiene* is assigned the lemma **zestien** (sixteen).

Tagging:

Cardinal numbers

Tagging the word *een* (one/a) is somewhat problematic. It could either be a cardinal number, an indefinite article, or an adjective (cf. *aen die ene side ende aen die andere side* (on the one side and on the other side)). While manually assigning tags already is hampered by such difficulties regarding the choice between an article, numeral, or adjective, an automated tag program such as Adelheid has similar if not bigger difficulty assigning the best tag. Likewise, the word *ander* (other) could either be an adjective or a numeral meaning 'two'.

Num(card)/300/ = number cardinal

Num(card,forme)/301/ = number cardinal flexion -e

Num(card,formn)/304/ = number cardinal flexion -n

Num(card,formr)/305/ = number cardinal flexion -r

Num(card,forms)/302/ = number cardinal flexion -s

Num(indef)/320/ = number indefinite

Num(indef,forme)/321/ = number indefinite flexion -e

Num(indef,formn)/324/ = number indefinite flexion -n

Num(indef,formr)/325/ = number indefinite flexion -r

Num(indef,forms)/322/ = number indefinite flexion -s

Num(ord)/310/ = number ordinal

Num(ord,forme)/311/ = number ordinal flexion -e

Num(ord,formn)/314/ = number ordinal flexion -n

Num(ord,formr)/315/ = number ordinal flexion -r

Num(ord,forms)/312/ = number ordinal flexion -s

Num(ord,unclear)/319/ = number ordinal unclear

Remarks regarding these tags:

The tag Num(card,forme) is assigned to cardinal numbers ending in -e (*drie* (three), *twee* (two), *ene* (one), *dertiene* (thirteen)).

The tag Num(card,formn) is assigned to cardinal numbers ending in -e(n) (*drien* (three), *tween* (two)).

The tag Num(card,formr) is assigned to cardinal numbers ending in -(e)r/re (*eenre* (one), *drier* (three), *tweer* (two)).

The tag Num(indef) is to indefinite numbers without a particular ending (*al* (all), *elc* (each, every)).

The tag Num(indef,forme) is assigned to indefinite numbers ending in -e (*alle* (all), *elke* (each)).

The tag Num(indef,formn) is assigned to indefinite numbers ending in -(e)n (*allen* (all), *elken* (each)).

The tag Num(indef,formr) is assigned to indefinite numbers ending in -(e)r/re (*aller* (all), *elker* (each)).

The tag Num(indef,forms) is assigned to indefinite numbers ending in -(e)s/x (*alles* (all), *elx* (each)).

The tag Num(ord) is assigned to ordinal numbers without a particular ending (*vijft* = 'vijfde' (fifth), *ert* = 'eerst' (first), *xii* = lemma **xiide** meaning 'twaalfde' (twelfth)).

The tag Num(ord,forme) is assigned to ordinal numbers ending in -e (*irste* (first), *vierde* (fourth), *twelfde* (twelfth)).

The tag Num(ord,formn) is assigned to ordinal numbers ending in -(e)n (*ersten* (first), *tienden* (tenth)).

The tag Num(ord,formr) is assigned to ordinal numbers ending in *-(e)r/re* (*twelfder* (twelfth), *derteinder* (thirteenth)).

The code Num(ord,forms) is assigned to ordinal numbers ending in *-(e)s* (*eerstes* (first), *dertyndes* (thirteenth)).

Adposition

Dutch knows both prepositions and postpositions, which are both tagged Adp. The (grammatical) information that is conveyed by prepositions in Modern Dutch used to be conveyed mainly by a case system in older forms of Dutch. In the Middle Ages, however, already a great many prepositions are found that derive from adverbs originally.

The following tag is used for adposition:

Adp()/700/ = adposition

Examples are: *aen* (on), *achter* (behind), *bachten* (behind), *bi* (by), *beneuen* (next (to)), *bouen* (over, above), *buten* (outside), *doer* (through), *jeghen* (against), *met* (with), *na* (after), *neuen* (next (to)), *om* (around, for, to), *op* (on), *over* (over, across), *sonder* (without), *te* (too), *tot* (until), *tuschen* (between), *van* (of), *voer* (for), *vte* (out)

Conjunction

A conjunction is a word that connects two clauses, in most cases a main clause and a subclause (in which case it is called a subordinating conjunction). A conjunction can also connect a main clause with another main clause, or a subclause with another subclause (in which case it is called a coordinating conjunction). The word *ende* 'and' generally is the typical coordinating conjunction found in Middle Dutch and Early Modern Dutch.

The following conjunctions are distinguished:

Coordinating conjunction (*ende* (and), *of/ofte/ochte* (or), *no/noch* (nor), *maar* (but))

Subordinating conjunction (*dat* (that), *als* (if), *also* (thus), *doe* (then), *omdat* (because), *so* (so, thus), *opdat* (on that, because), *want* (because), *of* (or), *wanneer* (when), etc.)

Comparative conjunction (*als* (if), *dan* (then), *ghelijc* (equal (to)))

Conjunction *alse* in the sense of 'namely': *Kenlic zij allen lieden hoe dat ene voerwaerde ende besprec ghemaect es als van eenen engiene van eere huurclocke te makene binne(n) der steden van ardemburgh.*

Next to the above, the conjunction *en/ne* could be tagged as a conjunction when it occurs in the second part of a biclausal sentence. In that case, we would be dealing with something similar to the modern *balansschikking*, for example in: *Want en es knecht noch here, Hine hebber af bate ende ere (...) Can hi lesen enen brief.* (Want er is geen knecht of heer of hij heeft er baat en eer van (...) als hij een brief kan lezen. (Because there is no servant nor lord who does not benefit or gain honour (...) by being able to read a letter)). Both parts of the combination would be tagged XXX/830/. However, it appears this tag has not been used in the CRM, which also implies that *Adelheid* will never generate it. Without a doubt, works of a more literary nature will show such constructions, despite the scarcity in the current corpus, and we expect it to be used when more tagged material becomes available.

The word *als* 'if' is not tagged quite consistently in the corpus fourteenth century (CRM), which will probably result in tagging problems in *Adelheid*.

The following tags are used for conjunctions:

Conj(coord)/800/ = coordinating conjunction

Conj(subord)/810/ = subordinating conjunction

Conj(comp)/820/ = conjunction of comparison

Conj(XXX)/830/ = XXX conjunction

Conj(alse)/840/ = conjunction 'alse'

Miscellaneous

There are always types of tokens which cannot be described properly with the word classes given above. Examples from tag sets for modern languages are usually mathematical formulae and foreign words. The former do not occur in our current text material. As for the latter, we find only Latin words, for which we have used the tag Misc(latin). Note that there are also cases where Dutch lemmas have been given a Latin form; these we do not tag as Misc(latin), but as their proper class given the lemma, but then with the attribute formlatin.

Apart from Latin words, the main tag Misc is only used for tokens which are mostly unreadable or otherwise uninterpretable, in which cases we use the tag Misc(unclear).

This means we currently distinguished the following forms of miscellaneous tokens:

Misc(latin)/900/ = Latin

Misc(unclear)/999/ = unclear

Punctuation

In this Middle Dutch and Early Modern Dutch period, standardized conventions for punctuation were mostly absent. In CRM we marked the punctuation that was present in the charter.

The following forms of punctuation are distinguished in CRM and Adelheid:

Punc(colon)/Punc(ldp)/ = :

Punc(comma)/Punc(lk)/ = ,

Punc(gcomma)/Punc(ldk)/ = / (the so-called German comma)

Punc(hyph)/Punc(ls)/ = -

Punc(other)/Punc(lv)/ = ?

Punc(period)/Punc(lp)/ = .

Punc(semicolon)/Punc(lpk)/ = ;

Punc(tilde)/Punc(lt)/ = ~

Appendix: Full list of tags

Tags are in alphabetical order (including clitic forms).

The tags as they are currently being used in Adelheid are given first, followed by a numeric tag used in the past (given in between slashes [/ /]) and a short description of each particular code.

Adjective

Adj()/100/ = adjective

Adj(forme)/101/ = adjective flexion -e

Adj(formn)/104/ = adjective flexion -n

Adj(formr)/105/ = adjective flexion -r

Adj(forms)/102/ = adjective flexion -s

Adj(unclear)/109/ = adjective form/flexion unclear

Adposition

Adp()/700/ = adposition

Adp()+Adj()/700+100/ = adposition + adjective

Adp()+Adj(forme)/700+101/ = adposition + adjective flexion -e

Adp()+Adj(formn)/700+104/ = adposition + adjective flexion -n

Adp()+Art(def)/700+470/ = adposition + article definite

Adp()+Art(def,formn)/700+474/ = adposition + article definite flexion -n

Adp()+Art(def,formr)/700+475/ = adposition + article definite flexion -r

Adp()+Art(indef,forme)/700+481/ = adposition + article indefinite flexion -e

Adp()+Art(indef,formn)/700+484/ = adposition + article indefinite flexion -n

Adp()+N(prop)/700+020/ = adposition + noun proper name

Adp()+N(prop,forme)/700+021/ = adposition + noun proper name flexion -e

Adp()+N(sing,forme)/700+001/ = adposition + noun singular flexion -e

Adp()+Num(indef,formn)/700+324/ = adposition + number indefinite flexion -n

Adp()+Pron(indef,formn)/700+444/ = adposition + pronoun indefinite flexion -n

Adp()+Pron(poss,formn)/700+454/ = adposition + pronoun possessive flexion -n

Adp()+V(infin)/700+250/ = adposition + verb infinitive

Adp()+V(infin,forme)/700+251/ = adposition + verb infinitive flexion -e

Adverb

Adv()/500/ = adverb

Adv(dem)/510/ = adverb demonstrative

Adv(gener)/500/ = adverb general

Adv(indef)/540/ = adverb indefinite

Adv(inter)/530/ = adverb interrogative

Adv(neg)/560/ = adverb negative

Adv(rel)/520/ = adverb relative

Adv(resum)/550/ = adverb resuming

Article

Art(def)/470/ = article definite

Art(def)+Adj()/470+100/ = article definite + adjective

Art(def)+Adj(forme)/470+101/ = article definite + adjective flexion -e

Art(def)+N(sing)/470+000/ = article definite + noun singular

Art(def)+N(sing,formn)/470+004/ = article definite + noun singular flexion -n
 Art(def)+N(sing,forms)/470+002/ = article definite + noun singular flexion -s
 Art(def)+Num(ord,forme)/470+311/ = article definite + number ordinal flexion -e
 Art(def)+Num(ord,formn)/470+314/ = article definite + number ordinal flexion -n
 Art(def)+Pron(dem,forme)/470+411/ = article definite + pronoun demonstrative flexion -e
 Art(def)+Pron(poss,forme)/470+451/ = article definite + pronoun possessive flexion -e
 Art(def,forme)/471/ = article definite flexion -e
 Art(def,formn)/474/ = article definite flexion -n
 Art(def,formn)+Pron(dem,formn)/474+414/ = article definite flexion -n + pronoun demonstrative flexion -n
 Art(def,formr)/475/ = article definite flexion -r
 Art(def,forms)/472/ = article definite flexion -s
 Art(def,forms)+N(prop,forms)/472+022/ = article definite flexion -s + noun proper name flexion -s
 Art(def,forms)+N(sing)/472+000/ = article definite flexion -s + noun singular
 Art(def,forms)+N(sing,forme)/472+001/ = article definite flexion -s + noun singular flexion -e
 Art(def,forms)+N(sing,formn)/472+004/ = article definite flexion -s + noun singular flexion -n
 Art(def,forms)+N(sing,forms)/472+002/ = article definite flexion -s + noun singular flexion -s
 Art(def,unclear)/479/ = article definite form/flexion is unclear
 Art(indef)/480/ = article indefinite
 Art(indef,forme)/481/ = article indefinite flexion -e
 Art(indef,formn)/484/ = article indefinite flexion -n
 Art(indef,formr)/485/ = article indefinite flexion -r
 Art(indef,forms)/482/ = article indefinite flexion -s

Conjunction

Conj(alse)/840/ = conjunction 'also'
 Conj(comp)/820/ = conjunction of comparison
 Conj(coord)/800/ = coordinate conjunction
 Conj(coord)+Pron(pers,3,sing)/800+403/ = coordinate conjunction
 Conj(subord)/810/ = subordinate conjunction (subjunction)
 Conj(subord)+Pron(dem,forms)/810+412/ = subordinate conjunction (subjunction) + pronoun demonstrative flexion -s
 Conj(subord)+Pron(pers,3,plu)/810+406/ = subordinate conjunction (subjunction) + pronoun personal third person plural
 Conj(subord)+Pron(pers,3,sing)/810+403/ = subordinate conjunction (subjunction) + pronoun personal third person singular
 Conj(subord)+Pron(pers,3,sing)+Pron(pers,3,sing)/810+403+403/ = subordinate conjunction (subjunction) + pronoun personal third person singular + pronoun personal third person singular
 Conj(subord)+PronAdv(pers)/810+605/ = subordinate conjunction (subjunction) + pronominal adverb personal

Miscellaneous

Misc(latin)/900/ = Miscellaneous word is Latin
 Misc(unclear)/999/ = Miscellaneous word is unclear

Noun

N(prop)/020/ = noun proper name
 N(prop,forme)/021/ = noun proper name flexion -e

N(prop,formlatin) = noun proper flexion Latin
N(prop,formn)/024/ = noun proper name flexion -n
N(prop,forms)/022/ = noun proper name flexion -s
N(prop,unclear)/029/ = noun proper name form/flexion unclear
N(plu,formbase)/010/ = noun plural (no plural ending)
N(plu,forme)/011/ = noun plural flexion -e
N(plu,formlatin)/[no such tag]/ = noun plural flexion Latin
N(plu,formn)/014/ = noun plural flexion -n
N(plu,formr)/015/ = noun plural flexion -r
N(plu,forms)/012/ = noun plural flexion -s
N(plu,unclear)/019/ = noun plural form/flexion unclear
N(sing)/000/ = noun singular
N(sing,forme)/001/ = noun singular flexion -e
N(sing,formn)/004/ = noun singular flexion -n
N(sing,forms)/002/ = noun singular flexion -s
N(sing,formlatin)/[no such tag] = noun singular flexion Latin
N(sing,unclear)/009/ = noun singular form/flexion unclear

Numeral

Num(card)/300/ = numeral cardinal
Num(card,forme)/301/ = numeral cardinal flexion -e
Num(card,formn)/304/ = numeral cardinal flexion -n
Num(card,formr)/305/ = numeral cardinal flexion -r
Num(card,forms)/302/ = numeral cardinal flexion -s
Num(indef)/320/ = numeral indefinite
Num(indef)+Art(def)/320+470/ = numeral indefinite + article definite
Num(indef)+Art(def,formn)/320+474/ = numeral indefinite + article definite flexion -n
Num(indef)+Art(def,formr)/320+475/ = numeral indefinite + article definite flexion -r
Num(indef,forme)/321/ = numeral indefinite flexion -e
Num(indef,formn)/324/ = numeral indefinite flexion -n
Num(indef,formr)/325/ = numeral indefinite flexion -r
Num(indef,forms)/322/ = numeral indefinite flexion -s
Num(ord)/310/ = numeral ordinal
Num(ord,forme)/311/ = numeral ordinal flexion -e
Num(ord,formn)/314/ = numeral ordinal flexion -n
Num(ord,formr)/315/ = numeral ordinal flexion -r
Num(ord,forms)/312/ = numeral ordinal flexion -s
Num(ord,unclear)/319/ = numeral ordinal unclear

Pronominal Adverb

PronAdv(dem)/615/ = pronominal adverb demonstrative
PronAdv(indef)/645/ = pronominal adverb indefinite
PronAdv(inter)/635/ = pronominal adverb interrogative
PronAdv(pers)/605/ = pronominal adverb personal
PronAdv(preposition)/655/ = pronominal adverb preposition
PronAdv(rel)/625/ = pronominal adverb relative

Pronoun

Pron(dem)/410/ = pronoun demonstrative

Pron(dem)+N(sing,forme)/410+001/ = pronoun demonstrative + noun singular flexion -e

Pron(dem)+V(fin,pres,aux_cop)/410+210/ = pronoun demonstrative + verb finite presens auxiliary verb or copula

Pron(dem,forme)/411/ = pronoun demonstrative flexion -e

Pron(dem,formn)/414/ = pronoun demonstrative flexion -n

Pron(dem,formr)/415/ = pronoun demonstrative flexion -r

Pron(dem,forms)/412/ = pronoun demonstrative flexion -s

Pron(dem,unclear)/419/ = pronoun demonstrative form/flexion unclear

Pron(indef)/440/ = pronoun indefinite

Pron(indef)+Pron(pers,3,sing)/440+403/ = pronoun indefinite + pronoun personal third person singular

Pron(indef,forme)/441/ = pronoun indefinite flexion -e

Pron(indef,formn)/444/ = pronoun indefinite flexion -n

Pron(indef,formr)/445/ = pronoun indefinite flexion -r

Pron(indef,forms)/442/ = pronoun indefinite flexion -s

Pron(inter)/430/ = pronoun interrogative

Pron(inter,forme)/431/ = pronoun interrogative flexion -e

Pron(inter,formn)/434/ = pronoun interrogative flexion -n

Pron(inter,formr)/435/ = pronoun interrogative flexion -r

Pron(inter,forms)/432/ = pronoun interrogative flexion -s

Pron(inter,unclear)/439/ = pronoun interrogative form/flexion unclear

Pron(pers,1,plu)/404/ = pronoun personal first person plural

Pron(pers,1,plu)+Pron(pers,3,sing)/404+403/ = pronoun personal first person plural + pronoun personal third person singular

Pron(pers,1,sing)/401/ = pronoun personal first person singular

Pron(pers,2,plu)/405/ = pronoun personal second person plural

Pron(pers,3,plu)/406/ = pronoun personal third person plural

Pron(pers,3,plu)+Pron(dem,forms)/406+406/ = pronoun personal third person plural + pronoun demonstrative flexion -s

Pron(pers,3,plu)+Pron(pers,3,sing)/406+403/ = pronoun personal third person plural + pronoun personal third person singular

Pron(pers,3,sing)/403/ = pronoun personal third person singular

Pron(pers,3,sing)+Pron(pers,3,plu)/403+406/ = pronoun personal third person singular + pronoun personal third person plural

Pron(pers,3,sing)+Pron(pers,3,sing)/403+403/ = pronoun personal third person singular + pronoun personal third person singular

Pron(poss)/450/ = pronoun possessive

Pron(poss,forme)/451/ = pronoun possessive flexion -e

Pron(poss,formn)/454/ = pronoun possessive flexion -n

Pron(poss,formr)/455/ = pronoun possessive flexion -r

Pron(poss,forms)/452/ = pronoun possessive flexion -s

Pron(poss,unclear)/459/ = pronoun possessive form/flexion unclear

Pron(refl,1,plu)/464/ = pronoun reflexive first person plural

Pron(refl,2,plu)/465/ = pronoun reflexive second person plural

Pron(refl,3,plu)/466/ = pronoun reflexive third person plural

Pron(refl,1,sing)/461/ = pronoun reflexive first person singular

Pron(refl,2,sing)/462/ = pronoun reflexive second person singular

Pron(refl,3,sing)/463/ = pronoun reflexive third person singular

Pron(refl,ERROR)/469/ = pronoun reflexive form/flexion unreadable

Pron(rel)/420/ = pronoun relative
 Pron(rel)+PronAdv(pers)/420+605/ = pronoun relative + pronominal adverb personal
 Pron(rel,forme)/421/ = pronoun relative flexion -e
 Pron(rel,forme)+Adv(gener)/421+500/ = pronoun relative flexion -e + adverb general
 Pron(rel,forme)+PronAdv(pers)/421+605/ = pronoun relative flexion -e + pronominal adverb personal
 Pron(rel,formn)/424/ = pronoun relative flexion -n
 Pron(rel,formn)+Pron(pers,3,sing)/424+403/ = pronoun relative flexion -n + pronoun personal third person singular
 Pron(rel,formr)/425/ = pronoun relative flexion -r
 Pron(rel,forms)/422/ = pronoun relative flexion -s
 Pron(rel,unclear)/429/ = pronoun relative form/flexion unclear

Punctuation

Punc(colon)/Punc(ldp)/ = :
 Punc(comma)/Punc(lk)/ = ,
 Punc(gcomma)/Punc(ldk)/ = /
 Punc(hyph)/Punc(ls)/ = -
 Punc(other)/Punc(lv)/ = ?
 Punc(period)/Punc(lp)/ = .
 Punc(semicolon)/Punc(lpk)/ = ;
 Punc(tilde)/punc(lt)/ = ~

Verb

V(fin,past,aux_cop)/230/ = verb finite past auxiliary verb or copula
 V(fin,past,aux_cop)+Pron(pers,3,plu)/230+406/ = verb finite past auxiliary verb or copula + pronoun personal third person plural
 V(fin,past,aux_cop)+Pron(pers,3,sing)/230+403/ = verb finite past auxiliary verb or copula + pronoun personal third person singular
 V(fin,past,aux_cop,forme)/231/ = verb finite past auxiliary verb or copula flexion -e
 V(fin,past,aux_cop,formn)/234/ = verb finite past auxiliary verb or copula flexion -n
 V(fin,past,aux_cop,format)/237/ = verb finite past auxiliary verb or copula flexion -nt
 V(fin,past,aux_cop,format)/233/ = verb finite past auxiliary verb or copula flexion -t
 V(fin,past,lex)/220/ = verb finite past lexical
 V(fin,past,lex)+Pron(pers,3,sing)/220+403/ = verb finite past lexical + pronoun personal third person singular
 V(fin,past,lex,forme)/221/ = verb finite past lexical flexion -e
 V(fin,past,lex,forme)+Pron(pers,3,sing)/221+403/ = verb finite past lexical flexion -e + pronoun personal third person singular
 V(fin,past,lex,formn)/224/ = verb finite past lexical flexion -n
 V(fin,past,lex,formnt)/227/ = verb finite past lexical flexion -nt
 V(fin,past,lex,formt)/223/ = verb finite past lexical flexion -t
 V(fin,past,lex,unclear)/209/ = verb finite past lexical form/flexion unclear
 V(fin,pres,aux_cop)/210/ = verb finite presens auxiliary verb or copula
 V(fin,pres,aux_cop)+Pron(pers,3,sing)/210+403/ = verb finite presens auxiliary verb or copula + pronoun personal third person singular
 V(fin,pres,aux_cop)+PronAdv(pers)/210+605/ = verb finite presens auxiliary verb or copula + pronominal adverb personal
 V(fin,pres,aux_cop,forme)/211/ = verb finite presens auxiliary verb or copula flexion -e
 V(fin,pres,aux_cop,formn)/214/ = verb finite presens auxiliary verb or copula flexion -n

V(fin,pres,aux_cop,formn)+Pron(pers,3,sing)/214+403/ = verb finite presens auxiliary verb or copula + pronoun personal third person singular
V(fin,pres,aux_cop,formnt)/217/ = verb finite presens auxiliary verb or copula flexion -nt
V(fin,pres,aux_cop,formt)/213/ = verb finite presens auxiliary verb or copula flexion -t
V(fin,pres,aux_cop,unclear)/219/ = verb finite presens auxiliary verb or copula form/flexion unclear
V(fin,pres,lex)/200/ = verb finite presens lexical
V(fin,pres,lex)+Pron(pers,3,sing)/200+403/ = verb finite presens lexical + pronoun personal third person singular
V(fin,pres,lex,forme)/201/ = verb finite presens lexical flexion -e
V(fin,pres,lex,formn)/204/ = verb finite presens lexical flexion -n
V(fin,pres,lex,formt)/203/ = verb finite presens lexical flexion -t
V(fin,pres,lex,unclear)/209/ = verb finite presens lexical form/flexion unclear
V(imp)/240/ = imperative
V(imp,formt)/243/ = imperative flexion -t
V(infin)/250/ = verb infinitive
V(infin)+Pron(pers,3,sing)/250+403/ = verb infinitive + pronoun personal third person singular
V(infin,forme)/251/ = verb infinitive flexion -e
V(infin,formn)/254/ = verb infinitive flexion -n
V(infin,unclear)/259/ = verb infinitive flexion unclear
V(participle,past,formn)/274/ = verb participle past flexion -n
V(participle,pres,forms)/272/ = verb participle presens flexion -s
V(participle,past,formt)/273/ = verb participle past flexion -t
V(participle,pres,forme)/271/ = verb participle presens flexion -e
V(participle, past, unclear)/279/= verb participle past form/flexion unclear
V(prtcl)/285/ Verb particle