Workbook: Using Corpora for Linguistic Research

DIGS 13 Workshop

Beatrice Santorini and Joel C. Wallenberg
joel.wallenberg@gmail.com

Contents

1 TP and vP/VP

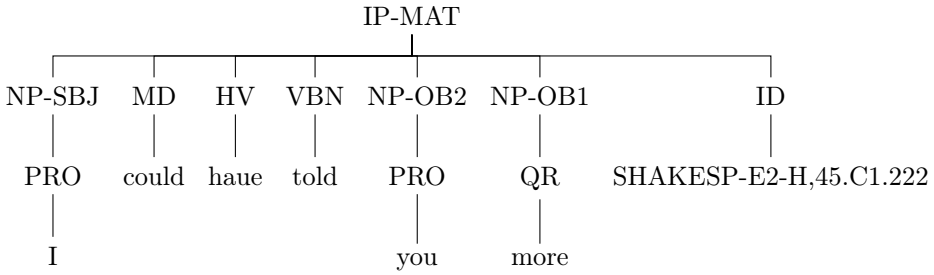
1.1 Immediate dominance, precedence, negation, and wildcards

Every clause in corpora following the Penn format is labelled "IP", with extended tags for various types of clause. We will initially be mostly concerned with finite clauses, subordinate clauses "IP-SUB" and matrix clauses "IP-MAT". All DPs in the corpus are labelled "NP", and the arguments of a verb have extended tags according to their function: "-SBJ", "-OB1" for direct object, and "NP-OB2" for indirect. For full parsing documentation and a full list of tags, the reader should see:

`http://www.ling.upenn.edu/hist-corpora/annotation/`

However, all of the phrase labels in the corpora follow this pattern of basic tag with a possible extended tag: TAG-EXTENDED. In this way, a matrix clause containing an indirect object (as in the structure in (1)) will have the structure below:

(1) **Finite Matrix Clause with Two Objects**



Note that there is no VP represented in the corpus annotation. There are many reasons for this (see the full documentation at the URL above), but two reasons are: first, in many sentences it is difficult to determine the VP-boundaries, particularly in Old and Middle English during the period of change in the structure of vP/VP. Secondly, the VP node would frequently complicate searches without obvious benefit. Leaving out the VP node, on the other hand, does not have any detrimental results, as you will see as you begin to query VP-related structures. Sentences containing the structure above in (1) can be retrieved from the corpus with the following query:

Example: Finite Matrix Clause with an Indirect Object

```
node: IP-MAT
query: IP-MAT idoms NP-OB2
```

The function "idoms" means *immediately dominates*, and it is one of the most useful search functions. If we wanted to find any type of clause containing an indirect object, we replace " **SUB** " with the wildcard " ***** ", which stands in for "any character any number of times".

Any Clause with an Indirect Object

```
node: IP*
query: IP* idoms NP-OB2
```

It is important to remember that the "**node**" is the structure that CorpusSearch actually counts when it tabulates the results. This means that the results for the above two queries will be counting the number of clauses, not the number of indirect objects. Of course, this difference is not relevant for the above example, since no single clause will contain multiple indirect objects. However, the following query looks

for either indirect or direct objects, so the choice of "**IP***" rather than "**NP***" is more important.

Finite Subordinate Clause with an Object

```
node: IP-SUB
query: IP-SUB idoms NP-OB*
```

Use the negator " **!** " to find intransitive clauses. Note that the negator always comes before arguments (as shown below), not in front of the function call itself. In fact, CorpusSearch interprets "**YP idoms !XP**" to mean, "YP immediately dominates some constituent which does not have the label XP, and none of the constituents that YP immediately dominates have the label XP."

Finite, Intransitive Subordinate Clause

```
node: IP-SUB
query: IP-SUB idoms !NP-OB*
```

In order to look at the ordering relationships between clausal constituents, we have to introduce a new search function "precedes". For instance, in order to search for a clause with a subject preceding an object, we can add two clauses to the previous query and arrive at the query below. It is important to note that by default, CorpusSearch treats the same label as referring to the same element if it is mentioned multiple times in the same query.

Clause with a Subject preceding an Object

```
node: IP*
query: (IP* idoms NP-OB*)
AND (IP* idoms NP-SBJ)
AND (NP-SBJ precedes NP-OB*)
```

Since this is a default word order, it will be extremely common in the corpus. In the workshop sample corpus, this query should return a result of 24636 hits (see the very bottom of the search output). Note that these hits occur in 18839 tokens. This is because since the query includes both matrix and subordinate clauses, one clause fitting the pattern may contain an arbitrary number of subordinate clauses which also fit the pattern. CorpusSearch recursively searches all of the nodes described under "node:" in the query and returns all of the hits.

As I mentioned above, multiple references to the same label in a CS query are treated as referring to the same element, by default. If you specifically want to distinguish two mentions of the same label and make sure they refer to different elements in the structure, you need to add indices before the label, as shown below:

Clause with a Subject preceding an Object

```
node: IP*
query: (IP* idoms [1]NP-*)
AND (IP* idoms [2]NP-*)
AND ([1]NP-* precedes [2]NP-*)
```

The example query above searches for two different NPs of some type, and requires one of them to precede the other. If the "**[1]**" and "**[2]**" indices were omitted, the query would return 0 results, because it is not possible for the same element to precede itself. This information may be useful in the exercises

below.

In order to be able to solve Problem 1, you will also need to know how traces are annotated in the corpora and how to reference them in queries. Traces can come in various types (see corpus documentation), with the most common being traces of *wh*-movement, ***T***, and traces of other types of movement like extraposition, ***ICH***. The traces are coindexed with their antecedents by a numerical dash tag, so that a *wh*-extracted subject might have the form: **(NP-SBJ *T*-1)**. In a query, the " * " character is interpreted as a wildcard, as we said above. When searching for an asterisk in the corpus, it is therefore necessary to indicate this explicitly, by prefixing the asterisk with a backslash (this is known as "escaping" the asterisk). This is shown in the example below.

Clause with an Extracted Subject

```
node: IP*
query: (IP* idoms NP-SBJ)
AND (NP-SBJ idoms \*T*)
```

Problem 1
The order of objects in ditransitive clauses.

i. How many clauses contain a indirect object preceding a direct object? (Please find a way to exclude examples where this order is the result of topicalization.)

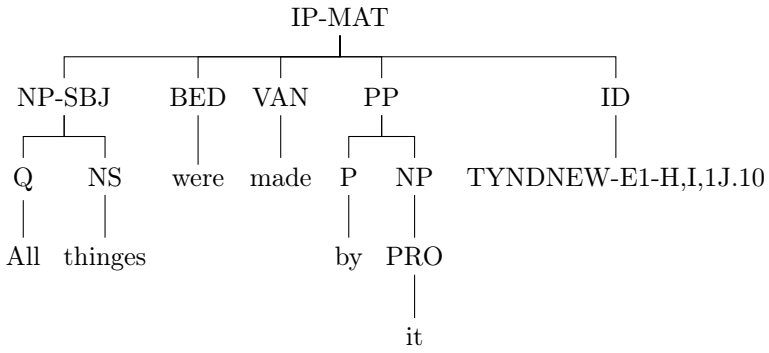
ii. How many clauses contain a direct object preceding an indirect object? (**Hint:** in the query for subordinate clauses, you will need to use the " **!** " negator to exclude traces. As before, please exclude topicalization.)

iii. When does the DO > IO order stop being possible in the language?

1.2 Verb Movement, Auxiliary Constructions, and Disjunction in Queries

The example in (1) above showed a number of verbal categories in the corpora, and this section looks into the verb types and their syntax in a bit more detail. The structure below in (2) shows an example of a passive clause from the PPCEME (Kroch et al., 2005).

(2) **Finite Passive Clause, English**

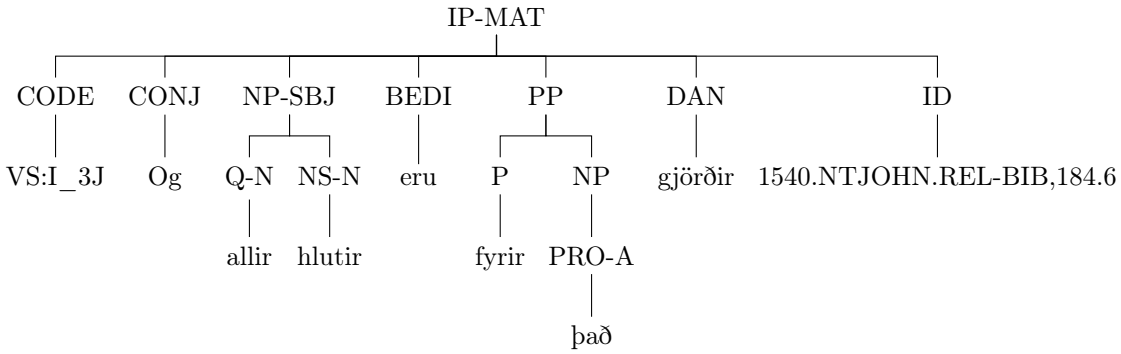


The parse of this sentence in the corpus and in the output files looks like this:

```
( (IP-MAT (NP-SBJ (Q All) (NS thinges))
          (BED were)
          (VAN made)
          (PP (P by)
              (NP (PRO it)))
         (. ,))
  (ID TYNDNEW-E1-H,I,1J.10))
```

As a comparison, the corresponding passive from the Icelandic translation of the New Testament in the Icelandic Parsed Historical Corpus (IcePaHC: Wallenberg et al., 2011) has a very similar representation:

(3) **Finite Passive Clause, Icelandic**

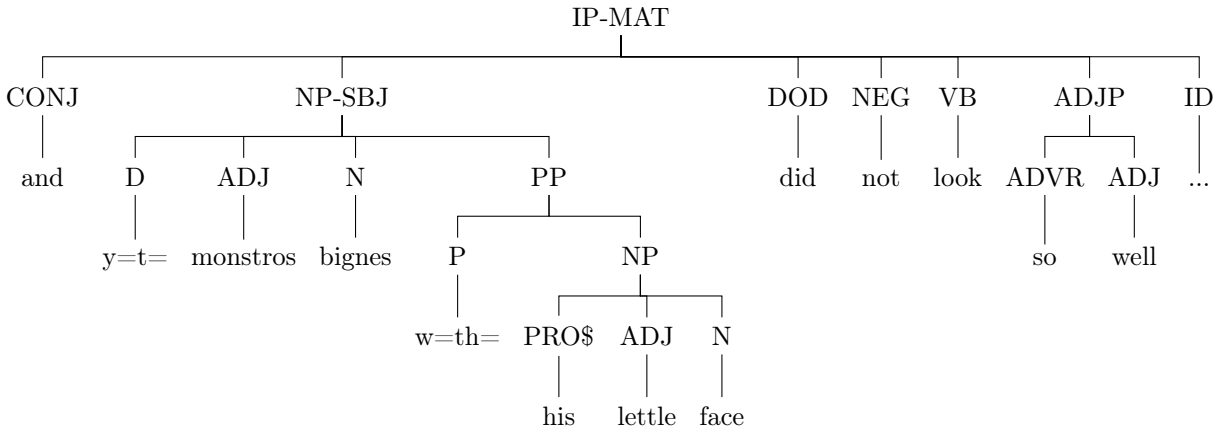


The representation of the Icelandic sentence in IcePaHC is below. You will notice that the parse is nearly identical (by design), with the main differences being: Case is marked on the relevant categories (e.g. " -N ", " -A "), subjunctive and indicative are marked on verbal categories (as they are in the YCOE for Old English; Taylor et al., 2003), and there are lemmas following each word. (Note also that the passive participle is "DAN" because this is a form of the Icelandic verb meaning *do*; see below in the *do*-support example.) This is an example of how the basic Penn corpus format can be augmented with further information without disturbing the basic structures or the ability to run the same queries on corpora in different languages.

```
( (IP-MAT (CODE VS:I_3J)
          (CONJ Og-og)
          (NP-SBJ (Q-N allir-allur) (NS-N hlutir-hlutur))
          (BEPI eru-vera)
          (PP (P fyrir-fyrir)
              (NP (PRO-A það-það)))
          (DAN gjörðir-gera)
          (. ,-,))
  (ID 1540.NTJOHN.REL-BIB,184.6))
```

The following structure is the representation of *do*-support in the corpora:

(4) ***do*-Support in the English Corpora**



Again, the sentence looks like this in the corpus and output files:

```
( (IP-MAT (CONJ and)
          (NP-SBJ (D y=t=)
                  (ADJ monstros)
                  (N bignes)
                  (PP (P w=th=)
                      (NP (PRO$ his) (ADJ lettle) (N face))))
          (DOD did)
          (NEG not)
          (VB look)
          (ADJP (ADVR so) (ADJ well))
(. .))
(ID ALHATTON-E3-H,2,241.20))
```

In order to look for constructions which may occur with a variety of verb and auxiliary forms, it is frequently necessary to use disjunction in your queries. Disjunction between search function calls is "**OR**". It is also frequently useful to use disjunction between node labels, which is " | " (just like it is in many scripting and programming languages). When the " | " disjunction is used between labels in a CorpusSearch query, CS interprets the entire disjunction as representing a single node which could be in the form of any of the disjoined labels. For instance, in the example below, "DO|HV|VB" stands for infinitive DO, HAVE, or some lexical verb, and CS interprets "IP-MAT* idoms DO|HV|VB" to mean: "matrix IP immediately dominates a particular node X, and X may appear as either DO, HV, or VB."

Example: Finite Matrix Clause with Negation and *do*-support

```
node: IP-MAT*
query: (IP-MAT* idoms DOP|DOD)
AND (IP-MAT* idoms VB|HV|DO)
AND (IP-MAT* idoms NEG)
AND (DOP|DOD precedes NEG)
AND (NEG precedes VB|HV|DO)
```

Since CorpusSearch by default treats later mentions of the same label as referring to the same element as earlier mentions, CS will interpret both references to e.g. "**VB|HV|DO**" in the query above as referring to the same infinitive verb. However, it is important to remember that the mentions are only treated in this way if they are exactly *identical*; a mention of "**VB|HV|DO**" and a later mention of "**VB|DO|HV**"

would not be interpreted as referring to the same object. If you run the above query on the DIGS sample corpus, you should find 356 hits.

Problem 2
Auxiliary Constructions.

i. **Excluding *do*-support**, do matrix clauses and subordinate clauses contain the same frequency of auxiliary use? You can restrict the answer to clauses containing lexical verbs (**V***) for the purposes of this exercise.

ii. If there is an asymmetry, can you tell which auxiliary/construction is responsible for the difference? Or is there an asymmetry in more than one? This part of the problem may take a while to do, so it might be better to finish it after the workshop. (Hint: you will need to distinguish perfects, progressives, and modal constructions for this question. You can also do this faster by searching the output of your previous queries, if you are running standard CorpusSearch.)

In order to look at copular clauses, we can make use of the fact that BE is tagged separately with its own tag, but we also need to be careful to exclude the auxiliary uses of BE. The example below shows a query for copular clauses with a finite copula:

Copular Clauses with Finite Copula:

```
node: IP*
query: (IP* idoms BEP|BED)
AND (IP* idoms !V*N|VAG|H*N|HAG|D*N|DAG|BEN|BAG)
```

The above query gets 12652 hits in the DIGS sample corpus, as copular clauses are quite common in all of the corpora. In order to look at all nonfinite copular clauses we can use the query below:

Copular Clauses with Nonfinite Copula:

```
node: IP*
query: (IP* idoms BEN|BAG|BE)
```

The information in the example queries should make the next problem quite easy to solve, with one additional fact: nominal predicates of copular clauses are tagged "**NP-OB1**" in the PPCME2, PPCEME, and PPCMBE. (They are tagged differently from normal objects in the York corpora of Old English and

in the IcePaHC.)

Problem 3
Specificational Sentences
This problem deals with copular clauses of the type in (5) below, in which the predicate is fronted and there is obligatory focus on the (postposed) subject.

(5) (Context: Who is the murderer?) The murderer is John.

i. How many potential specificational matrix clauses with a finite copula are there in the sample corpus?

ii. Does this query yield output that contains **only** specificational sentences? Or are there ambiguous copular clauses in the output?

iii. If there are some ambiguous clauses in your output, what syntactic factor is responsible for the ambiguity?

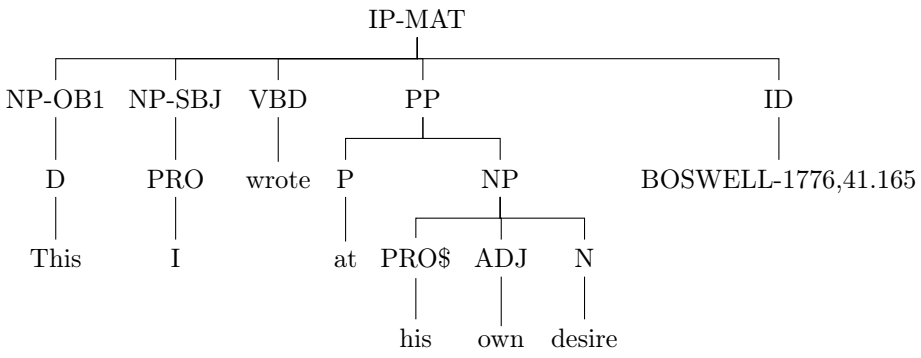
2 Left-periphery and CP-domain

2.1 Topicalization

The most important thing to keep in mind about the annotation conventions for this part of the phrase structure is that: there is a CP node (always with some extended dash tag; e.g. **CP-REL, CP-QUE, CP-THT**) dominating every subordinate clause (**IP-SUB**), but there is no CP-layer included for matrix topicalization, left-dislocation, or matrix V-to-C movement constructions with the exception of direct questions (these are **CP-QUE** with no C node). All left periphery phenomena in matrix non-questions occur under the **IP-MAT** node, just like any other clause type, and any embedded root phenomena occur under an **IP-SUB** node.

Thus, object topicalization has the structure shown in (6) below and the parse shown in (7).

(6) **Direct Object Topicalization**



(7) This sentence in the corpus and output files:

```
( (IP-MAT (NP-OB1 (D This))
          (NP-SBJ (PRO I))
          (VBD wrote)
          (PP (P at)
              (NP (PRO$ his) (ADJ own) (N desire)))
                  (. .))
(ID BOSWELL-1776,41.165))
```

Left-dislocations are parsed almost the same way as topicalizations like the one above, with the only difference being that the left-dislocated element has a "**-LFD**" dash tag and its resumptive element has a "**-RSP**" dash tag. You can find all object topicalizations in the corpus for clauses without auxiliaries with a query like the following one.

Example: Object Topicalization

```
node: IP-MAT*
query: (IP-MAT* idoms NP-OB*)
AND (IP-MAT* idoms VBP|VBD)
AND (IP-MAT* idoms NP-SBJ*)
AND (NP-OB* precedes VBP|VBD)
AND (NP-OB* precedes NP-SBJ*)
```

Note that the query above is restricted to clauses without auxiliaries and it does not control for the position of the verb in relation to the subject (or other elements of the clause). In a real study of topicalization, one would probably want to include clauses with auxiliaries and also control for verb-movement.

Problem 4
Object Topicalization

i. For matrix non-copular clauses without auxiliaries: what is the overall frequency of object (indirect or direct) topicalization in the sample corpus **without V-to-C movement or V2**? You can include left-dislocation for the purposes of this exercise.

ii. For the same clause types, what is the frequency of embedded topicalization in the sample corpus? (Hint: you will have to exclude traces of objects.)

iii. In the matrix clause topicalizations of the type you found in (i), how many of these have pronominal subjects (compared with nominal subjects)?

In order to complete part (iii) of this problem, you will need to use a new search function, "**idomsonly**". As the name suggests, **idomsonly** means that one node immediately dominates another node and no other nodes (i.e. there is only a single daughter). You can use this function in order to add the pronominal subject condition to your matrix clause query.

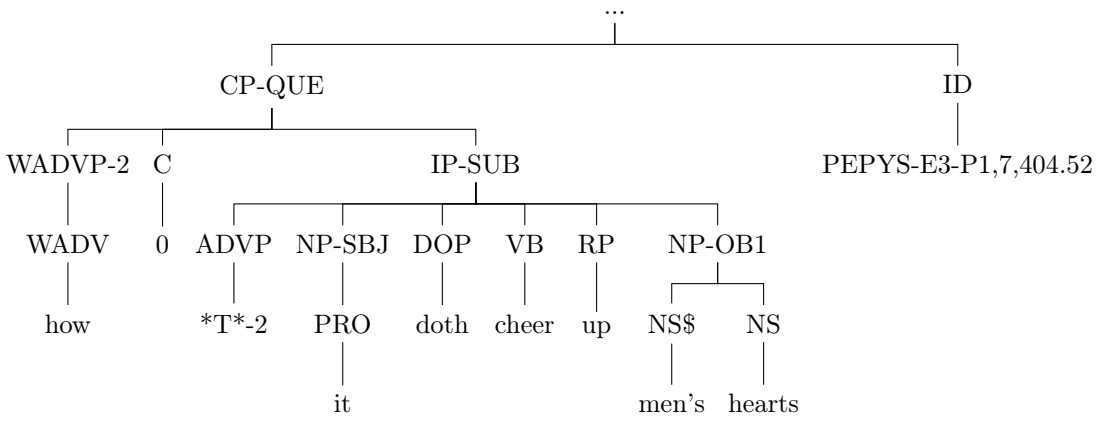
2.2 CPs in the corpora

Subordinate clauses (including all questions) in the corpus have a **CP-*** node, which immediately dominates an **IP-SUB** node. The CP node will also immediately dominate a C node (unless there is V-to-C

movement, as in direct questions), and a WXP node for all constructions that plausibly have operator movement (e.g. **CP-REL**, but not **CP-THT**). In the clauses that generally have operator movement, a null WXP is inserted if there is no overt *wh*-word, e.g. " **(WNP 0)** ".

Thus, an indirect question in the corpus has the following structure and parse:

(8) **Indirect Question**



(9) Actual parse for structure above:

```
(CP-QUE (WADVP-2 (WADV how))
        (C 0)
        (IP-SUB (ADVP *T*-2)
                (NP-SBJ (PRO it))
                (DOP doth)
                (VB cheer)
                (RP up)
                (NP-OB1 (NS$ men's) (NS hearts)))))
        (. .)) (ID PEPYS-E3-P1,7,404.52))
```

If we were interested in finding all relative clauses containing a null *wh*-element which is nominal in category, we could use the following query.

Example: Relative Clauses with null DP Operator

```
node: CP-REL*
query: (CP-REL* idoms WNP*)
AND (WNP* idomsonly 0)
```

If we were interested in seeing more detail about the nature of the extraction site, we can modify the above query and add some conditions. The query below searches for preposition stranding in relative clauses, and it introduces some new search functions, including "**sameindex**". This function means that two elements share the same numerical index, and it is very useful when looking at trace-antecedent

relationships. I have also removed the clause requiring a null operator for the purposes of this query.

Example: CPs with Extraction from inside a PP (preposition stranding)

```
node: CP-REL*
query: (CP-REL* idoms WNP*)
AND (CP-REL* idoms IP-SUB*)
AND (IP-SUB* idoms PP)
AND (PP idoms NP*)
AND (NP* idomsonly \*T*)
AND (WNP* sameindex \*T*)
```

The above query should return 199 hits in the sample corpus.

Problem 5
Overt and Zero *wh*-words and Complementizers

i. How many relative clauses in the sample corpus have an overt *wh*-word and a zero complementizer?

ii. How many relative clauses have an overt complementizer and a zero *wh*-word?

3 Answers to Problems

Problem 1. Order of direct vs. indirect object:

i. If you used the correct queries, included all clause types, and excluded traces in the appropriate way, you should have found 1018 examples of the IO > DO order in the DIGS sample corpus, but only 22 of the DO > IO order. Note that in order to complete the problem, you also must find a way to exclude topicalization from your queries (see section on topicalization).

ii. The latest text to include any example of the DO > IO is reade-1863.psd. So on the basis of this limited data, we can conclude that this pattern did not leave the language before 1863. (Note that this conclusion is only valid to the extent that this corpus is representative of the dialect and period of time under study, and it does not necessarily hold for nonstandard and/or Northern England varieties of English.)

Problem 2. The answers can be found in the following tables:

i. **Auxiliary Use and Clause Type**

	Auxiliary	No Auxiliary	%Auxiliary Use
Matrix	8183	15792	34.1%
Subordinate	11602	10943	51.5%

ii. The effect appears to be evenly distributed across the auxiliary constructions for the most part. However, modals, perfects, and passives account for most of the effect:

Auxiliary Type and Clause Type

	Modal	Perfect	Progressive	Passive	No Auxiliary
Matrix	4221 (17.1%)	1256 (5.07%)	192 (0.776%)	3298 (13.3%)	15792 (63.8%)
Subordinate	5630 (23.6%)	2532 (10.6%)	309 (1.29%)	4429 (18.6%)	10943 (48.9%)

Problem 3. i. There are only 4 potentially specificational matrix clauses with finite BE (and no nonfinite verb) in the sample corpus. (This is unfortunate for those of us interested in specificational sentences. But fortunately, there is a much larger corpus available than just the sample corpus!)

ii. Many of the clauses returned in the answer to (i) are not really specificational but are instead copular clauses with some fronted element and V2 (or V-to-C with a particular trigger). While these are formally ambiguous between specificational sentences and non-specificational sentences with V2, (ID FRYER-E3-P2,1,209.32) is clearly not specificational in context and (ID GIFFORD-E2-P2,F3R.60) is an *it*-cleft, so it cannot be specificational and *it* follows the verb because of V2.

iii. V2 in Early Modern English, and also V-to-C with triggers such as fronted negatives.

Problem 4. Answers to (i) and (ii) can be found in the table below:

	Topicalized	Object Elsewhere	%Topicalization
Matrix	138	5782	2.33%
Subordinate	15	3841	0.389%

Note also that almost all of the subordinate clause topicalizations occur in Queen Elizabeth I's translation of *Boethius*, which makes it likely that she is overusing the construction because of a translation effect from Latin.

iii. 107 of the 138 clauses have pronominal subjects. Note how uncommon it is for object topicalization to co-occur with a nominal subject.

Problem 5. i. 5820.

ii. 2242.

References

Kroch, Anthony, Beatrice Santorini, and Lauren Delfs. 2005. Penn-helsinki Parsed Corpus of Early Modern English. Size 1.8 Million Words.

Kroch, Anthony S., Beatrice Santorini, and Ariel Diertani. 2010. Penn Parsed Corpus of Modern British English. Size $\sim$ 950000 words.

Kroch, Anthony S., and Ann Taylor. 2000. Penn-Helsinki Parsed Corpus of Middle English. CD-ROM. Second Edition. Size: 1.3 million words.

Taylor, Ann, Arja Nurmi, Anthony Warner, Susan Pintzuk, and Terttu Nevalainen. 2006. York-Helsinki Parsed Corpus of Early English correspondence. Compiled by the CEEC Project Team. York: University of York and Helsinki: University of Helsinki. Distributed through the Oxford Text Archive.

Taylor, Ann, Anthony Warner, Susan Pintzuk, and Frank Beths. 2003. The York-Toronto-Helsinki Parsed Corpus of Old English Prose.

Wallenberg, J. C., A. K. Ingason, E. F. Sigurðsson, and E. Rögnvaldsson. 2011. Icelandic Parsed Historical Corpus (IcePaHC). Version 0.3. Size: 260 thousand words. URL `http://www.linguist.is/icelandic_treebank`.